Automatic scene understanding and object identification in point clouds

Egil Bae

Egil Bae, "Automatic scene understanding and object identification in point clouds," Proc. SPIE 11160, Electro-Optical Remote Sensing XIII, 111600M (10 October 2019); doi: 10.1117/12.2534984

Event: SPIE Security + Defence, 2019, Strasbourg, France

Automatic scene understanding and object identification in point clouds

Egil Bae

Norwegian Defence Research Establishment (FFI), P.O. Box 25, 2027 Kjeller, Norway

ABSTRACT

A ladar can acquire a dense set of 3D coordinates of a scene, a so-called point cloud, in sub-second time from ranges of several kilometers. This paper presents algorithms for segmenting a point cloud into meaningful classes of similar objects, and for identifying a specific object within its respective class. The segmentation algorithm incorporates several low level features derived from surface patches of objects from different classes and the interphases between them. On a mathematical level, it partitions the point cloud in a way that optimally balances these considerations by finding the global minimizer to a so-called variational problem over a graph, utilizing recently published results on general high-dimensional data classification. The subsequent recognition step makes use of higher level features for identifying a particular object, represented by a 3D model, among the respective class of segmented objects. It measures similarity of shape between the 3D model and each observed object, considering them as two pieces in a puzzle. The simulated shadow and visibility of the 3D model are measured for consistency with the point cloud shadows. The recognition step is also formulated as an optimization problem and solved by mathematically well-founded techniques. Results demonstrate that point clouds acquired in maritime, urban and rural scenes can be segmented into meaningful object classes and that individual vessels can be identified with a high confidence.

Keywords: segmentation, object recognition, scene analysis, point cloud, ladar

1. INTRODUCTION

Laser detection and ranging (ladar) is an imaging technique for generating detailed three-dimensional (3D) information of a scene. Recent progress in laser and detector technology have made it possible to generate hundreds of thousands 3D points within seconds at several kilometers distance. There are two main variants of the ladar principle. The first, often called flash ladar, transmits a single highly divergent laser pulse with high power that spreads out and reflects from several points in the scene. A 2D array detector concept similar to 2D digital cameras is used to detect the reflected pulses, such that a distance can be calculated for each cell in the 2D array. The second variant, often called line scanning ladar, involves transmitting more convergent laser pulses with lower power that are mechanically scanned over the scene. A line scanning ladar has several advantages, for instance the shape, size and resolution of the area to be imaged can be adjusted with great flexibility. However, the resulting data sets become more complex, in particular since the points will not normally be aligned in a 2D array with a simple, regular neighborhood structure.

This paper presents algorithms for segmenting a point cloud into meaningful classes of similar objects, and for identifying a specific object within its respective class. The algorithms are designed for segmenting complex point clouds acquired by the line scanning principle in maritime, urban and rural scenes, and for identifying rigid human-made objects that may only be partially visible, such as ships or vehicles. The algorithms are also well-suited for point cloud acquired by other ladar principles. The first segmentation step extracts objects of interest from the full point cloud using properties of the object interior and boundary by an algorithm that can be executed rather efficiently. The second "recognition" step proceeds by calculating a recognition confidence measure for each such interesting object by a more computationally demanding algorithm that utilizes the full 3D shape of the object one is searching for, its simulated shadow and optionally certain intrinsic reflectance properties.

Further author information:
E-mail: Egil.Bae@ffi.no, Telephone: +47 63 80 73 99

Electro-Optical Remote Sensing XIII, edited by Gary W. Kamerman, Ove Steinvall, Proceedings of SPIE Vol. 11160, 111600M · © 2019 SPIE · CCC code: 0277-786X/19/$21 · doi: 10.1117/12.2534984

Segmentation and recognition of 3D point clouds have already been the subject of a body of research, a closer review will be given at the start of Section 2 and 3. A usual trick is to first convert the point cloud into a 2D image, where the depth acts as a function of the azimuth and elevation angle, playing the same role as the image intensity. Established techniques for 2D image segmentation can then be used to segment the point cloud based on the depth information. There are, however, a number of limitations to this approach. Unless the point cloud is acquired by a flash ladar, the points must be converted to a regular 2D grid with uniform distances in the azimuth and elevation angles, which leads to loss of information. The depth must be described uniquely in terms of the 2D coordinates, which means that several points cannot share the same azimuth and elevation angle. This is not true in general if the ladar is in motion or if a single pulse reflects from several points in the scene, for instance when hitting vegetation and objects behind. Another common approach is to convert the point cloud to a 3D voxel image and apply image processing techniques in 3D. This avoids issues with non-uniqueness, but leads to information loss and large data sets.

The proposed segmentation approach falls within a newer class of methods that process the 3D points directly on a graph, without first converting them to a regular 2D or 3D grid. Consequently, no information is lost and situations with depth overlap and motion are naturally handled. The algorithm is based on solving a so-called variational problem on the graph, which is a recent concept that is known to give very good solutions to various tasks in pattern recognition. The basic idea is to first formulate the desired solution abstractly as the state of minimum energy of an energy potential. Optimization algorithms can then be designed to find the state of minimum energy on a computer, that typically iterates from some initial state, moving to a state of smaller energy in each step. Until recently, such algorithms have not been considered practical for real time applications due to their high computational demand, and low robustness with respect to initializations. However, recent progress in computational optimization theory have resulted in optimization algorithms that are a lot faster and more robust. Combined with the ongoing progress of miniaturized parallel hardware, real time applications are now getting possible.

Recognition based on matching of 3D shapes has been proposed earlier, e.g. ref.[1] To the best of our knowledge, we are the first to construct an approach for partially visible objects that takes into account both the visible part using 3D shape matching and the invisible part by utilizing the point cloud shadows. The combination of several such cues is especially useful for preventing false recognition of other scene objects. This makes it possible to distinguish very similarly looking objects, such as different variants of the same ship, also when occlusions prevent large parts of the object from being directly observed.

This article is divided into two main sections. Section 2 presents the segmentation step and Section 3 presents the recognition step. Both sections start by explaining the overall concepts in simple words, then go into more mathematical details and present experiments in the end.

2. POINT CLOUD SEGMENTATION AS ENERGY MINIMIZATION OVER A GRAPH

This section is focused on the task of simplifying a 3D point cloud acquired from a ladar by dividing it into two or several regions, which each represent different classes of objects. The purpose in the context of object recognition is to separate out "interesting" objects that share similar characteristics to the object one is searching for. The ability to distinguish interesting objects from the rest of the scene can be greatly enhanced if one is able to find good characteristics of the background as well.

If the scene is sufficiently simple, it suffices to segment the point cloud into a foreground region consisting of the "interesting" objects, and a background region consisting of the remaining objects. This is the case for most maritime scenarios, as will be explained in more detail in Section 2.3. For the more complicated land scenarios, the background is much less uniform. For instance, the ground plane and vegetation are both part of the background, but are geometrically very different from each other. Therefore, it is beneficial to separate the point cloud into several regions, where only one of the regions contains "interesting" objects that are similar to the object one is searching for. There has been proposed several approaches for point cloud segmentation, e.g. ref.[2–7] to name a few. Most of them either interprets the point cloud as an image of 2D pixels or 3D voxels, and applies established image analysis techniques, such of morphological operations. Convolutional neural networks (deep learning) have recently been applied on point clouds, but also require a regular 2D ref.[8–10] or 3D ref.[10, 11] grid, as well as a large amount of training data.

This work interprets the point cloud as a weighted graph, which facilitates direct processing of the raw data without information loss due to preconversion. Graph based approaches for point cloud segmentation have also appeared in e.g. ref.[5–7] in a probabilistic context in the form of "Markov Random Fields". This work is instead based on solving a variational problem, where the desired output is mathematically encoded as the minimum state of an energy potential. Energy minimization methods have over the last two decades shown to give very good results for various tasks in computer vision and pattern recognition, such as image segmentation, restoration, scene reconstruction etc. Combined with recent advances in optimization theory, such methods have now reached a level of maturity where real time applications are possible. Variational approaches on graphs for data classification and point cloud segmentation have recently appeared in ref.[12–15] In particular, our recent work ref.[15] proposed a robust and efficient algorithm for solving certain variational problems on graphs that was shown to give very competitive results on some benchmark high-dimensional data classification problems. The general algorithm could also be applied for segmentation of point clouds acquired in a land environment, and this was demonstrated in a subsection. Whereas ref.[15] was aimed for a technically inclined mathematical audience, we will here give a more comprehensive and pedagogical presentation of the method, specialized for the particular application of point cloud segmentation in a land environment. In addition, it will be shown how the method can be specialized for maritime environments.

2.1 Constructing a graph over the 3D points

Assume that we are given the coordinates of N points in 3D space, obtained by a ladar or another similar imaging technique. The N points will be denoted as $\mathbf{p}^1, \mathbf{p}^2, ..., \mathbf{p}^N$ in the rest of this article. In order to formulate segmentation of the points as a minimization problem, the points are first organized in an undirected graph. A graph is a set of vertices, of which 3D points is a special case, together with a pairwise interconnection between the set of vertices. The set of all vertices is denoted by V. A pair of vertices that are connected is called an edge in the graph. The set of all edges is denoted by E. Two vertices $\mathbf{p}$ and $\mathbf{q}$ are connected if and only if their vertex pair $(\mathbf{p}, \mathbf{q})$ is contained in the set of edges E.

Figure 1 depicts a graph where each vertex is a 2D point. The points are visualized as round balls, whose centers indicate the point coordinates. The edges are visualized as lines connecting two points with each other. In this work, the graph is constructed such that two points are connected if one of them is among the "k-nearest" neighbors of the other. That is, for each point p an edge (p, q) is constructed if q is one of the k closest points of p. We set $k = 20$ in experiments. Constructing a k nearest neighbors graph can be achieved efficiently by exploiting the redundancy between the point coordinates. A common algorithm involves first building a k-d search tree over the points and then looking for nearest neighbors by traversing the branches of the tree, see ref.[16, 17] for details. Both construction of the tree and looking up nearest neighbors can be performed in $O(N\log(N))$ complexity. In practice we found that the whole process took around 0.3 seconds for point clouds with around hundred thousand points in case of 20 nearest neighbors when implemented on CPU. This could be speeded up further by using a GPU ref.[18]

A graph that is constructed based on the k nearest neighbors may consist of several "connected components". A connected component is a subset of the graph where any two vertices (points) are connected by a path of edges, that is, it is possible to travel from one vertex to the other by only moving along the edges. The graph in Figure 1 contains three such connected components. Vertices in two different connected components are not connected by a path of edges. Therefore, constructing the graph provides some basic segmentation of the point cloud, where each connected component can be regarded as a region. Such a segmentation may be meaningful if the object is completely separated in distance from other points in the point cloud. However, this can only rarely be expected to be the case. The next section explains how to obtain a much finer and more meaningful segmentation by minimizing an energy potential over the graph.

Once the graph has been constructed, it is possible to define functions over either the vertices or the edges. A weight function $w(\mathbf{p}, \mathbf{q})$ is defined on the edges $(\mathbf{p}, \mathbf{q}) \in E$ and can be used to measure the similarity between the

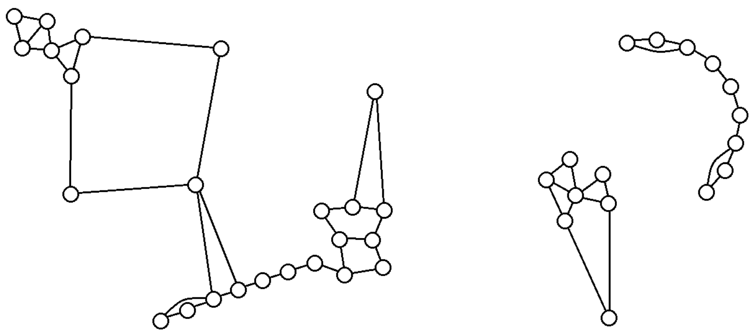


Figure 1: Example of a graph with 2D points as vertices, where each vertex is connected to its two nearest neighbors. Note that some vertices may still be connected to more than two vertices and the graph need not be connected, but may consist of several connected components.

points $\mathbf{p}$ and $\mathbf{q}$. A high value of $w(\mathbf{p}, \mathbf{q})$ indicates that $\mathbf{p}$ and $\mathbf{q}$ are similar and a low value indicates that they are dissimilar. A popular choice of the weight function for general data classification problems is the Gaussian

$$w(\mathbf{p}, \mathbf{q}) = e^{-\frac{d(\mathbf{p},\mathbf{q})^2}{\sigma^2}}, \tag{1}$$

where $d(\mathbf{p}, \mathbf{q})$ is the distance between $\mathbf{p}$ and $\mathbf{q}$, for instance the Euclidean distance. In case each point is a 3D point, the distance between $\mathbf{p}$ and $\mathbf{q}$ is

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}. \tag{2}$$

In this work, we will both make use of the weight function (1) and design specific weight functions for particular applications of point cloud segmentation.

2.2 Construction of general energy potential over graph

The problem of segmenting a point cloud into n regions is mathematically the problem of finding a partition of the set of points V into subsets $V_1, ..., V_n$ that do not overlap with each other. Point cloud segmentation can be formulated as an energy minimization problem by defining an energy potential that assigns an energy value to every possible partition $V_1, ..., V_n$ of V, and where the desired partition is encoded as the one with lowest energy. The exact energy minimization problem will differ depending on the applications, but can be formulated in general form as

$$\min_{\{V_i\}_{i=1}^n} \sum_{i=1}^n \sum_{\mathbf{p} \in V_i} D_i(\mathbf{p}) + \sum_{i=1}^n \sum_{\substack{(\mathbf{p},\mathbf{q}) \in E\,: \\ \mathbf{p} \in V_i,\, \mathbf{q} \notin V_i}} w(\mathbf{p}, \mathbf{q}) \tag{3}$$

$$\text{such that } \cup_{i=1}^n V_i = V\,, \quad V_k \cap V_l = \emptyset\,, \forall k \neq l\,. \tag{4}$$

The first line (3) seeks to minimize the energy potential in terms of the subregions $V_1, ..., V_n$. The functions $D_i(\mathbf{p})$ and $w(\mathbf{p}, \mathbf{q})$ can be custom designed for particular applications of point cloud segmentation. Their meaning will be explained below. The bottom line (4) is just a constraint that imposes there should be no vacuum or overlap between the subregions $V_1, ..., V_n$.

The first term of the energy potential in (3) is used to encourage region homogeneity. It measures how well each point fits with each region individually. The region fitting function $D_i(\mathbf{p})$ is the contribution to the energy potential if the point $\mathbf{p}$ is assigned to the region V_i. It should take a low value if $\mathbf{p}$ matches well with characteristics that are assumed typical of region V_i. Such characteristics could for instance be the point density, orientation of normal vectors, variation in orientation of normal vectors, height, depth etc, and/or different statistical distributions of the signal intensities. Concrete examples will be given in the following sections. Figure 2 illustrates a 2D example where the points are grouped into a region of low point density (yellow),

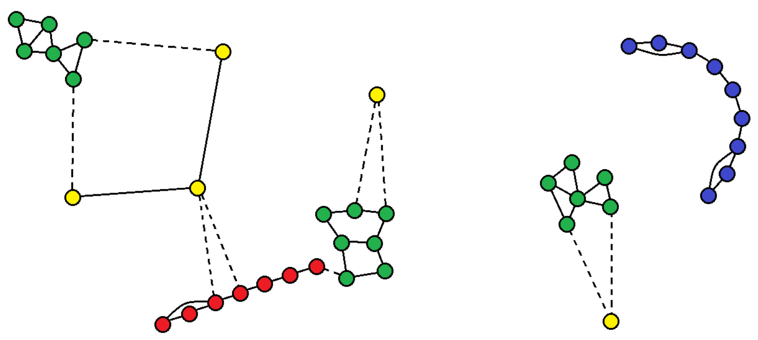


Figure 2: Example of segmentation of graph of 2D points into regions of low density (yellow), high degree of statistical correlation of coordinates between neighboring points (red), medium correlation (blue) and low correlation (green). The image is reproduced from the author's previous work[15] under the terms of the Creative Commons Attribution Licence.

and regions of high (red), medium (blue) and low (green) correlation between the coordinates of each point and its neighboring points. A similar segmentation will be constructed for 3D point clouds, where human-made structures typically have a high degree of correlation between the point coordinates and vegetation have a low degree of correlation.

The last term of (3) is used to encourage both spatial grouping of points to the same region and alignment of the boundary at meaningful locations. It sums together the weights on all edges whose endpoints belong to two different regions. In order to make this term small, the weights on these edges should be small, indicating that it is favourable to separate the two points in the point pair into two different regions. In Figure 2, the edges with ends points in two different regions are visualized as dashed lines. For most of the edges, the distances between the two end points are large, implying that the weights are small if they are constructed by the formula (1).

Our recent work ref.[15] proposed an algorithm for solving general problems of the form (3) that was efficient, provably convergent and could avoid getting stuck in inferior local minima. Within the same framework it was also possible to impose certain constraints on the class sizes. However, this concerned the size of the entire classes and not necessarily individual connected components. In the present work, we found it most practical to incorporate size information in a post-filtering step, where individual connected components are measured for consistency with the expected size of objects in the relevant class.

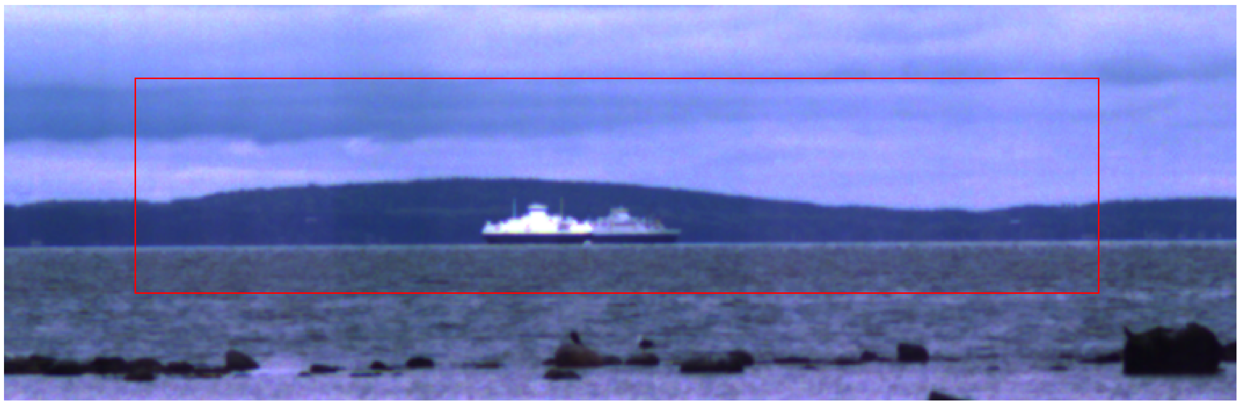
2.3 Basic energy potential in maritime environments

The ocean and sky reflect few of the laser pulses compared to firm objects. If the visibility is clear, the pulses are transmitted with high energy, and the threshold for detecting the reflected incoming pulses is set high, just constructing the graph may result in a segmentation of vessels and other firm objects into one connected component. If the threshold for detecting incoming pulses is set lower, more noisy points may get detected from the surroundings, but more points may also get detected from the object of interest. This is particularly beneficial under bad weather conditions and at far distances.

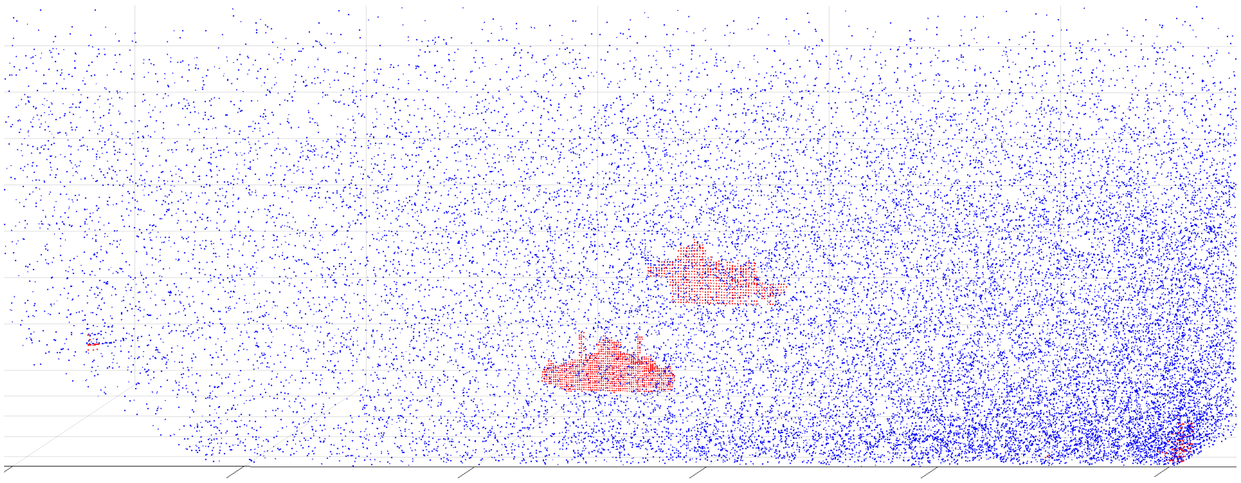
By constructing a special energy potential over the graph, vessels can be separated from the rest of the scene even if a large amount of points are detected from the surroundings. Vessels can be characterized by a relatively higher density of points than the ocean and atmosphere. Most basically, one would therefore like to separate the point cloud into regions of high and low point densities. Once the graph has been constructed, the density $\rho(\mathbf{p})$ at each point $\mathbf{p}$ can be constructed as a function that is inverse proportional to the distance to the neighboring points. One way to define the density is

$$\rho(\mathbf{p}) = \frac{k}{\sum_{\mathbf{q}\in N_k(\mathbf{p})} d(\mathbf{p},\mathbf{q})}|p_1|, \quad \forall \mathbf{p} \in V, \tag{5}$$

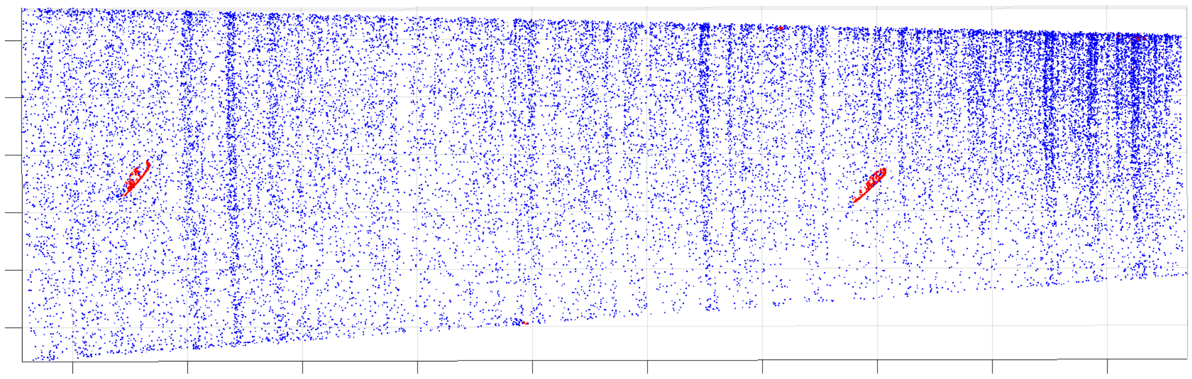
which is the inverse of the average distance to the neighboring points, weighted by the depth coordinate p_1 to make the density independent of the distance to the ladar. $N_k(\mathbf{p})$ is a notation for all the k nearest neighboring



(a) scanning area



(b) Segmentation of point cloud, slightly elevated view point



(c) Segmentation of point cloud, view point from top

Figure 3: Ladar scan in the Oslo fjord of two overlapping ferries. The ferry "Bastø IV" is in the front and "Bastø II" is in the back. (b)-(c): Segmentation of point cloud into region of high point density (red) and low point density (blue).

points of the point $\mathbf{p}$. In order to separate out ships, vessels and other objects in maritime scenes, the point cloud is segmented into regions of high and low point density. This can be achieved by defining the region homogeneity functions in the energy minimization problem (3) as

$$D_1(\mathbf{p}) = \rho(\mathbf{p}) - t\,, \quad D_2(\mathbf{p}) = t - \rho(\mathbf{p})\,, \quad \forall \mathbf{p} \in V, \tag{6}$$

where t is a "threshold level": if $\rho(\mathbf{p})$ is larger than t then $D_2(\mathbf{p})$ takes a lower value than $D_1(\mathbf{p})$ and if $\rho(\mathbf{p})$ is smaller than t then $D_1(\mathbf{p})$ takes a lower value than $D_2(\mathbf{p})$. Minimizing (3) without the first term, would therefore be equivalent to thresholding the density function ρ at level t. However, including the first term will favor a stronger grouping of the 3D points by favoring the same class membership for points with close distance between each other.

Figure 3 shows an example of two ferries recorded from the shore in the Oslo fjord. Red indicates the high density region while blue indicates the low density region. Points detected on the ferries get correctly separated from the surrounding noise, and each ferry may be extracted as a single connected component for further analysis. This example will be continued in Section 3.

2.4 Basic energy potential in land environments

Whereas the segmentation of objects over the ocean is relatively easy and unambiguous, segmentation of objects in point clouds acquired over land is more complicated and open to interpretations for a number of reasons:

- The ground plane generally reflects as large proportion of points as the object itself and may not be clearly distinguished from the object
- The object may be partially occluded and/or hidden among vegetation, clutter or other human-made objects

For scans over land areas, the density of points does not seem to be a good indicator of different regions, since the density seems to be quite uniformly distributed across different objects. Instead, the orientation of the 3D points, indicated by their normal vectors, carry a lot of useful information. This information can be used in two main different ways. First, the direction of the normal vectors themselves can be used as an indicator of certain regions. For instance, the normal vectors of points at the ground plane are expected to point predominantly in the upward direction along the y-axis. Secondly, the extent to which the normal vectors varies between neighboring points can be used to discriminate different regions. Human-made objects, such as vehicles and houses, consist mostly of straight planes or smooth surfaces. They can therefore be characterized by a low variance in the normal vectors between neighboring points. In contrast, vegetation and other natural objects have a more irregular shape. Points sampled from vegetation are expected to have a very random relation to each other. Thus, the normal vectors estimated from the sampling points are expected to vary a lot between neighboring points. Information about the normal vectors was used to classify objects separated from the ground plane into clutter and human-made objects in ref.[19] A crucial difference compared to ref.[19] is that information of the normal vectors is incorporated directly in the segmentation algorithm in this work, before it has been estimated what points constitute an object. This allows one to separate two different objects that appear merged together in the point cloud into different classes.

Information about the normal vectors can be incorporated in the segmentation model (3) both by defining appropriate region fitting functions D_i, $i = 1, ..., n$, and weight functions w. The most basic segmentation of point clouds acquired over land involves separating the point cloud into three regions: 1) the ground plane, 2) vegetation and 3) human-made objects. This is achieved by defining a region fitting function for the region of human-made objects that is small if the normal vectors at the neighboring points does not vary a lot. In contrast, the region fitting function for the vegetation region is small if there is large variation in the normal vectors at the neighboring points. The region fitting function for the ground plane region is small if a number of criteria are satisfied: The normal vectors should point predominately in the upward direction; The "height" of the point, indicated by its y-coordinate, is within some rough local estimate of the height of the ground plane, for instance smaller than the average height of neighboring points; Like human-made structures, the variation of the normal vectors between neighboring points is relatively small. The weight function w is constructed such that points with close distance between each other is grouped together, but in addition it can be constructed to align the region boundaries at favorable locations. For instance, w should take a low value at the transition between the ground plane and objects, which is characterized by sharp convex shaped bends in the scene.

For each point $\mathbf{p} \in V$ one can define an orientation, or normal vector, which is an estimation of the normal vector of the underlying surface the points are sampled from. There are several ways of estimating the normal vectors. In this report, they are estimated from the eigenvectors of the scatter matrix at each point.

For each point $\mathbf{p} \in V$, let $\mathbf{q}^1, ..., \mathbf{q}^k$ denote the set of neighboring points and define for notational convenience $\mathbf{q}^0 = \mathbf{p}$. Define the normalized vectors $\bar{\mathbf{q}}^i = \mathbf{q}^i - \text{mean}(\mathbf{q}^0, \mathbf{q}^1, ..., \mathbf{q}^k)$, for $i = 0, 1, ..., k$ and construct the matrix

$$\mathbf{Y} = [\bar{\mathbf{q}}^0 \bar{\mathbf{q}}^1 \bar{\mathbf{q}}^2 ... \bar{\mathbf{q}}^k] \tag{7}$$

The correlation matrix, or scatter matrix, $\mathbf{Y}\mathbf{Y}^T$, indicates how the points in the neighborhood of $\mathbf{p}$ are oriented in relation to each other. Let $\mathbf{v}^1(\mathbf{p}), \mathbf{v}^2(\mathbf{p}), \mathbf{v}^3(\mathbf{p})$ be the eigenvectors and $\lambda^1(\mathbf{p}), \lambda^2(\mathbf{p}), \lambda^3(\mathbf{p})$ be the eigenvalues of $\mathbf{Y}\mathbf{Y}^T$. The first eigenvector $\mathbf{v}^1(\mathbf{p})$ points in the direction of least variation between the points $\bar{\mathbf{q}}^1, ..., \bar{\mathbf{q}}^k$ and the first eigenvalue $\lambda^1(\mathbf{p})$ indicates the variation along the direction of $\mathbf{v}^1(\mathbf{p})$. $\mathbf{v}^1(\mathbf{p})$ is a discrete estimation of the normal vector at $\mathbf{p}$ and has norm 1. If all the points were laying in a plane, then $\lambda^1(\mathbf{p})$ would be zero and $\mathbf{v}^1(\mathbf{p})$ would be the normal vector of the plane. Therefore $\lambda^1(\mathbf{p})$ indicates the variation in orientation of the points around $\mathbf{p}$. Assume one knows an estimation λ_i of how much the normal vectors vary on average for certain kinds of objects. For instance one can estimate the average value of $\lambda^1(\mathbf{p})$ for vegetation and human-made objects from measurements. The region fitting term for region V_i should be small at the point $\mathbf{p}$ if the value of $\lambda^1(\mathbf{p})$ is close to the estimated value λ_i of region i, and large otherwise. This can be achieved by incorporating the following term in the region fitting function $D_i(\mathbf{p})$:

$$\alpha_i \left| \lambda^1(\mathbf{p}) - \lambda_i \right|^2 \tag{8}$$

It is now described how to design region terms that favour normal vectors that are oriented either parallel with or perpendicular to a certain direction. In the ground plane region for instance, the normal vectors are expected to be oriented predominantly in the upward direction. Let $\mathbf{n}^i$ be the expected orientation of the points in region V_i. The region fitting term $D_i(\mathbf{p})$ for region V_i should be small if the normal vector $\mathbf{v}^1(\mathbf{p})$ is close to the expected normal vector $\mathbf{n}^i$ and big otherwise. This can be achieved by including the following term in $D_i(\mathbf{p})$:

$$\beta_i \left| \mathbf{v}^1(\mathbf{p}) \cdot \mathbf{n}^i(\mathbf{p}) \right|, \tag{9}$$

where $\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + a_3 b_3$ is the inner product operator and β_i is a parameter that should be negative. If β_i is positive, however, the term (9) will instead favour normal vectors that are oriented within the plane perpendicular to $\mathbf{n}^i$. This concerns for instance walls, if $\mathbf{n}^i$ is directed upwards. Using information about the direction of normal vectors to characterize the ground plane may cause the roofs of vehicles and the roofs of flat buildings etc. to get incorrectly assigned to the ground plane region. This can be avoided by a more complex construction of the region terms and the weight function as described below.

By using the standard weight function (1) on the edges in the graph, the energy function (3) will encourage grouping of points with close distance between each other to the same region. This will to some degree prevent points on the roofs of vehicles and buildings from being assigned to the ground plane region if such points do not constitute a large area. However, it is beneficial to modify the weight function such that it is favorable to align the border between regions at locations where the normal vectors change from pointing upwards to pointing outwards, i.e. where the scene is convex shaped. On the contrary, locations where the scene is concave, such as the transition from the side of vehicles to the roof, should be unfavorable for the region boundaries. Such assumptions can be incorporated in the energy function by modifying the weight function w as follows:

$$w(x, y) = e^{-\frac{d(x,y)^2}{\sigma^2} + \gamma \frac{v_3^1(y) - v_3^1(x)}{d(x,y)} \text{SIGN}(y_1 - x_1)} \tag{10}$$

Here $v_1^1(x)$ and $v_3^1(x)$ are the first and third components of the vector $\mathbf{v}^1(x)$, and a coordinate system has been assumed where the positive x_1 axis points outwards from the view direction and the positive x_3 axis points

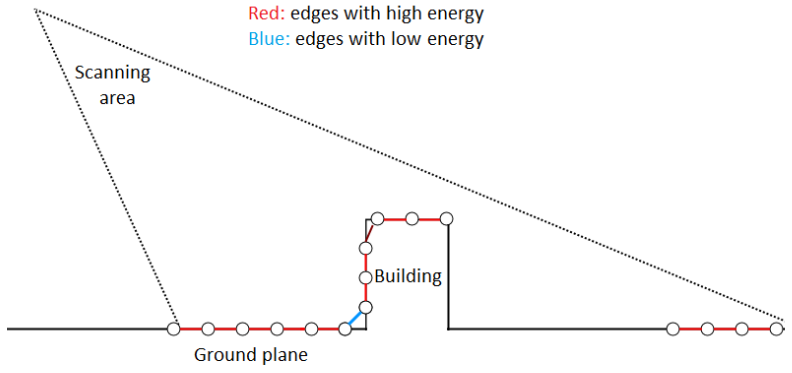


Figure 4: Illustration of construction of graph to separate ground plane from human-made structures, view point from the side. The edges are assigned a low energy at convex parts of the scene, marked in light blue, making it favorable to place the boundary between the regions at such locations. The image is reproduced from the author's previous work[15] under the terms of the Creative Commons Attribution Licence.

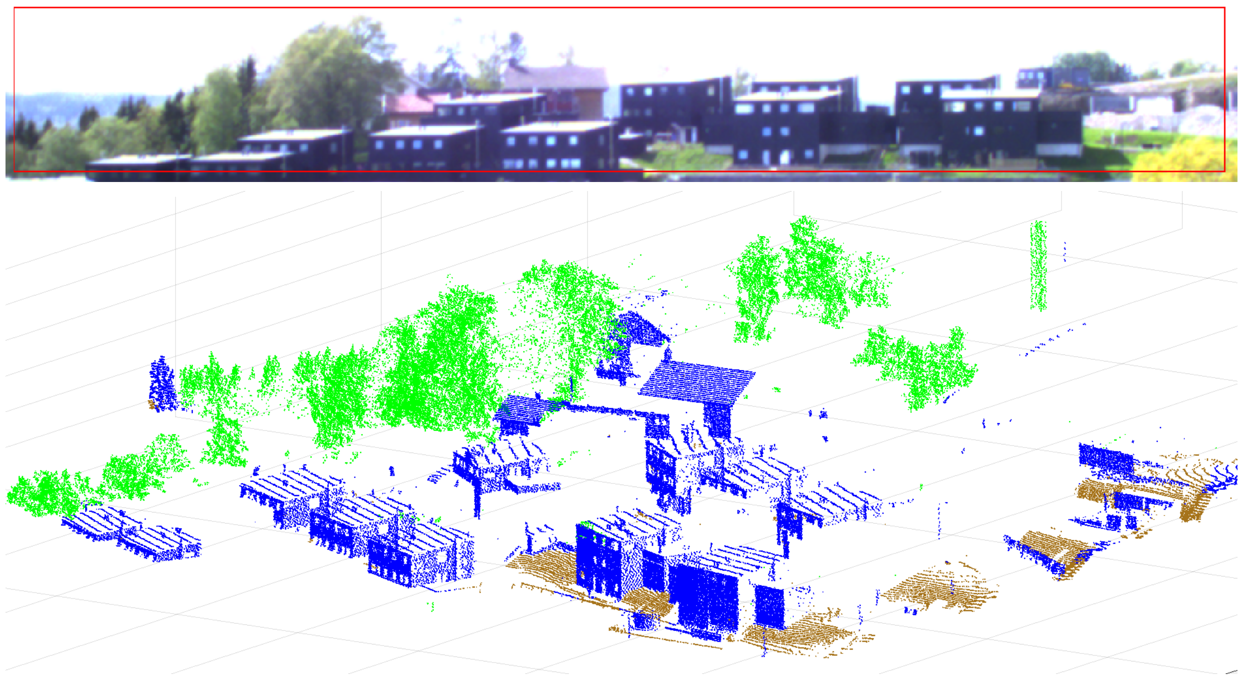


Figure 5: Scan at Kjeller in Norway. The point cloud has been segmented into 3 regions: ground plane (brown), vegetation (green) and human-made objects (blue).

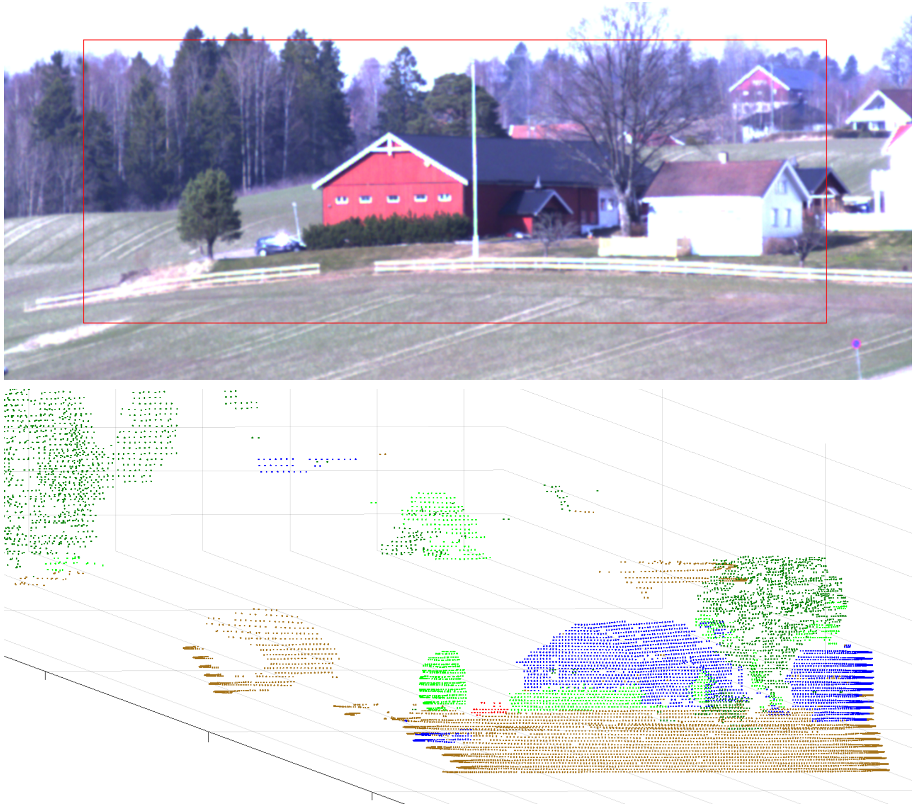


Figure 6: Scan at Kjeller in Norway. The point cloud has been segmented into 4 regions: ground plane (brown), vegetation with leaves (light green) vegetation without leaves (dark green) and human-made objects (blue). Points marked in red indicate all objects within the region of human-made structures approximately the size of a car.

upwards. An illustration is given in Figure 4, where edges at convex parts of the scene are given a low energy value, marked by the color code light blue.

In addition to normal vectors pointing in the upward direction, the ground plane can be characterized by its height, defined by the p_3-coordinate in the point cloud, which is generally lower than the heights of other objects. The height information must be used with care, since the height of the ground plane may vary across the point cloud, and to prevent short objects or the lower part of bigger objects from being assigned to the ground plane region. When used in combination with the other features described above, height information can be useful for obtaining a good segmentation. Assuming a rough estimate of the local height of the ground plane $h^*(\mathbf{p})$ at the point $\mathbf{p}$ is known, the fidelity term (3) can be modified to take into account both normal vectors and height by incorporating the following terms in $D_i(\mathbf{p})$

$$\beta_i \, \mathbf{v}^1(\mathbf{p}) \cdot \mathbf{n}^i(\mathbf{p}) + \theta_i \left(p_3 - h^*(\mathbf{p})\right), \tag{11}$$

where the first term incorporates information about the normal vectors and the last term incorporates information about the height. The last term is small if the height coordinate p_3 is small in relation to the estimated height $h^*(\mathbf{p})$. The parameter θ_i controls to which extent the height plays a role in characterizing the ground plane in relation to the normal vectors. In practice, we set it to a relatively low positive value compared to $|\beta_i|$, such that the normal vectors play the predominant role. There are several ways of computing an estimate of the height $h^*(\mathbf{p})$. The article ref.[19] described an approach based on morphological operations on a histogram of the heights as a function of the depth. In this report, $h^*(\mathbf{p})$ is simply computed as the average of the height coordinates of all the neighboring points of $\mathbf{p}$. Like human-made structures, the ground plane is also distinguished by relatively low variation in the normal vectors between neighboring points. Assuming λ_i is the expected variation of the normal vectors, the complete fitting term for the ground plane region can be constructed as

$$D_i(\mathbf{p}) = \alpha_i \left|\lambda^1(\mathbf{p}) - \lambda_i\right|^2 + \beta_i \, \mathbf{v}^1(\mathbf{p}) \cdot \mathbf{n}^i(\mathbf{p}) + \theta_i \left(p_3 - h^*(\mathbf{p})\right) + \zeta_i \,, \quad \forall \mathbf{p} \in V. \tag{12}$$

The parameters α_i, β_i and γ_i control how large role each feature plays in relation to the others, and the parameter ζ_i is used for calibration with other classes.

The region terms for the other classes are constructed in the same way (12), but with different values of the parameters with index i. For instance β_i can be set to zero or nearly zero for the vegetation region, since the direction of their normal vectors are arbitrary (there is a slight bias towards the horizontal plane in our experience). The parameters should be tuned, or learned from multiple examples, for each particular ladar instrument to achieve optimal results.

2.4.1 Experiments

Some illustrative experiments are shown in Figure 5 and 6. The ladar has scanned the surrounding scenery at Kjeller near Oslo in various directions. Some ordinary photographs of the scenes are shown in subfigures (a), where the red rectangles depict the area that has been scanned by the ladar. In Figure 5 the point clouds have been segmented into three different regions as described above and the results are visualized by brown color for points assigned to the ground plane region, green color for points assigned to the vegetation region and blue color for points assigned to the region of human-made objects. These scenes are particularly challenging because the ground plane tilts and its height varies a lot across the scene due to the hilly landscape. The top of buildings get correctly assigned to the region of human-made objects, due to their concave transition from the walls and due to their locally large height compared to nearby points.

Figure 6 is an interesting example for many reasons. Many different kinds of vegetation are present in the scene, both leaveless trees, bushes and trees with thick leaves. A car is placed among the bushes and next to a tree and building. Trees without leaves have an even more irregular structure than trees with leaves, therefore their normal vectors tend to vary more. We have used two different vegetation regions, where the expected normal variation λ_i for the region of leaveless vegetation is larger than for the region of vegetation with leaves. In Figure 6, the region of leaveless vegetation is visualized by dark green and the region with leaves is visualized by light green. As before, the ground plane and human-made objects are visualized by brown and blue respectively. On this example, all "car-sized" objects within the region of human-made objects have been marked by red points. This makes it possible to distinguish the car located among the bushes and tree into one single connected component.

3. RECOGNITION OF PARTIALLY VISIBLE OBJECTS IN POINT CLOUDS

The second "recognition"/"identification" step proceeds by calculating a recognition confidence measure for relevant objects extracted by the segmentation step by a computationally more demanding algorithms. Earlier work such as ref.[1, 20, 21] have proposed to recognize objects in point clouds based on how well they match with some 3D model. In case the object is only partially visible we have found this approach to be insufficient, because arbitrary scene objects may too easily match partially with the 3D model and thus get falsely recognized. We have instead designed a recognition confidence measure that deals naturally with occluded objects by taking into account both the visible part using 3D shape matching and the invisible part by utilizing the point cloud shadows. Utilization of the shadows in ladar point clouds has been a surprisingly scarce topic in previous literature as far as we are aware. Ref.[22] analysed shadows in ladar point cloud from a physical perspective and demonstrated how they can aid in manual object detection, classification and recognition. In ref.[23, 24] the point cloud shadows were used for automatic object detection. The recognition confidence measure in this work is based on several criteria:

(I) How closely can the observed object points be aligned with a 3D model of the object of interest? To answer this question while also accounting for objects that may only be partially visible, the algorithm rotates and translates the observed object in such a way that the average distance between each observed point and its closest corresponding point on the 3D model becomes as small as possible. Results are depicted in subplots (a)-(d) of Figures 8 and 9, where observed points are marked in blue and model points are marked in yellow. The mean point distance is indicated by C_1 in the rest of this article, and is shown below each figure.

(II) How well does the 3D model match with the point cloud shadows? Assuming the object of interest is constructed of non-transparent material, points are not expected to appear behind it along the line of sight. This criterion is useful for ruling out other scene objects, when searching for an object that may be partially occluded, or partially non-reflective. In Figures 8 and 9 (e)-(h), dark and red indicate the total projected area of the simulated shadow of the 3D model from the view point of the ladar. Red indicates points that are observed within the shadow, violating the non-transparency assumption. The fraction of the total expected shadow area containing such red violating points is indicated by C_2 in the rest of this article, and is shown below each figure.

(III) How well does the distribution of detected points over the projected area of the 3D model fit with expectations? Occluded parts of the area are not expected to reflect pulses. Over visible parts of the projected area, the likelyhood of absence of detected points may be estimated rather well from the incidence angle of the laser beam with the surface of the 3D model, the distance and the strength of other nearby return pulses. Some knowledge about reflectance properties of the object surface would of course make it possible to tune this criterion.

(IV) How many object points are observed? The confidence measure is weighted by a function that decreases (logarithmically) as the number of observed points decreases. This factor counteracts the fact that objects with few points can more easily match with the 3D model by coincidence.

Criterion II is most useful in land environments where objects are usually imaged against a firm background such that their shadows become visible. However, in an open sea environment, the scarser set of "noisy" reflection points in the atmosphere and water also provides a background that actually benefits the recognition algorithm.

3.1 Mathematical details

The confidence that some observed object point cloud O is measured of some object of interest M can be defined by combining criterion I-IV in the "probability" function

$$P(O) = \mathrm{nlog}_t(|O|)\exp\big(-\alpha\, C_1 - \beta\, C_2 - \gamma\, C_3\big), \tag{13}$$

where C_1, C_2 and C_3 are indicators from criterion I,II and III that should be small if O are measured of M. $\mathrm{nlog}_t(|O|)$ is a normalized logarithmic function in the number observed object points $|O|$ that implements criterion IV. The parameters α, β and γ should be tuned, or learned from examples, for each particular ladar instrument in order for P to closely reflect a true probability. Details on how to compute C_1, C_2, C_3 and nlog_t are provided below.

Assume the observed object O contains K points, which are denoted $\mathbf{p}^1, \mathbf{p}^2, ..., \mathbf{p}^K$, and the 3D model contains m points denoted $\mathbf{q}^1, \mathbf{q}^2, ..., \mathbf{q}^m$. The mean point distance in criterion I can be expressed as finding a rotation matrix $\mathbf{R}$ and translation vector $\mathbf{T}$ that minimizes

$$C_1 = \Big\{\frac{1}{K} \min_{\mathbf{R}\in\mathbb{R}^{3\,x\,3}\ \mathrm{s.t.}\mathbf{R}=\mathbf{R}^T, \mathbf{T}\in\mathbb{R}^3} \sum_{i=1}^{K} \min_{j\in 1,\dots,m} ||(\mathbf{R}\,\mathbf{p}^i + \mathbf{T}) - \mathbf{q}^j||^2\Big\} \tag{14}$$

The "Iterative Closest Point" (ICP) algorithm[20, 21] is used to solve the minimization problem (14). The algorithm searches for the nearest local minimum of (14) from some initial alignment of the two sets of points. It consists of two steps that are alternated iteratively until convergence, where the observed points are rotated and translated incrementally every iteration. The first step searches for the nearest model point for each observed point at the current iteration. The second step rotates and translates the observed points in such a way that the least squares distance to the preselected model points is minimized. The observed object is initialized by aligning the axes along the boundaries of the smallest area bounding box surrounding the set of points.[19, 25]

Criterion II can be calculated by comparing the distance map of the point cloud as a function of the azimuth (az) and elevation (el) angles $D_p(\mathrm{az}, \mathrm{el})$ with the distance map of the 3D model $D_q(\mathrm{az}, \mathrm{el})$ after being inserted into the scene by inverting the transformation from criterion I ($\mathbf{q} \leftarrow \mathbf{R}^{-1}\mathbf{q} - \mathbf{R}^{-1}\mathbf{T}$). If $D_p(\mathrm{az}, \mathrm{el})$ is bigger than $D_q(\mathrm{az}, \mathrm{el})$ for some coordinate az, el, that is, if a point is observed behind the 3D model along the line of sight, criterion II is violated at that point. Construct the indicator functions

$$I_{D_p > D_q}(\mathrm{az}, \mathrm{el}) \begin{pmatrix} 1 & \text{if} & D_p(\mathrm{az}, \mathrm{el}) > D_q(\mathrm{az}, \mathrm{el}) + \epsilon \\ 0 & \text{else} & \end{pmatrix}, \quad I_{D_q > 0}(\mathrm{az}, \mathrm{el}) \begin{pmatrix} 1 & \text{if} & D_q(\mathrm{az}, \mathrm{el}) > 0 \\ 0 & \text{else} & \end{pmatrix},$$

where ϵ is some small number. Criterion II measures the number of violating points over the total number of points and can be expressed as

$$C_2 = \sum_{\mathrm{az}, \mathrm{el}} \frac{I_{D_p > D_q}(\mathrm{az}, \mathrm{el})}{I_{D_q > 0}(\mathrm{az}, \mathrm{el})}. \tag{15}$$

Criterion III is opposite of II, it basically says that if some part of the object is not observed along the line of sight, then one expects to either observe some occluding object in front, i.e. $D_p(\mathrm{az}, \mathrm{el}) < D_q(\mathrm{az}, \mathrm{el})$, or that the corresponding surface of the object reflects poorly. The relevant points are marked by the following indicator function

$$I_{D_p = 0\,,\, D_q\; = 0}(\mathrm{az}, \mathrm{el}) = \begin{pmatrix} 1 & \text{if} & D_p(\mathrm{az}, \mathrm{el}) = 0 \text{ and } D_q(\mathrm{az}, \mathrm{el}) \; = 0 \\ 0 & \text{else} & \end{pmatrix}.$$

Although different material reflects light differently, as a general rule the power of the return pulses decay proportionally to the cosine of the incidence angle with the object surface. The "penalty" for absence of an observed point can therefore be gradually decreased proportionally with the cosine of the incidence angle:

$$C_3 = \sum_{\mathrm{az}, \mathrm{el}} \frac{I_{D_p = 0\,,\, D_q\; = 0}(\mathrm{az}, \mathrm{el}) \left| \mathbf{n}_\ell(\mathrm{az}, \mathrm{el}) \cdot \mathbf{n}_q(\mathrm{az}, \mathrm{el}) \right|}{I_{D_q > 0}(\mathrm{az}, \mathrm{el})}, \tag{16}$$

Here $\mathbf{n}_\ell(\mathrm{az}, \mathrm{el})$ is the unit vector indicating the light direction and $\mathbf{n}_q(\mathrm{az}, \mathrm{el})$ is the expected normal vector of the object surface, estimated by inserting the 3D model in the scene.

$\mathrm{nlog}_t(|O|)$ suppresses the recognition score proportionally to $\log(|O|)$ if less that a certain threshold t of points are observed

$$\mathrm{nlog}_t(|O|) = \begin{pmatrix} \frac{\log(|O|)}{\log(t)} & \text{if} & D(|O|) > n_t \\ 1 & \text{else} & \end{pmatrix}.$$

3.2 Experiments

The algorithm is demonstrated on a dataset of ferries traveling across the Oslo fjord in Norway between the towns of Moss and Horten that were recorded from the shore in 2014. At this time, 5 different ferries that only differ slightly in sizes and shapes were in operation. The ferries are named Bastø I, Bastø II, Bastø III, Bastø IV and Bastø V and some photographs and details about their lengths are provided in Figure 7. Bastø I and II are sister ships with and can therefore not be distinguished based on their shapes. The figure also shows 3D models of the ferries that have been constructed by combining ladar scans from different view angels.

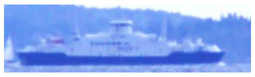
Figure 8 and 9 present the result of the recognition step on two of the most difficult data sets. Figure 8 is a continuation of the segmentation example shown earlier in Figure 3 where two of the ferries, Bastø II and Bastø IV, are crossing each other such that Bastø II in the background is only partially visible. The two ferries are still clearly distinguished in the point cloud because of their different distances. Figure 8 illustrates the 3D matching step, where the observed points, marked in blue, have been optimally aligned with the points on the 3D model, marked in yellow, and the simulated shadow of the 3D model is measured for consistency with the point cloud shadows. Even though Bastø II in the background is only partially visible, the algorithm correctly leads to the best match for Bastø I/II ($P = 0.972$) followed by Bastø III ($P = 0.763$). The ship in the front is correctly recognized as Bastø IV ($P = 0.989$), with Bastø V giving the second best match ($P = 0.786$). Figure 9 presents a case with even more overlap, where only about 1/3 of the ship in the background is visible. The

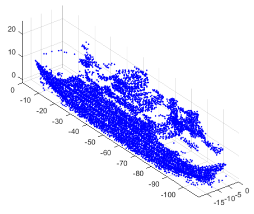
Bastø I, length: 109 m

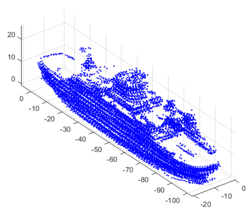
Bastø II, length: 109 m

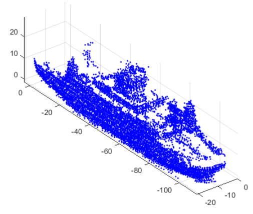
Bastø III, length: 116 m

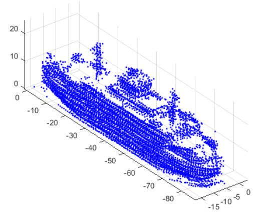
Bastø IV, length: 86 m


Bastø V, length: 92 m


Bastø I, point cloud


Bastø II, point cloud


Bastø III, point cloud


Bastø IV, point cloud

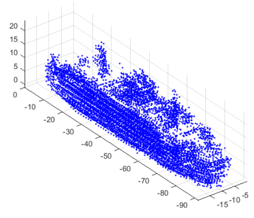
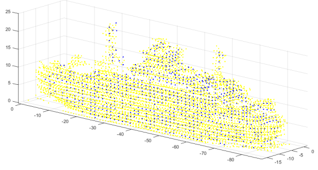
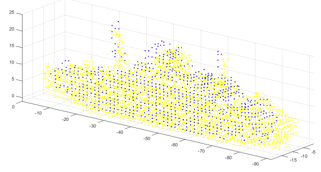
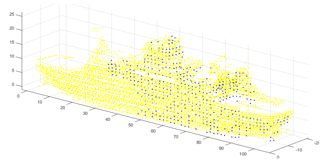
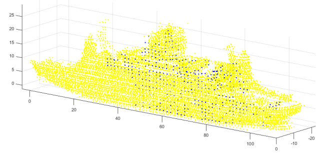

Bastø V, point cloud

Figure 7: The five Bastø ferries and their 3D models obtained by mixing the point clouds acquired by ladar scans from different view points.


(a) Match Bastø IV (best), average point distance: 0.64 m

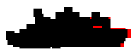

(b) Match Bastø V (2nd best), mean point distance: 1.18 m


(c) Match Bastø I/II (best), mean point distance: 0.64 m


(d) Match Bastø III (2nd best), mean point distance: 0.85 m

(e) Bastø IV shadow inconsistency (C_2): 0.009

(f) Bastø V shadow inconsistency (C_2): 0.100


(g) Bastø I/II shadow inconsistency (C_2): 0.026

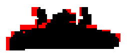
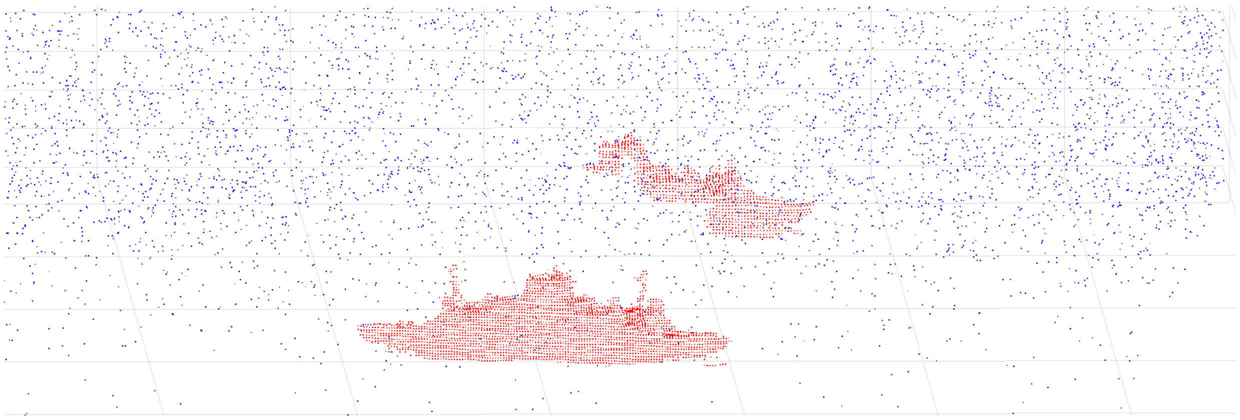
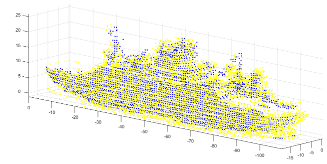

(h) Bastø V shadow inconsistency (C_2): 0.120

Figure 8: Continuation of segmentation experiment shown in Figure 3. Each ferry segmented from the point cloud is matched against the 5 Bastø ferry 3D models. The best and second best match for the frontmost ferry (Bastø IV) is shown in (a)-(b) and for the partially visible backmost ferry (Bastø II) in (c)-(d). Criterion II, the point cloud shadow inconsistency, is shown in (e)-(h) in the same order.

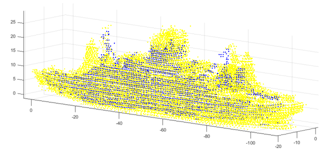
(a) scanning area



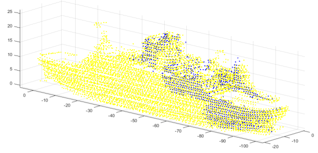
(b) Segmentation of point cloud, slightly elevated view point



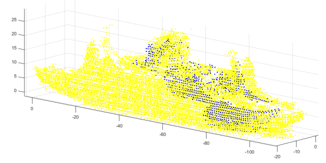
(c) Match Bastø I/II (best): mean point distance (C_1): 0.59 m



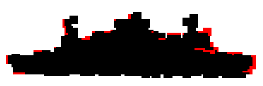
(d) Match Bastø III (2nd best): mean point distance (C_1): 0.97 m



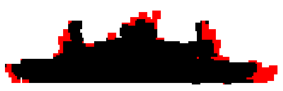
(e) Match Bastø I/II (best): mean point distance (C_1): 0.58 m



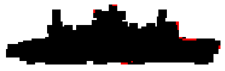
(f) Match Bastø III (2nd best): mean point distance (C_1): 1.16 m



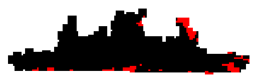
(g) Bastø I/II shadow inconsistency (C_2): 0.040



(h) Bastø III shadow inconsistency (C_2): 0.136



(i) Bastø I/II shadow inconsistency (C_2): 0.011



(j) Bastø III shadow inconsistency (C_2): 0.062

Figure 9: Ladar scan in the Oslo Fjord. The ferry Bastø I is in front and Bastø II is in the back. (b) Segmentation of point cloud into region of high point density (red) and low point density (blue). Best and second best 3D matches (criterion I) of the frontmost ferry (c)-(d) and backmost ferry (e)-(f). Criterion II, the point cloud shadow inconsistency, is shown in (g)-(h).

algorithm is yet able to recognize the backmost ship as Bastø I/II ($P = 0.968$), with Bastø III resulting in the second best match ($P = 0.743$). The correct ferry gets consistently a significantly better recognition value then other ferries in even the most challenging cases. Observe also that the values of C_1 cluster around 0.55-0.65 for the correct ferry and is significantly higher for the others. It can therefore rightfully be said that the ferries get identified with high confidence.

4. CONCLUSIONS

This paper has presented algorithms for segmenting a point cloud acquired by a ladar into several meaningful classes of objects and for recognizing a particular rigid object within its respective class. The algorithms were designed to preserve information by processing the raw data directly without relying on a preconversion step. The algorithms exploited a novel hierarchy of features, starting with low level properties of the interior and boundary of objects in the segmentation step, before taking into account information of the full 3D shape and point cloud shadow map in the recognition step. The segmentation step utilized a recent algorithm for solving certain variational problems on graphs, which has proven to give high quality of results in other data classification applications. Its application to point cloud segmentation in a land environment was further elaborated in this work, and a novel application to maritime scenarios was described. The recognition step optimized a recognition confidence measure that dealt naturally with occluded objects by taking into account both the visible part using 3D shape matching and the invisible part by utilizing the point cloud shadows. Experiments demonstrated that land scenes could be segmented into regions of vegetation, ground, buildings and vehicles. Ships in maritime scenes could be accurately segmented from the surroundings and be identified within a class of very similar ships.

REFERENCES

[1] Armbruster, W. and Hammer, M., "Maritime target identification in flash-ladar imagery," in [*Proc. SPIE 8391, Automatic Target Recognition XXII, 83910C*], (2012).

[2] Jiang, X. and Bunke, H., "Fast segmentation of range images into planar regions by scan line grouping," *Machine Vision and Applications* **7**, 115–122 (1994).

[3] Armbruster, W., [*Model-based object recognition in range imagery*], Forschungsinstitut für Optronik und Mustererkennung, FOM-Bericht 2004/21 (2004).

[4] Felip, R., Ferrandans, S., Diaz-Caro, J., and Binefa, X., "Target detection in ladar data using robust statistics," in [*Proceedings of SPIE*], 1–11 (2005).

[5] Anguelov, D., Taskar, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., and Ng, A. Y., "Discriminative learning of markov random fields for segmentation of 3d scan data," in [*IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA*], 169–176 (2005).

[6] Triebel, R., Kersting, K., and Burgard, W., "Robust 3d scan point classification using associative markov networks," in [*Proc. of the International Conference on Robotics and Automation(ICRA)*], 2603–2608 (2006).

[7] Golovinskiy, A., Kim, V. G., and Funkhouser, T., "Shape-based recognition of 3D point clouds in urban environments," in [*International Conference on Computer Vision (ICCV)*], 2154–2161 (2009).

[8] Z. Wu, S. S., A. Khosla, F. Y., Zhang, L., Tang, X., and Xiao., J., "3d shapenets: A deep representation for volumetric shapes," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 19121920 (2015).

[9] Maturana, D. and Scherer, S., "Voxnet : A 3d convolutional neural network for real-time object recognition," in [*IEEE/RSJ International Conference on Intelligent Robots and Systems*], (2015).

[10] Qi, C. R., H. Su, M. N., Dai, A., Yan, M., and Guibas, L., "Volumetric and multi-view cnns for object classification on 3d data," in [*IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*], (2016).

[11] Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. G., "Multi-view convolutional neural networks for 3d shape recognition," in [*IEEE International Conference on Computer Vision (ICCV)*], 945 – 953 (2015).

[12] Lezoray, O., Elmoataz, A., and Ta, V.-T., "Nonlocal pdes on graphs for active contours models with applications to image segmentation and data clustering," in [*IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*], 873–876 (2012).

[13] Lozes, F., Elmoataz, A., and Lezoray, O., "Partial difference operators on weighted graphs for image processing on surfaces and point clouds," *IEEE Transactions on Image Processing* **23**, 3896–3909 (2014).

[14] Tenbrinck, D., Lozes, F., and Elmoataz, A., "Solving minimal surface problems on surfaces and point clouds," in [*5th International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*], 601–612 (2015).

[15] Bae, E. and Merkurjev, E., "Convex variational methods on graphs for multiclass segmentation of high-dimensional data and point clouds," *Journal of Mathematical Imaging and Vision* **58**(3), 468–493 (2017).

[16] Bentley, J. L., "Multidimensional binary search trees used for associative searching," *Communications of the ACM* **18**, 509–517 (1975).

[17] Indyk, P., "Chapter 39 : Nearest neighbours in high-dimensional spaces," in [*Handbook of Discrete and Computational Geometry (2nd ed.). CRC Press.*], 1–16 (2004).

[18] Zhou, K., Hou, Q., Wang, R., and Guo, B., "Real-time kd-tree construction on graphics hardware," *ACM Trans. Graph.* **27**(5), 126 (2008).

[19] Palm, H. C., Haavardsholm., T., Ajer, H., and Jensen, C. V., "Extraction and classification of vehicles in ladar imagery," in [*Proc. SPIE8731, Laser Radar Technology and Applications XVIII*], **8731**, 873102–873102–15 (2013).

[20] Chen, Y. and Medioni, G., "Object modeling by registration of multiple range images," in [*Proc. IEEE International Conference on Robotics and Automation*], 2724–2729 (1991).

[21] Besl, P. J. and McKay, H. D., "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(2), 239–256 (1992).

[22] Steinvall, O., Sjokvist, L., and Jonsson, P., "Shadows in laser imaging and mapping," in [*Conference on Electro-Optical Remote Sensing XII Location: Berlin, GERMANY Date: SEP 12-13, 2018*], **10796**, 1–20 (2018).

[23] Kuntimad, G. and Delashmit, W., "Target detection in ladar range images using shadow analysis," in [*Proc. SPIE*], **6214**, 621405 (2006).

[24] Grönwall, C. A., Tolt, G., Chevalier, T., and Larsson, H., "Spatial filtering for detection of partly occluded targets," *Optical Engineering* **50**, 047201 (2011).

[25] Grönwall, C., Gustafsson, F., and Millnert, M., "Ground target recognition using rectangle estimation," *IEEE Transactions on Image Processing* **15**(11), 3400–3408 (2006).