Remote Sens. **2012**, *4*, 88-110; doi:10.3390/rs4010088

OPEN ACCESS

Remote Sensing

ISSN 2072-4292

www.mdpi.com/journal/remotesensing

Article

Extraction of Objects from Terrestrial Laser Scans by Integrating Geometry Image and Intensity Data with Demonstration on Trees

Shahar Barnea * and Sagi Filin

Department of Transportation and Geo-Information, Technion-Israel Institute of Technology, Haifa 32000, Israel; E-Mail: filin@technion.ac.il

* Author to whom correspondence should be addressed; E-Mail: barneas@technion.ac.il; Tel.: +972-4-829-5855; Fax: +972-4-829-2365.

Received: 29 November 2011; in revised form: 29 December 2011 / Accepted: 29 December 2011 / Published: 5 January 2012

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 still constitutes a gap in ensuring this data is usable for mapping. This paper proposes a framework for recognition of objects in laser scans; aiming to utilize all the available information, range, intensity and color information integrated into the extraction framework. Instead of using the 3D point cloud, which is complex to process since it lacks 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 considers the 3D nature of the data. The proposed algorithm is demonstrated on tree extraction and tested on scans containing complex objects in addition to trees. Results show a very high detection level and thereby the feasibility of the proposed framework.

Keywords: object extraction; terrestrial laser scanner; point cloud; data fusion

---

1. Introduction

Autonomous extraction of objects from terrestrial laser scanners becomes relevant when considering the data volume and the difficulty of interacting with irregularly distributed three dimensional point clouds. Additionally, the growing use of laser scanners for mapping purposes and the growing demand for reconstructing objects in 3D space turn such applications into a necessity. Indeed, object extraction from terrestrial laser scanners has been a research domain in recent years, ranging from reverse engineering problems to building reconstruction, forestry applications, and others [1]. In most cases, a model driven approach is applied, where domain knowledge about the sought after object shape drives the reconstruction and recognition process. Rabbani *et al*. [2] models industrial installations by using predefined solid object model properties. Bienert *et al*. [3] propose an *ad hoc* approach for tree detection, which is 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 a well defined object shape. Pu and Vosselman [4] extract building elements using a decision rule based approach. In a later publication they recover full building models using a knowledge based reconstruction approach [5]. Becker and Halla [6] reconstruct polyhedral models of the buildings to determine laser points on window edges. The extraction models are focused on a single type of object and assume high domain-knowledge.

Alternative approaches, which can still be categorized as model driven, involve generating databases consisting of diverse instantiations of 3D objects. Upon the arrival of new, unseen data, a good matching score is sought 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 [7,8] and show good results using spin-image based descriptors. Additionally, Frome *et al*. [9] introduce 3D and harmonic shape context descriptors, and Mian *et al*. [10] present a matching score which is based on a robust multidimensional table representation of objects. These methods require generation of massive object instantiation 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 model driven approach, is based on the extraction of primitives (points, sticks, patches) and modeling their inter-relations as a means to recover the object class. Pechuk *et al*. [11] extract primitives and then map the links among them as cues for the recognition part. This is demonstrated on scenes containing a few well defined objects, having a relatively small number of primitives (e.g., chairs and tables).

Recent works, mainly related to scene understanding from images (e.g., [12–14]) have demonstrated how the application of segmentation processes for recognition tasks yields promising results, both in terms of object class recognition and correcting the segmentation of the searched objects. Such approaches are categorized as joint categorization and segmentation (JCaS). The main concern of these methods is to correctly categorize the objects of an image towards scene understanding, while compromising the exact segmentation of the object. In contrast, when using terrestrial laser scanning data the objective is to accurately detect all the 3D points related to an object, as a means of autonomous modeling and mapping.

Segmentation is a fundamental, mid-level processing phase, which involves grouping pixels containing redundant information into segments. It concerns partitioning the data into disjoint, salient,

regions, usually under the assumption that segments tend to represent individual objects or object parts. Recent table scanners related segmentation works have proposed using genetic algorithms to estimate planar and quadric surfaces [15], using a randomized Hough transform to extract planar surfaces from the scan [16], or development of a surface selection criterion to evaluate the quality of the extracted segments [17]. Lavva *et al*. [18] handle the data segmentation according to primitive fitting, which assumes that the scene consists of well-defined objects. All these methodologies have been applied to relatively simple datasets that do not represent the typical complexity of natural scenes. With terrestrial laser scanning data, extraction of planar faces has mostly been considered, e.g., Dold and Brenner [19], Biosca and Lerma [20] who use fuzzy clustering to extract planar surfaces, or Wang and Tseng [21] who use an octree based approach. Gorte [22] extracts planar faces using a panoramic representation of the range data. A model driven segmentation into a more general class of primitives, including planes, cylinders, and spheres, has been proposed by Rabbani *et al*. [2]. While useful in reverse engineering practices, it cannot be easily extended into general scenes. Therefore, we propose a new segmentation approach that can handle planar faces, as well as unstructured ones.

Contrasting with model driven approaches, we examine here the extraction of objects from detailed geometric information and radiometric data using a small number of training datasets and with limited domain knowledge. The approach is based on variability measures and on learning object characteristics. In general, the aim is to minimize the required domain knowledge by keeping the set features as "low-level" as possible and without relying on object libraries, databases of rules, or domain-knowledge. It is applied by first integrating the data into a common reference frame, which is then segmented into regions that are classified into "object" and "not-object" segments. A refinement phase eliminates non-related points and augments unsegmented ones. We demonstrate this approach on tree detection, primarily for their shape complexity. As we demonstrate, the choice of descriptive features makes the classification, which is the core of the proposed model, successful even using a relatively small training set. Using the proposed model achieves better results than those obtained by the individual segmentation, ones which reflect the actual scene's content.

2. Methodology

The irregular point distribution, large size of the point cloud and dominant scale variations, makes elementary tasks, e.g., establishing connectivity and accessing and processing the data challenging. Consequently, data representation becomes a fundamental consideration, particularly when different datasets are involved. Additionally, as images are a two-dimensional projection of 3D space their integration with the three-dimensional laser data should account for projection and dimension differences. We therefore begin with a description of the preprocessing stages.

2.1. Preprocessing

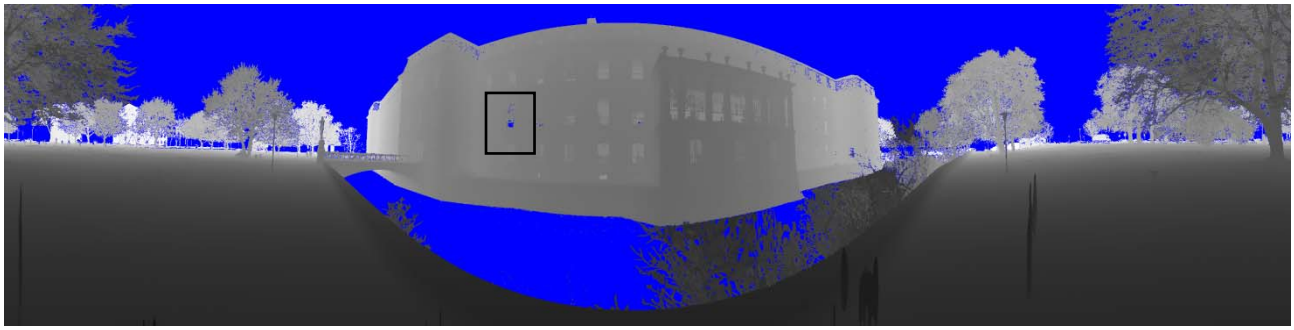
2.1.1. Data Representation

When processing range data, most approaches are applied directly to the point cloud in 3D space. The hard task is to calculate the descriptive information in the irregularly distributed laser point cloud. To facilitate the processing, the point cloud is treated in its panoramic representation,

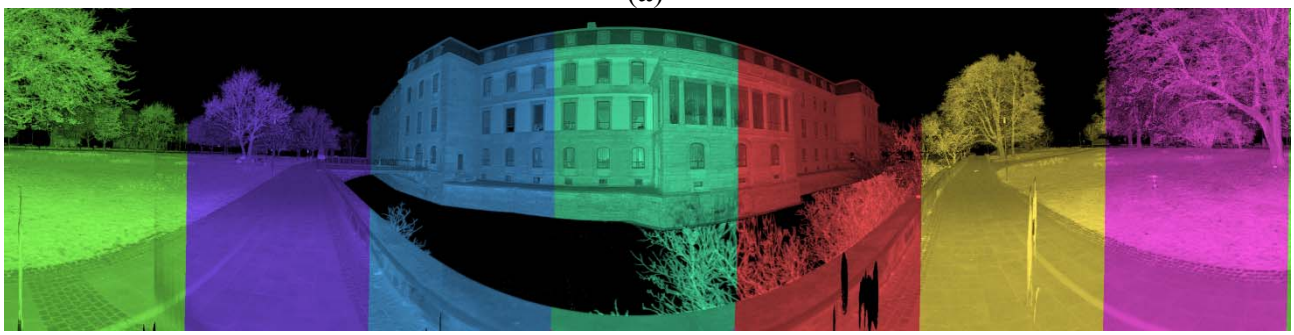
$$[x \quad y \quad z]^{\mathrm{T}} = \rho[\cos\varphi\cos\theta \quad \cos\varphi\sin\theta \quad \sin\varphi]^{\mathrm{T}} \tag{1}$$

where θ and φ are the latitudinal and longitudinal coordinates of the firing direction, and ρ the measured range. Because of the fixed angular spacing (defined by system specifications), a lossless grid structure can be set, thereby offering a compact image representation of the data (Figure 1(a)). Use of the image representation makes data manipulation and primitive extraction simpler to perform compared to directly processing the 3D point cloud. Thus, the other input channels are also transformed into it.

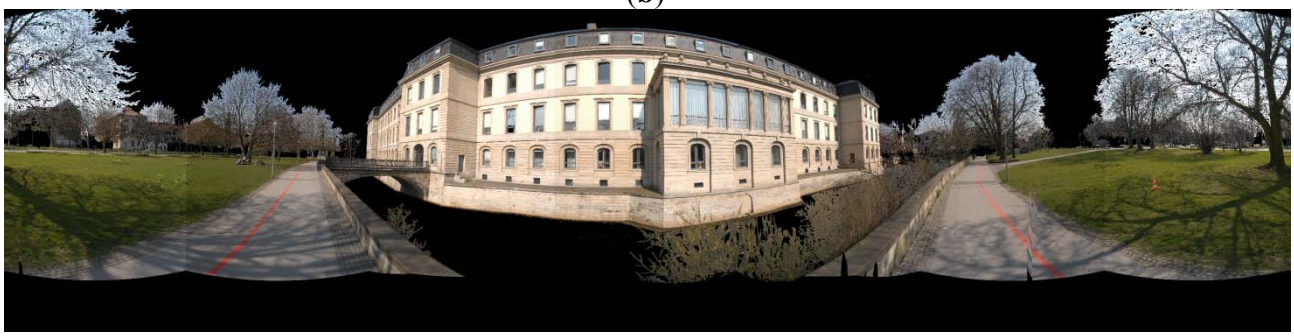
Figure 1. Polar representation of the color cue projected onto the range data. (**a**) The range data in polar representation (encircled is a no-reflectance area). (**b**) Selected region of the seven individual images for the construction the color channel. (**c**) Color content as projected to the scanner system.



(**a**)



(**b**)



(**c**)

2.1.2. Camera to Scanner Co-Alignment

Integration of the two data sources requires securing their co-alignment. The relative/boresight transformation between the mounted camera and scanner body consists of rotation, **R**, and a 3D offset,

t, and can be accommodated via the projection matrix $\mathbf{P}_{3\times4}$ between a 3D point, X, and a 2D image point, x, in homogeneous coordinates,

$$x = \mathbf{KR}\left[\mathbf{I}\,|\,-t\right]X = \mathbf{P}X \tag{2}$$

where

$$\mathbf{K} = \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

f_x and f_y are the focal lengths in the x and y directions respectively, s is the skew value, and x_0 and y_0 are the principal point coordinates [23]. The parameters in **K** account for the intrinsic camera parameters, while radial and decentering lens distortions are also calibrated and corrected for prior to their use in Equation (2). For each scan, n images are acquired at predefined "stops" (every 360/n degrees), thus n transformation matrices are computed. Assuming that, (i) the camera is rigidly mounted to the scanner, (ii) the intrinsic camera parameters are fixed and calibrated in advance, and (iii) the acquisition position (of the "stop") is fixed across all scanning positions, enabling use of the same projection matrices for all images of the same "stop" within different scans.

The camera-to-scanner linkage enables rendering the color content on the 3D laser points. Relating each 3D point to the best of the n images is implemented by projecting the point to all images and selecting the one where the projected 3D point's position is the closest to its perspective center (Figure 1(b)). It is noted that within the point's projection computation, points that appear at the back of the camera are avoided. The image projection into 3D space leads to some content loss due to change in resolution, but allows for processing of both information sources in a single reference frame (Figure 1(c)).

2.1.3. Data Cleaning

Prior to segmenting the range images, the data is cleaned by filling void regions and removing isolated range measurements. Void regions are caused by "no-return" areas in the scene (e.g., the skies) or "no reflectance" areas from object parts (*cf.* the encircled window in Figure 1(a)). No-reflectance regions appear as holes in the data (size is defined by a preset value) and are filled with neighboring values. No return regions are filled with a background value (maximal range).

2.2. Segmentation

Since most scenes are cluttered and contain entities of various shape and form, structured and unstructured, the data is segmented using multiple cues. Instead of seeking consistency along a single cue, which is likely to provide partial results, richer descriptive information is obtained by characterizing entities via a broader set of properties.

The segmentation makes use of the range data, derived surface normals, and color content as the three individual cues. Its outcome should form a collection of 3D points that must adhere to two conditions: One that segments maintain geometrical connectivity among all points constituting it, and the other is that features of the connected set of points share some degree of similarity. Similarity can be geometrically based, radiometrically based, or both. The transformation of the data into a panorama allows the use of image segmentation procedures for segmenting the point-cloud. We have chosen the

use of the mean-shift segmentation [24], due to its successful results with complex and cluttered images. Being non-parametric, it requires neither model parameters nor domain knowledge as inputs. The algorithm is controlled by only two dominant kernel related parameters: the spatial and range dimension size. The first affects the spatial neighborhood while the latter affects the permissible variability within the neighborhood. These two parameters are physical in a sense [24].

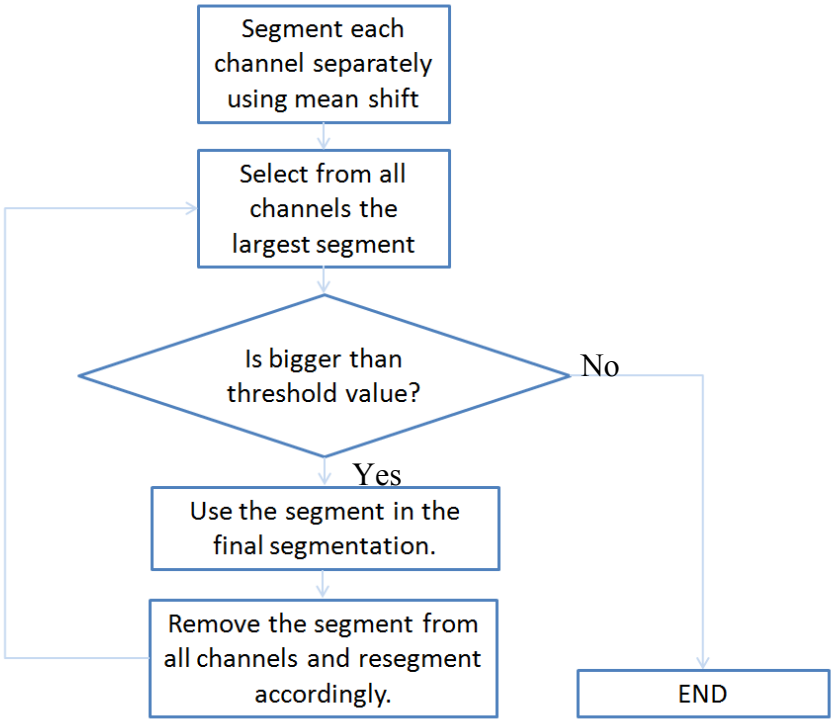
While the mean-shift acts as an instrument to segment the channels, the challenge is to handle the different space partitioning of the individual channels. The aim is to avoid segmentation that either overlays segments or concatenates all features into a feature vector. Such segmentation will be highly dimensional, computationally inefficient, and ultimately yield an over-segmented dataset. Recognizing that the individual channels reflect different data properties and provide optimal segments in some parts, each channel is segmented independently. Then a segmentation that integrates the individual channels is constructed by selecting the significant segments among each. This approach assumes that relevant segments exist in each channel; so, extracting the significant ones from the individual channels can provide a segmentation that features the dominant phenomena in the scene.

Generally, the objective is to obtain spatially significant and meaningful segments that are uniform in their measured properties. To achieve a spatially significant grouping in object space, a segment score is set as a function of its size in object space. Due to varying scale within a scan, a segment size in image-space (measured by the number of pixels) does not provide a suitable measure. Instead, a 3D coverage, r, is computed via

$$r = \Delta\varphi \cdot \Delta\theta \sum_{s \in S} \rho_s \Rightarrow r \approx \sum_{s \in S} \rho_s \tag{3}$$

where $\Delta\varphi$, $\Delta\theta$ are the angular spacing, and s is the set of pixels relating to segment S.

Figure 2. Iterative workflow of a multi channel segmentation process.



The proposed model is applied as follows: first, the largest segment is selected from all channels; if the segment quality is satisfactory, it is inserted into the integrated segmentation. All pixels relating to this segment are then subtracted from all the channels and isolated regions in the other channels are then regrouped and their attribute value is computed. Following, is the extraction of the next largest segment and the repetition of the process until reaching a segment whose size is smaller than a prescribed value and/or preset number of iterations. Figure 2 presents a flowchart diagram of the process. It is noted that due to the non-parametric nature of the mean-shift segmentation, re-segmenting the data between iterations has little effect. We also note that this segmentation can be further refined [25], but results obtained using this approach are sufficient to the present purpose.

2.3. Feature Space

To perform the segment classification, a set of descriptive cues for each of the segments is computed. Maintaining the framework as general as possible, only low-level features are considered. These features should describe both the internal textural characteristics of the segment and the characteristics of its silhouette shape. For a simple description, the aim is characterizing the object using a small set of descriptive features. Limiting the set of features is useful for avoiding dimensionality related problems as well as overfitting concerns.

The studied set of features is derived from all three channels: the range, intensity and color. For each of the channels the aim is to find the most distinguishable cue with respect to the targeted object. The following cues, which are derived from the individual channels, are evaluated: (i) range: cornerness, sum of derivatives, and the sum of the absolute values of the derivatives; (ii) intensity: the intensity values itself and cornerness; (iii) color: HSV and Luv representation. A mathematical formulation of the descriptive features for each of the cues is given in the Appendix.

2.4. Classification

Computation of features for each segment in the training set allows for creation of the feature space. As the data may not follow the classical form of two, well separated, hyper-Gaussian distributions, a non-parametric classification method is applied. We use the k-Nearest Neighbors (k-NN) algorithm [26] because of its simplicity and efficiency. The k-NN model evaluates an input sample against its neighborhood in the training data. Following the extraction of the *k* nearest neighbors from the training data, a voting procedure is performed. If more than *h* neighbors belong to either of the classes, the sample is classified accordingly.

The k-NN model is greatly affected by the different metrics of the individual cues, particularly because of their different units and scale. Great differences are expected in scale and distribution, motivating the need to normalize the data. For normalization, the whitening (Mahalanobis) transformation [26] is used to transform the data into the same scale and variance in all dimensions. If $\mathbf{X}$ is a training set of size N×3, with N, the number of segments distributed with $\sim\{\mu, \mathbf{\Sigma}\}$; using the SVD, $\mathbf{\Sigma}$ can be factorized into $\mathbf{\Sigma} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$, where $\mathbf{U}$ is orthonormal, $\mathbf{U}\mathbf{V}^{\mathrm{T}} = \mathbf{I}$, and $\mathbf{D}$ a diagonal matrix. The transformed $\mathbf{X}$ is calculated by:

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

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

The *k*-NN depends on the number of evaluated neighbors (*k*), and on the cardinality parameter (*h*). A bigger *k* will lead to a more general model (using more samples to decide means that more information is weighed in) but less distinct (the extreme is when all samples are used as neighbors). The choice of *h* affects the sensitivity of the classification model. Assigning *h* too small of a value, the model may become error prone, whereas setting *h* too strictly the number of false positives decreases at the expense of a large number of false negatives (type *II* error). Our choice of *h* is based on finding a value that provides 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 trial a confusion matrix, C, is recorded

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

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

2.5. Refinement

Thus 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 and background, while some may hold tree characteristics but are, in fact, non-tree segments. The refinement phase aims at: (i) linking segments that are part of the same tree, (ii) minimizing the number of false alarm detections, and (iii) separating mixture segments into object and background. Generally, this can be described as a split and merge problem among segments. It is approached here differently by weighing the inter-relation between the individual points, so that neighboring points (by 3D proximity measures) indicate potentially tight relations and consequently stronger utility in their link.

A common theme in object to background separation algorithms is the desire to group pixels that have a similar appearance (statistics) and to have distinct boundaries between pixels in different classes. Restricting the boundary measurements to be between immediate neighbors and compute region membership statistics by summing over pixels, we can formulate this as a classic pixel-based energy function problem, which is addressed here via graph cut energy minimization of the following term:

$$E = \sum_i E_d(i) + E_s(i) \tag{7}$$

with *E* the total energy, E_d, the data term, related to the "wish" of laser point to maintain its original classification and E_s is the smoothness term, which relates to the "wish" of highly connected points to share 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.

The energy function, E, can be modeled by a flow graph, where each point, i, in the cloud is a vertex (V_i), and additionally, two more vertices, a source (S) and the sink (T) are added. The E_d, term is computed by

$$E_d = f(p_i) \tag{8}$$

where p_i is the probability that i is an object. The probability is set according to the segment classification stage (high value to pixels that were assigned as an object and low to those assigned as non-object). The E_d elements are applied through the weights (capacities) assigned to the edges that link each point to the source and the sink. Points that are more compatible with the foreground (tree) or background (non-tree) region obtain stronger connections to the source or sink, respectively. Thus, the weights on the edges of a source-to-node and on a node-to-sink is set according to

$$\begin{aligned} w(S, v_i) &= |p_i - \alpha| \\ w(v_i, T) &= 1 - |p_i - \alpha| \end{aligned} \tag{9}$$

where α is the possible error in assigning a point to a given class (an 0.05 value was applied here).

The capacities of the other edges in the graph are derived from the smoothness energy term (E_s), *i.e.*, adjacent points have stronger links. The boundary term is therefore set as the inverse proportion of the 3D Euclidian distance between points i, j. In order to avoid a full graph containing all possible i, j links, the only pairs we assign are nearest-neighbor points, j, for each point i in the point-cloud. For each such pair (i,j) a link between the two nodes is built. The search for the nearest neighbor is performed via the Approximate Nearest Neighbor (ANN) method [27]. This process leads to a sparse and controllable graph.

Once the flow-graph is set, a minimum-cut/maximum-flow problem can be solved using a polynomial time algorithm (e.g., [28,29]), pixels on either side of the computed cut are labeled according to the source or sink to which they remain connected. The graph-cut algorithm finds the minimal cut (and the maximal flow) of the graph which also minimizes the energy function. The outcome of the graph cut based refinement algorithm is the separation between "object" and "non-object" points.

3. Results and Discussion

The proposed approach was tested on two sets, one containing 16 scans and the other eight. The scans were acquired by the Riegl LMS Z360i laser scanner (angular resolution of $\Delta\theta = \Delta\varphi = 0.12°$) generating 2.25 million ranges (creating a 750 × 3,000 pixels image), spanning 360° horizontally and 90° vertically with maximum ranges on order of 80 m. Seven, six-mega-pixel size images were acquired per scan using a Nikon-D100 camera with a 14 mm lens, and were processed in full resolution. Overlap between images was ~13.5%, equivalent to ~270 pixels. The first set covers a courtyard-like square and offers a typical urban environment that is dominated by manmade objects. The second set covers mainly an open park area, bordered by a castle façade, with trees and vegetation in a variety of forms and at different distances from the scanner (Figure 3). Both sets feature a significant amount of occlusions and clutter, featuring, cars, buildings, and other complex objects, in addition to trees.

As a first step, the response of the object for different cues derived from the input channels was examined. Beginning with the range channel (Figure 3) the cornerness, sum of derivatives, and sum of the absolute values of the derivatives were tested (Figures 4–6). For each of the cues, the distributions related to "tree" and "non-tree" classes are presented. These are made in reference to manually

extracted trees as ground truth. The distributions show that among the three, the one with the strongest distinction power is the cornerness.

Figure 3. Range data—grayscale values represent ranges from the scanner.

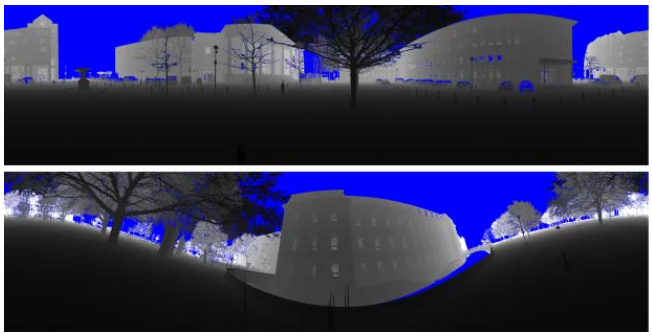


Figure 4. Cornerness values of the range channel and their tree/non-tree distribution.

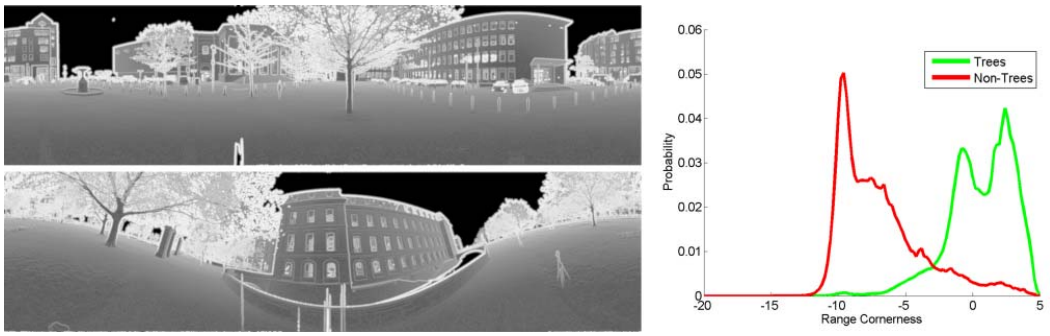


Figure 5. dx + dy values of the range channel, and their tree/non-tree distribution.

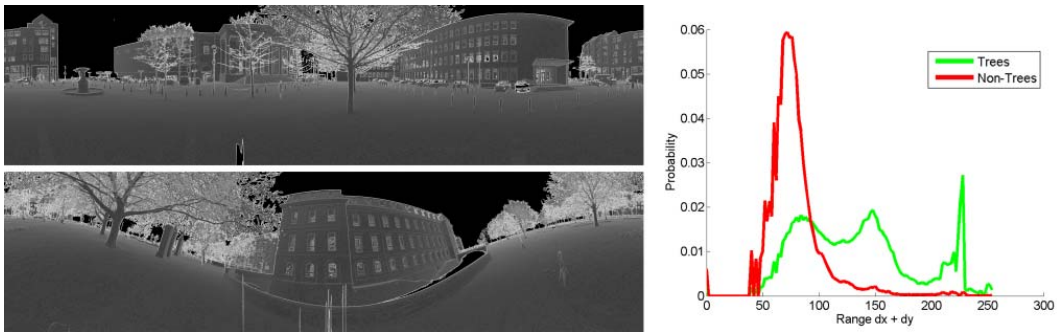
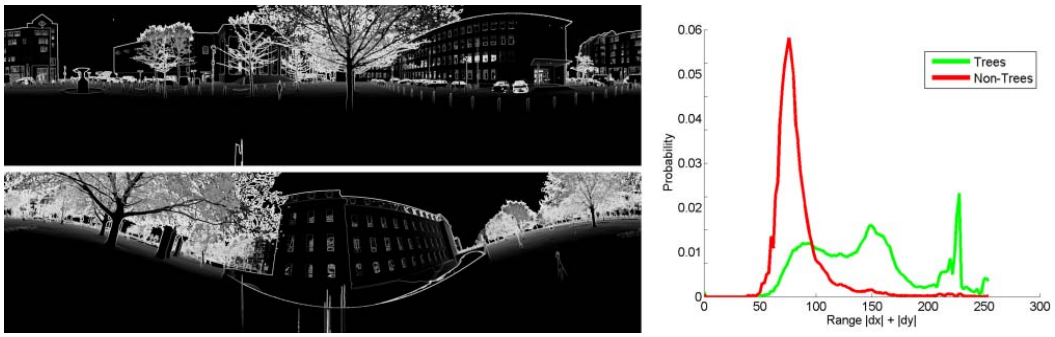


Figure 6. |dx| + |dy| values of the range channel, and their tree/non-tree distribution.



For the intensity channel, the two cues tested were the intensity values and cornerness (Figures 7 and 8). Between the two, the cornerness has the more pronounced separation, relative to the intensity cue where tree and non-tree classes are mixed.

Figure 7. The intensity channel, and the intensity values tree/non-tree distribution.

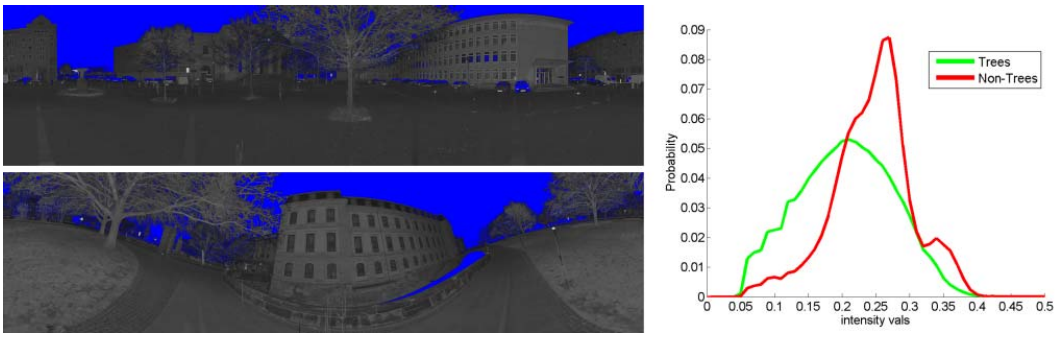
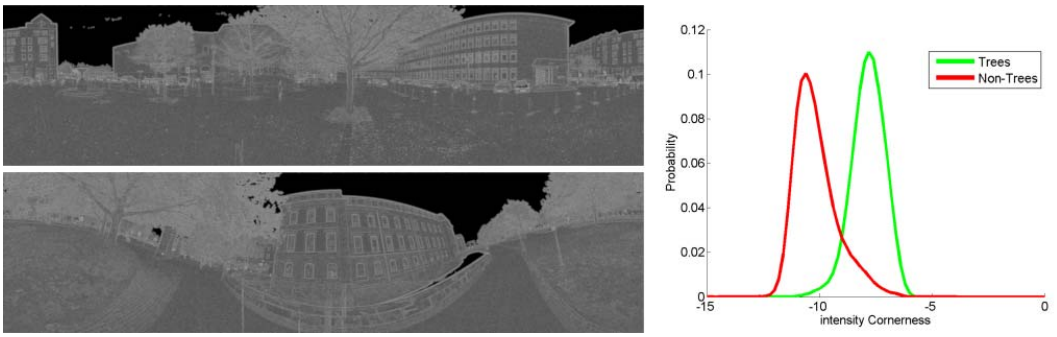


Figure 8. Intensity cornerness values, and their tree/non-tree distribution.



With the RGB related cues (Figure 9) the Luv decomposition was tested for the *u* (Figure 10), *v* (Figure 11) and *L* (Figure 12) values. As can be seen, the *L* value has no contribution to the separation, while *u* and *v* show the ability to resolve trees and non-trees segments (their combine effect is evaluated in the following). The HSV model was tested for the hue and saturation channels. Contrasting the *u* and *v* channels, neither channel adequately separated the two classes (data not shown). Among all tested cues, the range- and intensity-cornerness values provide the highest descriptive power for tree related points.

Following the testing of the single channel distinguishing power using a 1D histogram, we can test in a similar manner the distinguishing power of pairing two individual cues using a 2D histogram. The first natural cue coupling is the intensity and range cornerness measures since these two cues provide the most distinct separation individually. Their combination (Figure 13) shows indeed that the two enable separating the two classes, and that the correlation between the two classes is limited. Namely, each of them contributes to the separation (this can be noticed in the two-dimensional spread). Comparing the Hue-Saturation combination against the *u*-*v* combination (Figure 14(a,b) respectively), the *u*-*v* separation appear to perform better relative to the Hue-saturation. It is noted that the trees are leafless, and therefore this set may bias this conclusion. Even though the distinction power is limited compared to the range and intensity cornerness measures, they still show a level of separation which can be of value.

Figure 9. The RGB channel

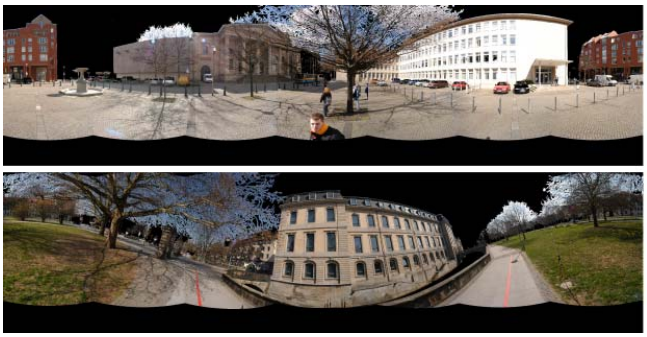


Figure 10. *u* values of the Luv representation of the color channel, and their tree/non-tree distribution.

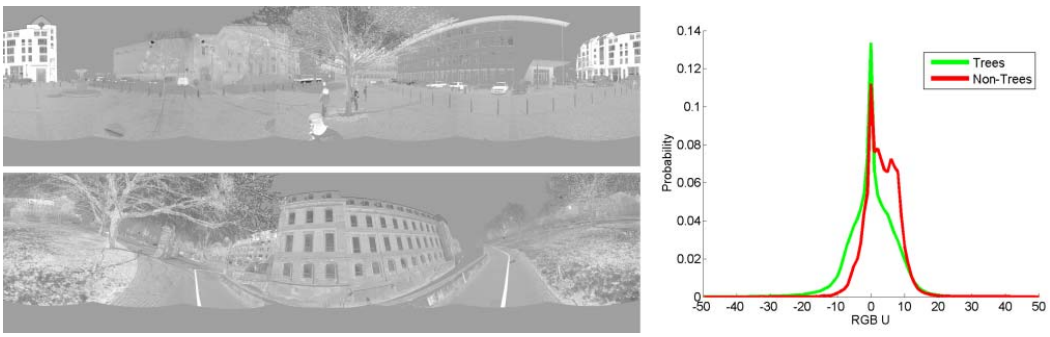


Figure 11. *v* values of the Luv representation of the color channel, and their tree/non-tree distribution.

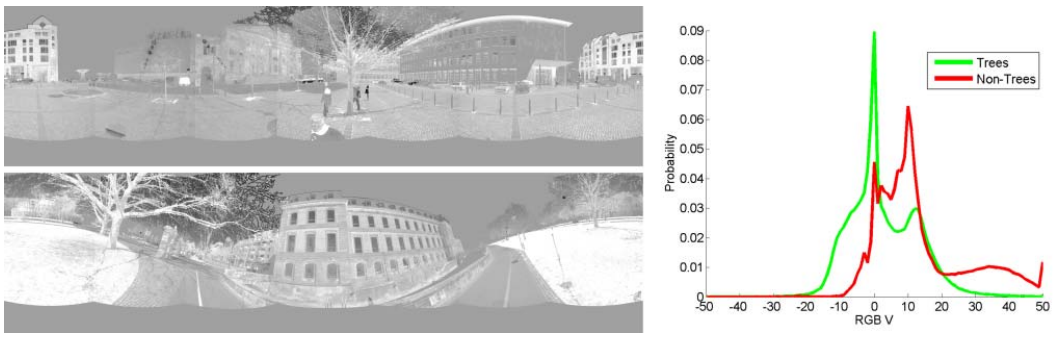


Figure 12. *L* values of the Luv representation of the color channel, and their tree/non-tree distribution.

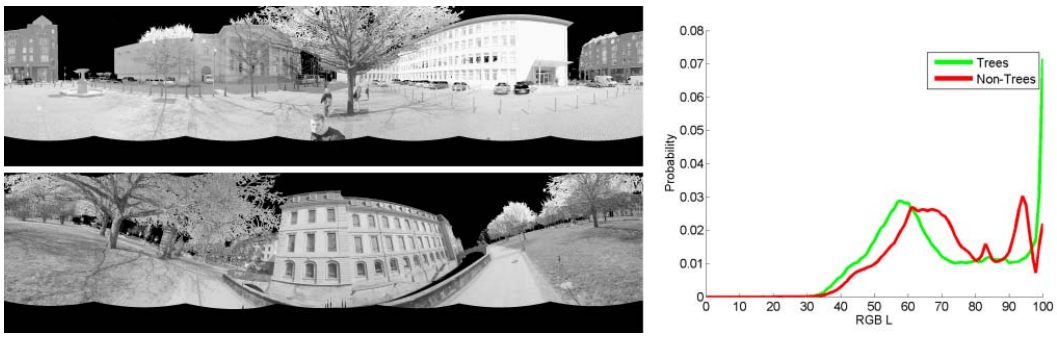


Figure 13. 2D distribution of the combined range and the intensity cornerness cues. The 2D feature space is made of 100 × 100 bins; each bin represents the rate of this cue set appearance in the training set.

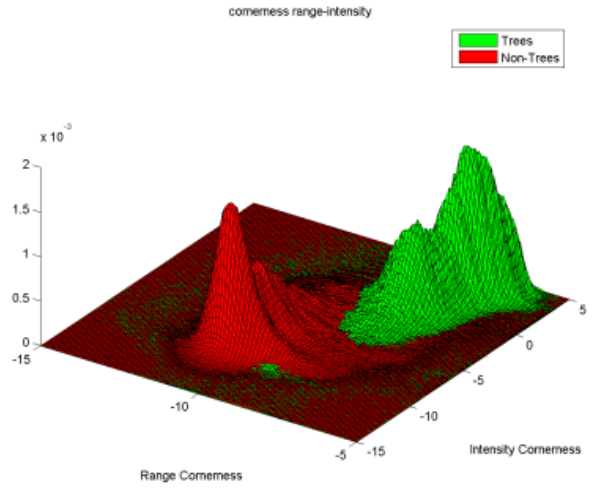
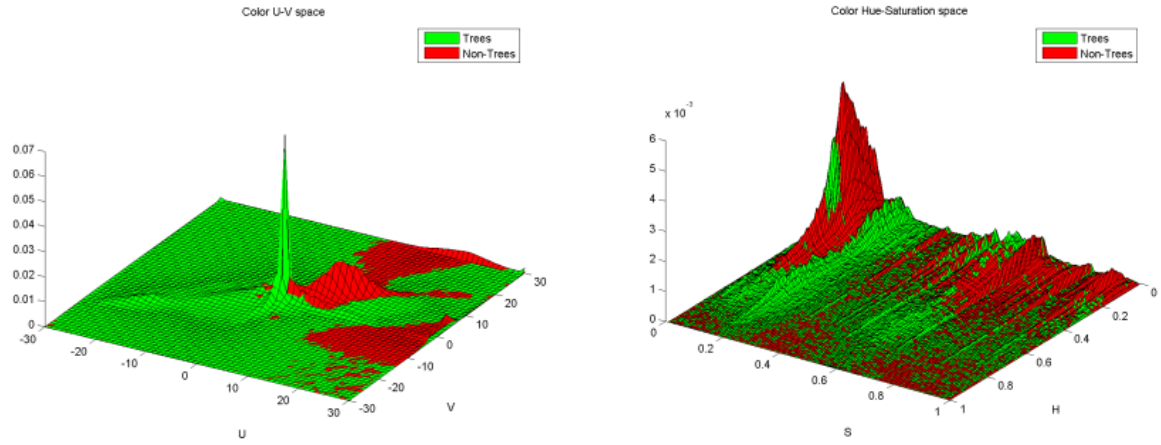


Figure 14. (**a**) 2D distribution of the combined *u*-*v* cues. (**b**) 2D distribution of the combined Hue and Saturation cues. The 2D feature space is made of 100x100 bins; each bin represents the rate of this cue set appearance in the training set.



(**a**) (**b**)

Next, the ability of the cues to separate tree segments (in contrast to pixels as in Figures 4–14) is studied. First, the detection power of the range and intensity cornerness measured is tested as they showed the highest distinguishing power both as an individual cues and as a combined cue in pixel-level examinations. For each segment the value of each of cues was averaged, creating a 2D feature vector. A scatter plot of the 2D feature space (the average range's and intensity's cornerness values per segment) is presented in Figure 15 (using scan 23—Figure 1(a)). For better visual emphasis, circle size is plotted in Figure 15 as a function of the segment size. Notice that the distribution along the two axes is not equal and that there is a correlation between the two cues. Following the whitening transform (Equation (4)) the two dimensions are equally scaled and become uncorrelated (Figure 16).

Figure 15. Distribution of the segment-averaged range-cornerness and intensity-cornerness values. Green/red, tree/non-tree related segments, respectively, according to the ground-truth. The circle size is proportional to the segment size.

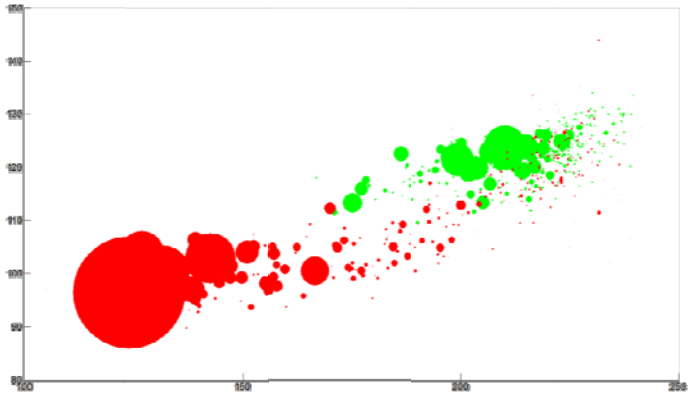
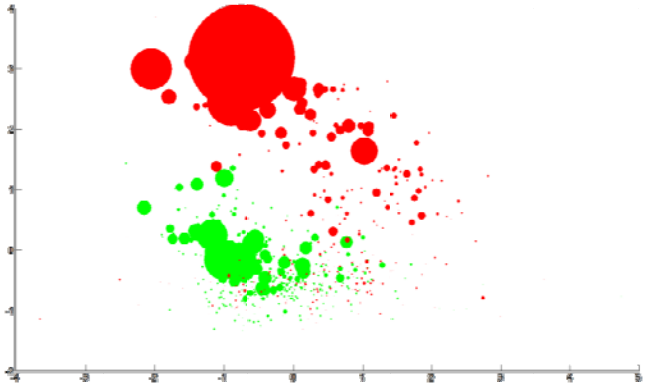


Figure 16. Distribution of the segment-averaged range-cornerness and intensity-cornerness values following the whitening transform. Green/red, tree/non-tree related segments, respectively, according to the ground-truth. The circle size is proportional to the segment size.



Analyzing the *k*-NN algorithm performance (discussion on the parameter selection appears in the following), the classification results are shown in Figure 17. As can be seen, most segments were correctly classified. On the pixel level the confusion matrix is:

$$C = \begin{bmatrix} 0.2246 & 0.0101 \\ 0.0061 & 0.7592 \end{bmatrix}$$

leading to an accuracy of 0.9838. Figure 17 bottom shows examples for type I and type II error entities. Testing the effect of adding the *u*-*v* color cues shows improvement, with the pixel level confusion matrix reaching:

$$C = \begin{bmatrix} 0.2250 & 0.0050 \\ 0.0057 & 0.7642 \end{bmatrix}$$

leading to an accuracy of 0.9892. As can be observed from the matrix, the false negative (type II errors) value has been cut in half. The classification results as applied on the dataset are presented in Figure 18.

Figure 17. Classification results using range and intensity cornerness cues. Green: true positive, Cyan: true negative, Magenta: false positive, Red: false negative.

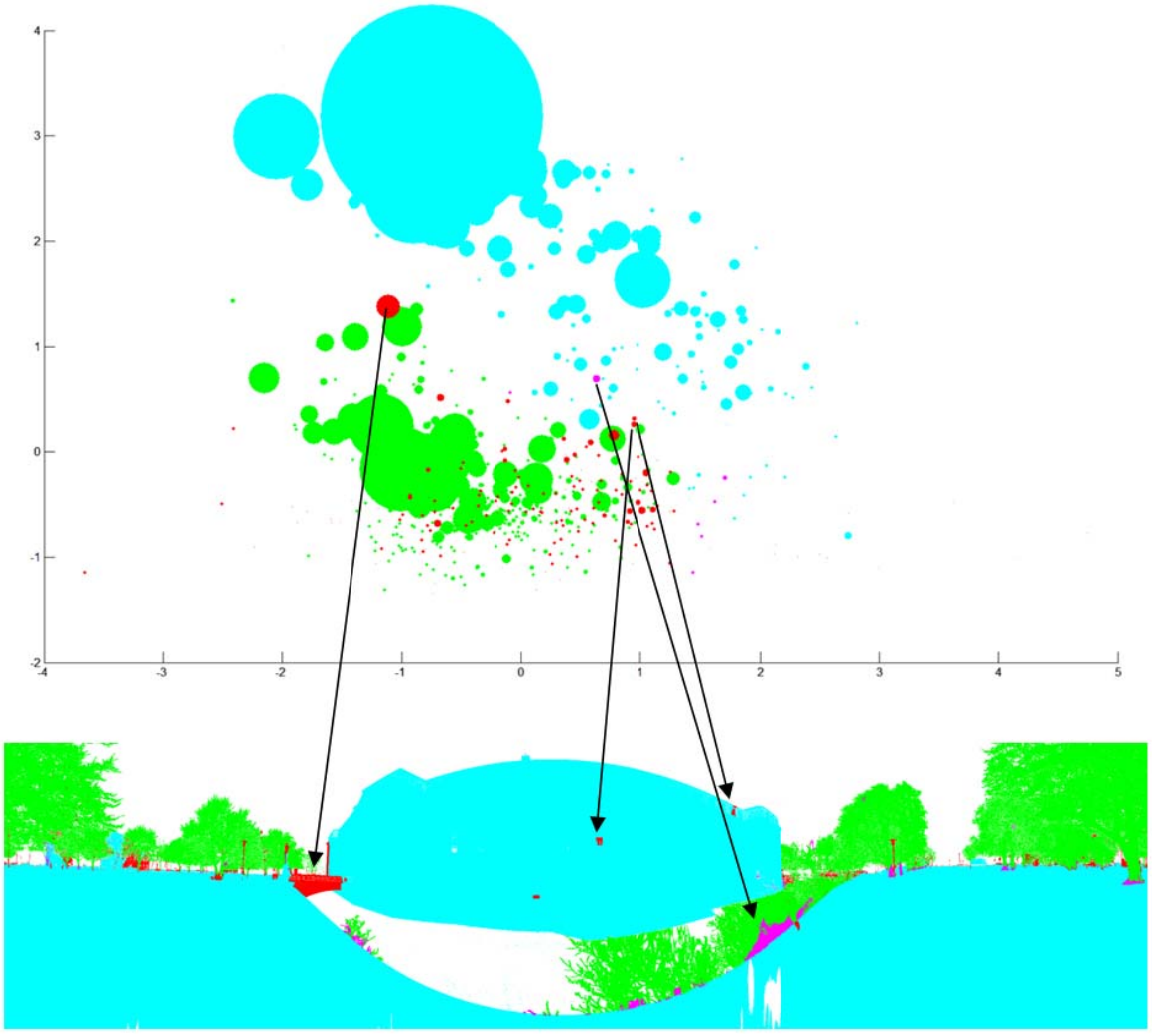


Figure 18. Classification results using range cornerness, intensity cornerness, and *u*-*v* cues. Green: true positive, Cyan: true negative, Magenta: false positive, Red: false negative.

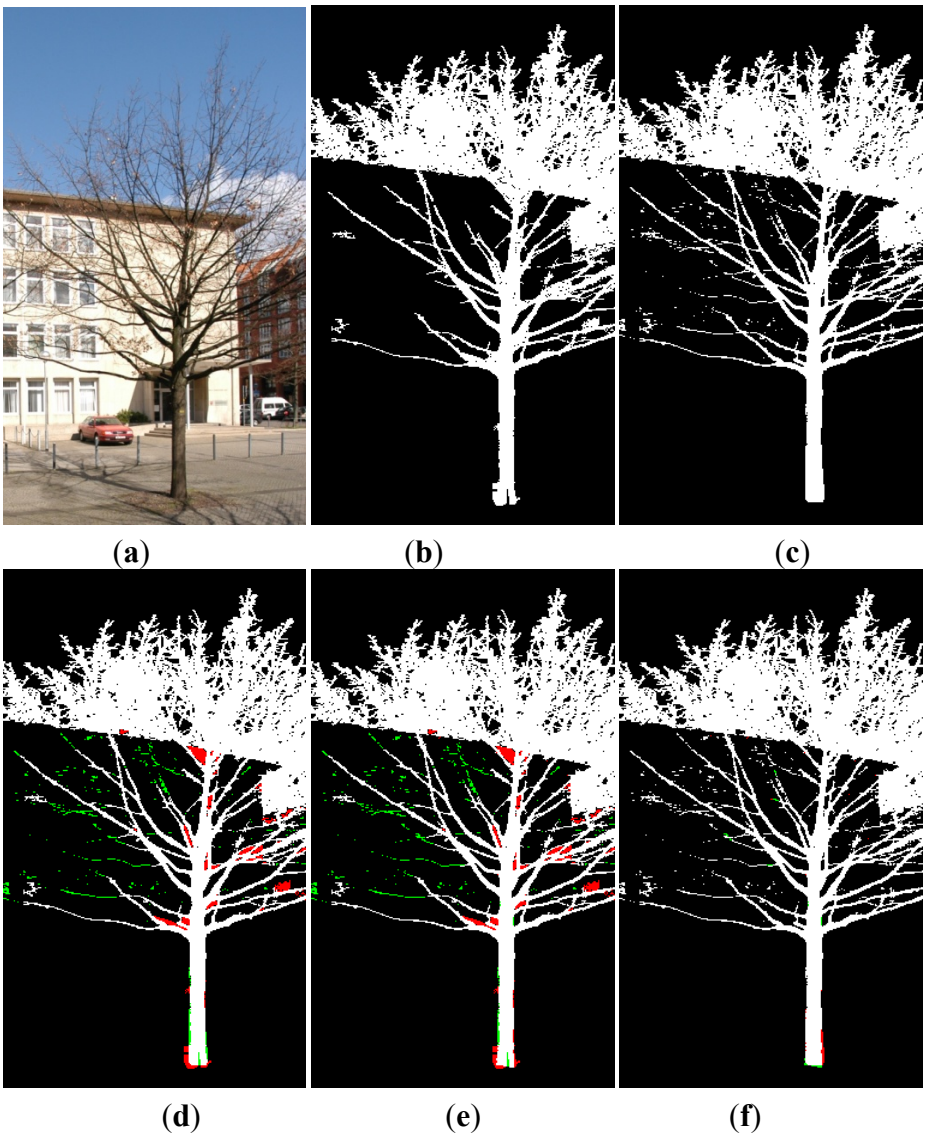
Applying the graph-cut refinement phase is demonstrated on a single tree (Figure 19(a)), and its outcome is shown in Figure 19(b,c), which show the 'before' and 'after' refinement states. Figure 19(d) shows the points that have changed their labeling. Figure 19(e,f) shows type I and type II errors before and after the refinement phase respectively, in which the number of misclassified points is greatly reduced. Small branches that were first unsegmented have now been added to the tree structure. The

only regions that were mis-classified following the refinement are at the bottom of the trunk where the connectivity to both ground and tree can be mixed. The application of the graph-cut on scan 23 further improves the confusion matrix to:

$$C = \begin{bmatrix} 0.2254 & 0.0049 \\ 0.0053 & 0.7643 \end{bmatrix}$$

leading to an accuracy of 0.9897.

Figure 19. Demonstration of the graph-cut application: (**a**) Image of the studied tree. (**b**) The classified tree segment before applying the graph-cut refinement phase. (**c**) The classified tree segment following the graph-cut application. (**d**) Differences following the application of the refinement phase: in green added points, in red deleted points. (**e**) Difference between the ground truth and the classified tree segment, before the graph-cut application: in red type-I error (false alarm) and in green type-II error (miss). (**f**) Difference between the ground truth and the classified tree segment, following the graph-cut application: in red type-I error (false alarm) and in green type-II error (miss).



Broadening the analysis, the algorithm is applied to eighteen scans that were acquired in urban and an open park environments. The classification phase is analyzed first. For the experiment, tree objects in these scans were manually marked and all related points were assigned as the ground-truth. The 24 scans generated ~16,000 segments total (~660 segments per scan). Learning by example models usually require a large training set data. Because of the relatively limited number of available scans, leave-one-out cross validation experiments were applied. For each scan the training feature space was recovered from the remaining 23 scans. As noted, the *k*-NN classification model depends on the choice of *k* and *h*. Following the formation of the feature-space, these parameters were studied by letting *k* vary between 1 and 11 while potential values for *h* ranged from 1 to *k*. The highest accuracy value recorded was ACC = 0.878 when using *k* = 9 and *h* = 5. The corresponding segments' confusion matrix (contrasting the pixels that were presented above) is:

$$C = \begin{bmatrix} 0.1861 & 0.0936 \\ 0.0283 & 0.6920 \end{bmatrix}$$

All confusion matrices resulting from this experiment (for all *k*'s and *h*'s) lead to the following receive operating characteristics (ROC) curve [26] in Figure 20. The area under the ROC curve is 0.872, which is evidence for good segment classification.

Following the classification examination, 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) and the correctly recognized background points (true-negative) resulting in

$$C = \begin{bmatrix} 0.2154 & 0.0052 \\ 0.0056 & 0.7738 \end{bmatrix}$$

and ACC = ~0.99. This result shows the high level of success of the complete algorithm. Notice that the pixel level classification is significantly higher than the segment level one. This effect is related to small non-tree segments that wear a form of tree segments, mostly because of their small size which is insufficient to obtain a consistent attribute. The refinement phase resolves most of these misclassifications. Examples for classification results on the complete scans are presented in Figure 21, marked on top of the intensity channel.

Figure 20. ROC curve created from k-NN classifications with different parameters.

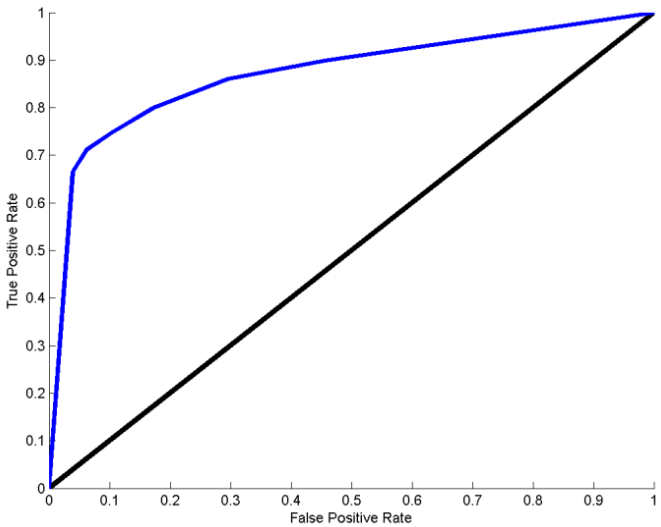
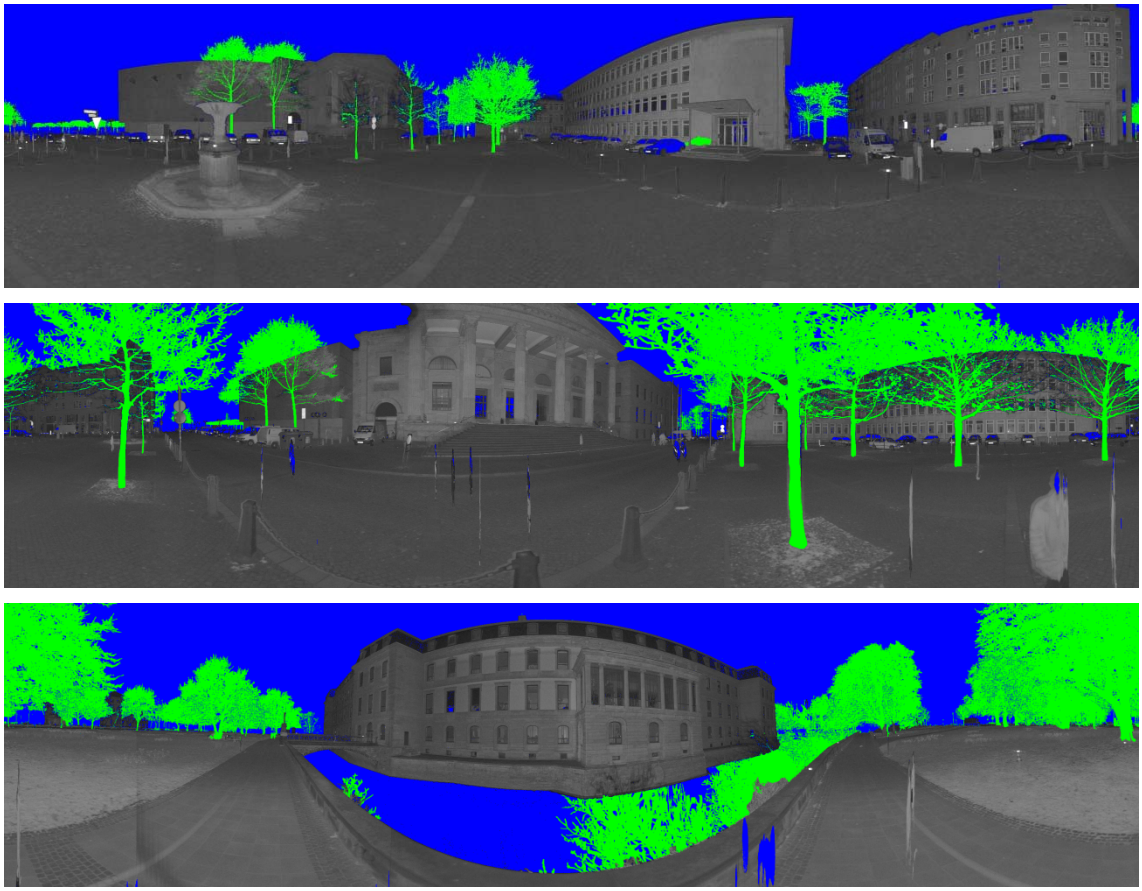


Figure 21. Results of the tree detection scheme applied on scans 3, 5, 23. The denser tree crowns against the sky background (contrasting those against buildings) are due to the last return operation mode of the laser scanner.



4. Summary

This paper presented a methodology for the extraction of spatial objects from terrestrial laser scanning data. The methodology is based on the fusion of individual channels, including range, intensity and color. In order to make the framework as general as possible, use of a set of low level features, such as cornerness, derivatives, and colorimetric measures, have been used. In contrast with existing model driven approaches, the proposed method is characterized by minimal reliance on domain knowledge. For the extraction, the channels are segmented first and then a set of features is extracted per segment. Each of the features is computed per segment and is then averaged. Classification of the segments is then carried out in a supervised manner using the non-parametric kNN classifier. To eliminate potential bias in the classification, the training data have been first normalized using the whitening transform. Following the segment-level classification, a pixel-level refinement phase has been applied using graph-cuts.

Results demonstrate 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 to identify trees both at different depths (scales) and partially occluded ones. Pixel classification accuracy was on the order of 99%. The small number of false alarm detections indicates the appropriateness of the selected features for the recognition.

Extensions of the proposed method would see avenues in the utilizing additional data channels, e.g., spectral data. It is noted that such channel addition is native to the proposed model but requires incorporation of an additional set of low-level features. Furthermore, the proposed approach can be extended to detect other objects e.g., buildings, and cars. Color saturation, size, surface smoothness, and other features which correspond to the channels to be used would be candidates for the application of such recognition.

Acknowledgments

Funding for this research was provided in part by the Volkswagen-Niedersachsen Foundation.

Appendix

The Set of Cues Evaluated in This Study

The following cues which are derived from the individual channels are evaluated: (i) Range: cornerness, sum of derivatives, and the sum of the absolute values of the derivatives. (ii) Intensity: the intensity values itself and cornerness. (iii) Color: HSV and Luv representation. Their mathematical formulation is:

Range:

(*i*) + (*ii*) *Sum & Absolute Sum of the First-Order Derivatives*

$$f_1 = \sum \left(d_\phi + d_\theta \right) \tag{10}$$

$$f_2 = \sum \left(|d_\phi| + |d_\theta| \right) \tag{11}$$

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, they are area dependent, and therefore are normalized with respect to the segment area. The choice of the two features is motivated by Bay *et al*. [30] who test the descriptiveness of the derivatives for the SURF descriptor. These 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.

(*iii*) *Cornerness of the Segment*

$$f_3(L_i) = \sum cornerness(L_i) \tag{12}$$

is motivated by the cluttered structure of the trees, which is expected to generate distinctive corner values due to the lack of structure. Cornerness is computed using the min-max operator [31]. The gradient projection in the minimal direction measures the point "cornerness" (*Cn*). Generally, because of their complex shape and depth variability, tree related segments tend to have high cornerness values.

In computing gradients, the need to control 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. 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 derivatives 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.

Intensity:

(*i*) *Intensity Value*

The choice of the intensity value as an attribute was motivated by the distinctive signature of vegetation in the near-infra-red spectral range, in which the scanner operates. The intensity value is expected to exhibit high values for vegetation features

(*ii*) *Cornerness*

The motivation is similar to that of using cornerness in the range data. Cornerness is computed using the min-max operator.

Color:

Two alternative color space representations were evaluated, the HSV and Luv. The first has an immediate link to the RGB representation, while the second is tightly linked to the XYy representation, which provides an undistorted color chart [32].

(*i*) HSV *channels*

The Hue, Saturation, Value model [33] captures the color content of the image objects. Trees are expected to be characterized by unique hue and saturation values relative to non-tree objects. The HSV are computed from the RGB values via:

$$\alpha = \frac{1}{2}(2R - G - B) \tag{13}$$

$$\beta = \frac{\sqrt{3}}{2}(G - B)a \tag{14}$$

$$H = \text{atan2}(\beta, \alpha) \tag{15}$$

$$S = \sqrt{a^2 + b^2} \tag{16}$$

(*ii*) Luv *Channels*

CIE-Luv is an attempt to define an encoding with uniformity in the perceptibility of color differences [32]. The non-linear relations for *L*, *u*, and v are given by:

$$XYZ = \begin{bmatrix} 0.4125 & 0.3576 & 0.1804 \\ 0.2125 & 0.7154 & 0.0721 \\ 0.0193 & 0.1192 & 0.9502 \end{bmatrix} RGB \tag{17}$$

$$L = \begin{cases} \left(\frac{29}{3}\right)^3 \frac{Y}{Y_N}, & \frac{Y}{Y_N} \le \left(\frac{6}{29}\right)^3 \\ 116\left(\frac{Y}{Y_N}\right)^{1/3} - 16, & \frac{Y}{Y_N} > \left(\frac{6}{29}\right)^3 \end{cases} \tag{18}$$

$$u' = \frac{4X}{X + 15Y + 3Z}, v' = \frac{9X}{X + 15Y + 3Z} \tag{19}$$

$$u = 13L \cdot (u' - u'_n),\ v = 13L \cdot (u' - u'_n) \tag{20}$$

where $u'_n = 0.1978,\ v'_n = 0.4683$.

The quantities u'_n and v'_n are the (u', v') chromaticity coordinates of a "specified white object", which may be termed the white point, and Y_N is its luminance. In reflection mode, this is often (but not always) taken as the (u', v') of the perfect reflecting diffuser under that illuminant.

References

1. Vosselman, G.; Maas, H.G. *Airborne and Terrestrial Laser Scanning*; Whittles Publishing: Dunbeath, UK, 2010.
2. Rabbani, T.; Dijkman, S.; van den Heuvel, F.; Vosselman, G. An integrated approach for modeling and global registration of point clouds. *ISPRS J. Photogramm.* **2007**, *61*, 355-370.
3. Bienert, A.; Maas, H.G.; Scheller, S. Analysis of the Information Content of Terrestrial Laserscanner Point Clouds for The Automatic Determination of Forest Inventory Parameters. In *Proceedings of the Workshop on 3D Remote Sensing in Forestry*, Vienna, Austria, 14–15 February 2006.
4. Pu, S.; Vosselman, G. Automatic Extraction of Building Features from Terrestrial Laser Scanning. In *Proceedings of the ISPRS Commission V Symposium "Image Engineering and Vision Metrology"*, Dresden, Germany, 25–27 September 2006; Volume 36.
5. Pu, S.; Vosselman, G. Building facade reconstruction by fusing terrestrial laser points and images. *Sensors* **2009**, *9*, 4525-4542.
6. Becker, S.; Haala, N. Refinement of Building Facades by Integrated Processing of LIDAR and Image Data. In *Proceedings of ISPRS Technical Commission III Symposium "Photogrammetric Image Analysis"*, Munich, Germany, 19–21 September 2007; Volume 36, Part 3/W49A, pp. 7-12.
7. Huber, D.; Hebert, M. Fully automatic registration of multiple 3D data sets. *Image Vis. Comput.* **2003**, *21*, 637-650.
8. Huber, D.; Kapuria, A.; Donamukkala, R.; Hebert, M. Parts-Based 3D Object Recognition. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 27 June–2 July 2004; Volume 2, pp. 82-89.
9. Frome A.; Huber, R.; Kolluri, T.; Buelow, M.J.; Recognizing Objects in Range Data Using Regional Point Descriptors. In *Proceedings of the European Conference on Computer Vision*, Prague, Czech Republic, 11–14 May 2004; Volume 3, pp. 224-237.

10. Mian, A.; Bennamoun, M.; Owens, R. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Trans. Pattern Anal. Machine Intell.* **2006**, *28*, 1584-1601.
11. Pechuk, M.; Soldea, O.; Rivlin, E. Function Based Classification from 3D Data via Generic and Symbolic Models. In *Proceedings of the National Conference on Artificial Intelligence*, Pittsburgh, PA, USA, 9–13 July 2005; pp. 950-955.
12. Russell, C.B.; Efros, A.; Sivic, J.; Freeman, T.W.; Zisserman, A. Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 17–22 June 2006; Volume 2, pp. 1605-1614.
13. Ladický, L.; Sturgess, P.; Alahari, K.; Russell, C.; Torr, P.H.S. What, Where & How Many? Combining Object Detectors and CRFs. In *Proceedings of the European Conference on Computer Vision*, Crete, Greece, 5–11 September 2010.
14. Singaraju D.; Vidal, R. Using Global Bag of Features Models in Random Fields for Joint Categorization and Segmentation of Objects. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, USA, 21–23 June 2011.
15. Gotardo, P.F.; Bellon, U.; Olga, R.P.; Boyer, K.L.; Silva, L. Range image segmentation into planar and quadric surfaces using an improved robust estimator and genetic algorithm. *IEEE Trans. Syst. Man Cybern.* **2004**, *34*, 2303-2316.
16. Ding, Y.; Ping, X.; Hu, M.; Wang, D. Range image segmentation based on randomized Hough transform. *Pattern Recog. Lett.* **2005**, *26*, 2033-2041.
17. Bab-Hadiashar, A.; Gheissari, N. Range image segmentation using surface selection criterion. *IEEE Trans. Image Process* **2006**, *15*, 2006-2018.
18. Lavva, I.; Hameiri, E.; Shimshoni, I. Robust methods for geometric primitive recovery and estimation from range images. *IEEE Trans. Syst. Man Cybern.* **2008**, *38*, 826-845.
19. Dold, C.; Brenner, C. Registration of Terrestrial Laser Scanning Data using Planar Patches and Image Data. In *Proceedings of ISPRS Commission V Symposium "Image Engineering and Vision Metrology"*, Dresden, Germany, September 2006; pp. 78-83.
20. Biosca, J.M.; Lerma, J.L. Unsupervised Robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. *ISPRS J. Photogramm.* **2008**, *63*, 84-98.
21. Wang, M.; Tseng, Y.-H. Incremental segmentation of LIDAR point clouds with an octree-structured voxel space. *Photogramm. Rec.* **2011**, *26*, 32-57.
22. Gorte, B. Planar Feature Extraction in Terrestrial Laser Scans Using Gradient Based Range Image Segmentation. In *Proceedings of ISPRS Workshop on Laser Scanning*, Espoo, Finland, 12–14 September 2007; pp. 173-177.
23. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press, Cambridge, UK, 2004.
24. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Aanal. Machine Intell.* **2002**, *24*, 603-619.
25. Barnea, S.; Filin, S. Segmentation of terrestrial laser scanning data using geometry and image information. *ISPRS J. Photogramm.* **2011**, submitted.

26. Duda, R.O.; Hart, P.E.; Stork D.G. *Pattern Classification*, 2nd ed.; John Wiley & Sons: New York, NY, USA, 2001.
27. Arya, S.; Mount, D.M.; Netanyahu, N.S.; Silverman, R.; Wu, A. An optimal algorithm for approximate nearest neighbor searching. *J. ACM* **1998**, *45*, 891-923.
28. Goldberg, A.V.; Tarjan, R.E. A new approach to the maximum-flow problem. *J. ACM* **1988**, *35*, 921-940.
29. Boykov, Y.; Kolmogorov, V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Machine Intell.* **2004**, *26*, 1124-1137.
30. Bay, H.; Andreas, T.; Tinne, E.; Van Gool, L. SURF: Speeded Up Robust Features. *Comput. Vis. Image Understand.* **2008**, *110*, 346-359.
31. Barnea, S.; Filin, S. Keypoint based autonomous registration of terrestrial laser point-clouds. *ISPRS J. Photogramm.* **2008**, *63*, 19-35.
32. Malacara, D. *Color Vision and Colorimetry: Theory and Applications*, 2nd ed.; SPIE Press: Bellingham, WA, USA, 2011.
33. Smith, A.R. Color gamut transform pairs. *Comput. Graph.* **1978**, *12*, 12-19.