Extraction and classification of vehicles in LADAR imagery

Hans Christian Palm*, Trym Vegard Haavardsholm, Halvor Ajer, Cathrine Vembre Jensen
Norwegian Defence Research Establishment (FFI), P.O. Box 25, N-2027 Kjeller, Norway

ABSTRACT

The work presented in this paper is based on a dataset recorded with an airborne sensor. It comprises targets like M-60, M-47, M-113, bridge layers, tank retrievers, and trucks in various types of scenes.

The background-object segmentation consists of first estimating the ground level everywhere in the scene, and then for each sample simply subtracting the measured height and ground level height. No assumptions concerning flat terrain etc. are made.

Samples with height above ground level higher than a certain threshold are clustered by utilizing a straightforward agglomerative clustering algorithm. Around each cluster the bounding box with minimum volume is determined. Based on these bounding boxes, too small as well as too large clusters can easily be removed.

However, vehicle-sized clutter will not be removed. Clutter detection is based on estimating the normal vector for a plane approximation around each sample. This approach is based on the fact that the surface normals of a vehicle is more "modulo 90º" distributed than clutter.

The aim of the classification has been to classify main battle tanks (MBTs) Two types of algorithms have been studied, one based on Dempster Shafer fusion theory, and one model based.

Our dataset comprises clusters of 269 vehicles (among them 131 MBTs), and 253 clutter objects (i.e. in practice vehicle-sized bushes). The experiments we have carried out show that the segmentation extracts all vehicles, the clutter detection removes 90% of the clutter, and the classification finds more than 95% of the MBTs as well as removes half of the remaining clutter.

Keywords: Outlier rejection, segmentation, detection, clutter rejection, classification, ladar

1. INTRODUCTION

LADAR imagery has shown some great capabilities for automatic target recognition (ATR) [1,2,3,4,5,6,7]. The 3D LADAR images provide a lot more information than conventional 2D images, and this can be really useful in many applications. E.g. the reliability of detecting land vehicles on a relatively long distance is much higher using a 3D point cloud than using a 2D set of pixels. Thus, LADAR imagery has a large potential in a future battlefield.

The most important LADAR data product is a range image that may be seen as an unstructured point cloud in 3D space. Thus, algorithms for analyzing such images can easily be computationally quite heavy. Our aim has been to investigate what can be achieved by using computationally relatively simple algorithms; algorithms which have potential for real-time processing on a portable platform but nevertheless give reliable output. All parts of the processing chain, from pre-processing to classification, have been addressed, and they will be described in section 3. In section 4 we will present results from experiments we have carried out. Finally, we give a summary and conclusion in section 5. However, before we start presenting the algorithms, we will very briefly present the dataset we have used.

*Corresponding author: hans-chr.palm @ffi.no; phone +47 63 80 70 00; fax +47 63 80 72 65

Laser Radar Technology and Applications XVIII, edited by Monte D. Turner, Gary W. Kamerman,
Proc. of SPIE Vol. 8731, 873102 · © 2013 SPIE · CCC code: 0277-786X/13/$18 · doi: 10.1117/12.2015363

2. DATASET

The dataset we have used in the experiments is recorded from the air. The Cartesian coordinate system which is used is not specified, but it seems to be tied to the airborne platform. Its origin is most likely the sensor position, while the positive *x*-axis is pointing in the direction of the nose of the platform, and the positive *z*-axis is pointing downwards[1].

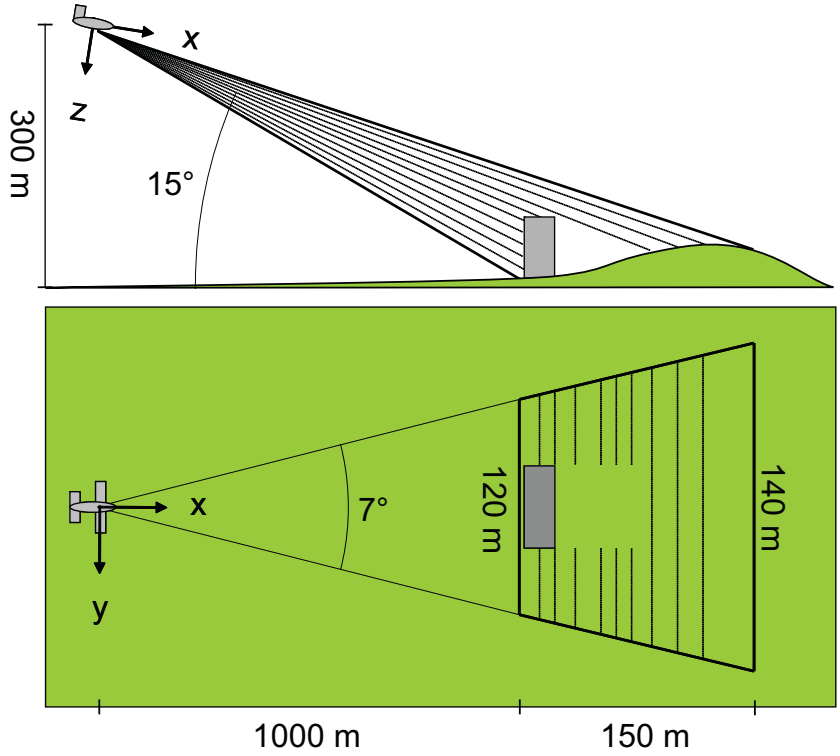

Figure 1 Side- and top-view illustrations of the forward-looking slant-view imaging geometry with typical values from the dataset used in this work. The figure is reproduced from [8].

The resolution seems to be around 0.2 – 0.3 mrad.

The dataset comprises various kinds of vehicles, both military and civilian. Examples of military vehicles are armored personnel carriers (APCs), main battle tanks (MBTs) and self-propelled artillery. SUVs and trucks are examples of civilian vehicles. The vehicles are placed in various scenes like a field, a small riverbed, a camp, and close to an industrial plant.

3. DATA PROCESSING ALGORITHMS

3.1 Overview

We will in this section briefly present the various parts of our procedure. They will be described in detail in subsequent sections. The procedure may be divided into the following parts:

- Outlier rejection. LADAR images may be somewhat noisy, thus outlier samples in range have to be found and filtered. A simple strategy based on median filtering is applied.
- Detection of vegetation and large buildings. The segmentation algorithm is based on local operators for estimating the terrain (e.g. ground level). Therefore it is not feasible for estimating the height of large objects like buil-

[1] Note that the sensor is a scanning device, and it is in motion during a scan. Thus it is not strictly correct to talk about an image set as a set of "flash (instantaneous) images". It should instead be treated as a simplified discussion meant to explain how we understand the coordinate system.

dings and "physical clutter" like tree groves. Hence, such parts of the scene have to be detected and handled properly.

- Segmentation. The aim of this process is to locate possible vehicles. A simple local operator is first applied for estimating the ground level. Next, the height above the ground level is easily determined by subtracting the ground level height from the point-cloud height. All samples 0.5 meter or more above ground level are considered as possible vehicle samples.
- Detection of possible vehicles. The set of possible vehicle samples is clustered, and the length, width, and height of each cluster are estimated. Clusters with size outside certain limits are removed as clutter.
- Clutter removal. Man-made objects are typically more rectangular shaped than natural objects like bushes. Several features, e.g. derived from (local) plane approximations, are computed and used as input to a classifier.
- Classification. MBTs have been considered as targets. Two algorithms have been implemented. One is based on Dempster-Shafer fusion, and combines various simple features to a *belief* measure. The other one, still under development, compares a detected cluster with a 3D model of the target.

3.2 Preprocessing

LADAR images might be somewhat "noisy". The most important component of this noise is isolated pixels whose range differs significantly from the ranges in neighbor pixels. In some cases, this is not only true for isolated pixels, but larger regions as well. There are several sources contributing to this noise. Particles in the air, the reflective properties of the terrain, and general sensor inaccuracy are some possible sources.

But whatever the cause may be, the noise will in general have consequences for the detection of objects. The noise must be handled in some way. Simultaneously, it is important to preserve edges between objects and terrain as well as scene and object structure. An important requirement for the pre-processing filter is therefore to remove the noise *while* preserving the *shape* of the objects.

In the literature, we have only found one filter that has been developed especially for pre-processing of LADAR images, [9]. Unfortunately, this method is very resource-demanding, therefore we have not explored it further.[2] The reason that pre-processing has not been an important issue in LADAR processing literature, is probably because most of the images that have been used are of high resolution and little noise. The need for pre-processing has thus not been very great. In the extent that pre-processing has been used, it has usually only been commented that one has implemented a *median filter*, which indicates that this filter is useful for these kinds of images.

A problem with the median filter is that it *can* remove thin structures, such as a tank gun barrel, or details on a tank turret. This can be mended in several ways. We have chosen to implement an *outlier detector*. An outlier is a measurement which greatly deviates from its expected value. In our datasets, such measurements can be detected by looking at the difference between the filtered and unfiltered values of a pixel. If this difference is "large", the filtered value is chosen. Otherwise the original value is used. The ordinary median filter has been used in the outlier detector.[3] More details can be found in [8].

3.3 Detection of physical clutter and high objects

As previously stated, we do not assume any planar ground. Therefore, the segmentation must be based on local operators. Obviously, these are not feasible for handling large man-made objects as well as large "physical clutter" like tree groves. Building roofs and trees could be considered as ground pixels. Hence the height of vehicles near buildings or trees could be wrongly estimated. In this section we will describe methods for detection of large and high objects as well as physical clutter like trees.

[2] The method is based on both a removal of outliers by a (local) minimization of the *median* of a sum of squared deviations, and a following anisotropic diffusion.

[3] A well-performing approximation to a 2D *nxn* median filter is a set of n+1 1D median filters (and if n=3, a 1D median can easily be computed with some min and max operations).

Detection of physical clutter

We define physical clutter as areas with a relatively large height variation in a (small) local area – larger than what we will find on a vertical surface like a wall. Vegetation areas fall into this definition.[4] Our strategy is based on computing eigenvalues determined from samples in a small neighborhood around the "current" sample followed by some postprocessing.

Eigenvalues computed from a scatter matrix has been used for determining planar surfaces, see e.g. [10]. For each sample (in 3D), the set of samples closer than *r* meters is determined. Then, the corresponding scatter matrix and eigenvalues are in turn computed. Output is the relative difference between the largest and smallest eigenvalue;

$$e_d = \frac{e_3 - e_1}{e_3} \quad (1)$$

where e_1, e_2, e_3 are the eigenvalues in ascending order. If the number of samples available for computing the scatter matrix is too low, 0 is returned.

If the pixels used in the computation for a given pixel are located in a plane (e.g. ground or a vertical wall), the smallest eigenvalue will be quite small, and consequently the relative difference will be high. The less this difference is, the less likely it is that the pixels are located on a plane.

The postprocessing is done in sensor perspective. First, the eigenvalue difference image is thresholded. Next, the thresholded image is defragmented by applying a spoke filter[5], and finally a morphological opening is applied to the defragmented image for removing building edges etc. More details are given in [11].

Detection of high objects

By high objects, we typically mean buildings, i.e. objects with vertical surfaces. This means; 1) in sensor perspective, there will be large differences in range between (parts of) the objects boundary and the neighbor pixels, and 2) there will be a large difference in height between pixels on (part of) the object and pixels in their (nearby) neighborhood (in terrain (top-down) perspective). In our database, utilizing these two features has been sufficient for the detection.

"Leap pixels" (part 1 above) is properly defined in [8], section 3.3, and these are simply pixels for which the difference (in range) to their neighbor pixel in the row above is sufficiently large.

Local height difference (part 2) is computed for each sample (pixel) by picking out all its neighbors, i.e. samples which are closer to "current" sample than *r* meters *in terrain* (i.e. in the *XY*-plane (top-down perspective)). The largest height difference between "current" sample and its neighbors is calculated and returned. In addition maximum height difference, defined as the maximum difference in height for all pixels in the same neighborhood, is computed and returned. This feature will be used later in the postprocessing.

The "leap-pixel image" is thresholded according to [8]. The local height difference image is also thresholded, a threshold of 0.5 meter is applied, and a connected component analysis is performed. All segments in the local height difference image, where the maximum height is sufficiently low, are discarded. This means that small objects are removed. Finally these two images are "ORed".

Next, small gaps are filled by a morphological closing. Then, segments are filled, and boundary pixels on side and upper parts of the segments are removed (often these pixels does not belong to the object).

Now, small noise segments and "particularities" on larger segments are removed by applying a morphological opening. Moreover, we see that physical-clutter areas are also considered as a high object. Such areas are treated as physical-

[4] However, other objects, e.g. small, vehicle-sized objects, may also be detected as physical clutter. If these objects are located on the ground, this does not cause any problem. If however, they are on a roof (or in general on top of a surface above ground plane), e.g. chimneys, it could easily cause severe troubles because their local minima – which in fact are not on the ground – are used for ground estimation. Therefore, some processing for handling these cases should be implemented. This is not done in the work presented here simply due to that no such problems occur in our dataset.

[5] Originally the spoke filter was designed for detection of blobs in infrared images. We, however, are applying it for filling regions without destroying their shape. In [11], Appendix A, it is given a description of how it is used.

clutter areas and thus removed as high objects. This may fragment the segments of high objects. Therefore, we have to investigate each segment ("fragment") in order to check whether it – or parts of it – might be a high object. The image containing possible high objects may be undersegmented. Therefore, different objects are first defined by a clustering process[6]. Next, for a given cluster, if the following two requirements are fulfilled;

1. the maximum height must be sufficiently large,
2. the local height differences in the lowest part of the cluster (in sensor perspective) have to be sufficiently large[7] (otherwise: no vertical surface),

the cluster is assumed to be a high object. More details are given in [11].

3.4 Segmentation

In [8], we investigated several segmentation strategies, both edge-based and region-based. They all estimated the ground level everywhere in the scene, and then computed a height estimate for every sample. The segmentation algorithms were based either on local-plane estimates, morphological operators, filtering operations, region growing, or detection of vertical surfaces. A very simple technique, a two-step averaging procedure, was found to give the overall best performance.

The segmentation consists of two parts. First the ground level is estimated. The estimation is based on 1) samples which are neither "physical clutter" nor "high-object samples", and 2) local minima in "physical-clutter areas". Second, the height above ground level computed, and all samples with height above ground level larger than 0.5 meter computed.

An obviously simple way of computing the (local) ground level for a given sample is by averaging the *z*-values (i.e. heights between sensor and ground points) in a small vicinity of the pixel. This is a computationally very simple task, and if there are no objects in this vicinity and the terrain is plane, the ground level will obviously be correctly estimated. However, these assumptions are in general not fulfilled, and in practice a simple averaging is thus useless. Therefore we propose a two-step averaging. The aim of the first averaging step is to identify possible object samples based on *z*-image (height image) (in addition to areas already detected in the previous step), and the second averaging step is to compute the local ground level based on the *z*-image and possible objects. The algorithm consists of the following steps:

1. Compute the average *z*-image (height image).
2. Compute the absolute difference between the height image and the averaged height image. (These two first steps constitute the absolute values of a highpass filtering of the *z*-image.)
3. Threshold this absolute difference image. All samples larger than the threshold are considered as possible object samples. This binary image is ORed with the masks described in the previous section, and used as a mask in the next step.
4. Compute an average *z*-image based on the pixels (in the *z*-image) which are not considered to be possible object samples.
5. Calculate the (object) height image as the difference between the *z*-image and the two-step averaged *z*-image (i.e. the output from step 4).

The average *z*-images (step 1 and step 4) may be computed in two perspectives; either in a sensor perspective or in a top-projection perspective. In the first one the averaging for a given sample (i.e. here pixel) is computed from the pixels in a $w_x \times w_y$ pixel moving window centered in the pixel, and in the latter one it is based on the pixels closer than a distance r from the given pixel. The window size as well as the maximum distance (r) have to reflect the size of our targets.

The threshold (step 3) does not seem to be critical.

---

[6] [8], section 4.

[7] This is implemented by picking out the 25% lowest samples (in sensor perspective). If the local height difference is larger than 2 meters for at least 60% of these samples, the requirement is assumed fulfilled. The local height difference is defined on the previous page. Observe that this approach assumes that the lowest part of a building's roof is larger than 2 meters. Observe that objects where this requirement does not hold (e.g. a pyramid), might not be considered as a high object.

In general, we are not guaranteed that a height estimate can be assigned to every sample (e.g. the areas of possible high objects may be larger than the (local) areas used for estimation). For each area of non-assigned samples, its contour is determined, a plane is calculated based on these samples, and heights above/below this plane calculated.

3.5 Detection of possible vehicles

Detection in our context means to indicate regions in an image where there is reason to believe that an interesting object is located. The detection will *not* be based on the objects' shape. Since we in this context are interested in finding vehicles, the detection process will in practice be to find objects with a length, width and height within a pre-defined interval, compatible with the vehicle dimensions.

Input to the detection is a set of clusters, which in fact is determined with a standard agglomerative clustering algorithm. The height, length, and width of each cluster have to be determined. The height is simply defined as the height of the highest pixel in the cluster to be considered. The length and width are calculated by first determining the bounding box (in the *XY*-plane) with minimum area. The length and width of the cluster are defined as the length and width of this bounding box.

Before any length and width can be computed, the orientation angle has to be estimated. Based on experiments carried out in [12], this is done by first computing 5 different orientation estimates;

1. The orientation angle minimizing the bounding-box area [13]
2. The proposal of Neulist and Armbruster [14]
3. The vector median [15]
4. The maximum Power Density Estimator [16][8]
5. Most represented angle [12]

Next, the median angle (modulo 90^o) of these estimates is computed and used as a final orientation estimate. The rationale for this procedure is that we haven't yet found one algorithm which work well in all situations. However, we have seen that the majority of the estimates are more or less correct. Therefore it is reasonable to use the median.

Clusters with height, length, or width outside a predefined interval is discarded from any further investigation. The rest is considered as possible targets. The intervals we have used in our experiments are: Height: 1.5-8 meters; Length: 3.0-12 meters; Width: 1.5-5.5 meters. These intervals may seem to be quite large. However, barrels may be elevated, and vehicles may be parked near bushes and small trees. If the interval is to be decreased, more robust (and hence computationally expensive!) methods for estimating height, length, and width have to be applied. Figure 2 shows the "detection shoebox".

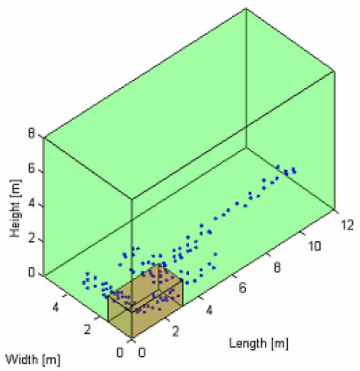

Figure 2 The detection shoebox. Objects that fit within the large box, but not the small box are within acceptable limits. An M-110 target is shown for comparison.

[8] This approach has large similarities with RANSAC [17], and also gives results which are very similar.

3.6 Clutter removal

The detection in the previous section does of course accept all kind of clusters of reasonable size. Thus, objects with different shape than our objects of interest are also accepted. Such objects are reasonable to consider as clutter. Primarily, we will consider natural objects like bushes and small trees as clutter. We have defined main battle tanks (MBT) as our objects of interest, and natural objects as clutter. It is acceptable that man-made vehicles other than MBTs are treated as clutter.

Our approach consists of two parts. First a set of features is extracted from a given cluster, and next it classified as clutter or non-clutter.

Features

Various kinds of features have been extracted:

- Difference between maximum height and height of maximum mode. MBTs are in general rectangularly shaped with a turret above its "roof". The majority of the objects will most likely be "roof-samples". Thus, by estimating the distribution of the object samples, the height with maximum density will most likely be the "roof height". A reasonable large difference between maximum height and height of maximum mode, may indicate that the object could be an MBT.[9]
- Average (local) height difference. The rationale for this feature is simply that the height variations among samples in a local neighborhood of a point is supposed to be larger for a natural clutter object (bush, etc.) than for a man-made object.
- Eigenvalues. If a point **z** is located on a planar surface, then one of the eigenvalues of the corresponding scatter matrix of a set of samples in a neighborhood of **z** will be small (compared to the two other eigenvalues). This is often the case for man-made objects. Points on natural clutter, however, are located on arbitrary surfaces, and all eigenvalues tend to be "large".
- Eigenvectors. Man-made objects are typically rectangularly shaped. Thus, the modes of the set of normal vectors (of local surface approximations) tend to be "modulo-90°" distributed. The normal vectors of the surface approximations of natural clutter are much more uniformly distributed.

Clutter detector

Based on the set of features derived from objects of interest, clutter is considered as anomalies to these objects. We have chosen to apply a cascade detector. For each object class of interest (here: i.e. MBTs), and for each feature, lower and upper thresholds of acceptable values are defined.

An unknown object with a feature vector **x** is considered as clutter if at least one of the features is outside the acceptable thresholds for each object class of interest.[10]

3.7 Classification

Several approaches can be used for cluster classification. One simple example is to combine normalized features $d_1,\ldots,d_n$ in a "fusion function" $d = F(d_1,\ldots,d_n)$ [18, 19]. Suggested functions have been minimum, maximum, mean, inner product (i.e. a linear discriminant function), and majority vote. However, it seems difficult to give any general guidelines for how these functions should be selected. A more sophisticated strategy is to employ fuzzy-logic as used in [6]. In a classifier based on fuzzy-logic, the features are used as input to a set of logical rules. We have previously had very good experiences with applying classifiers based on fuzzy-logic [20]. However, a disadvantage is that they are difficult to maintain. If a new feature is included, the classifier has to be re-designed. Another strategy, that doesn't have this disadvantage, is the Dempster-Shafer rule. In Dempster-Shafer, each feature is investigated in turn, and confidence measures are updated. If employing Dempster-Shafer, opposed to fuzzy-logic, increasing (or decreasing) the number of features is very simple. Therefore, we have chosen to apply this strategy.

[9] This feature is of course useless in case an armored personnel carrier (APC) without a turret is an object of interest.

[10] As might be seen, this implies that the decision boundaries are parallel to the coordinate system of the feature vector. This might not be optimal. The object-acceptance volume may be unnecessary large if the features are strongly correlated. Therefore, we have also tested a detector based on the Mahalanobis distance. However, the results were quite similar.

In practice, the features extracted for use in a Dempster-Shafer classifier (or a "conventional" classifier), give a relatively coarse shape description. Thus, we have also wanted to implement a model-based classifier. Such classifiers shall – in principle – have a larger discrimination ability.

Dempster-Shafer classifier

Dempster and Shafer developed a generalization of a Bayesian scheme [21, 22]. It permits a general uncertainty, which is systematically handled. It is based on so-called proportions which are hypothesis (e.g. M-60, M-47, M-113, M-35) or *combination* of hypothesis (e.g. wheeled or tracked vehicles). A belief value (i.e. a "confidence") is assigned to each proportion. In our application we have defined three proportions:

1. P1: MBT
2. P2: Any other object
3. P3 Either MBT or any other object (!)

Input to the algorithm is a set of features. For each feature and each proportion a probability-mass function has to be defined. As might be guessed, this is a function returning values in the interval [0, 1]. Based on the probability masses, two belief values can be calculated; *support* and *plausibility*. Support is defined as the maximum belief that a given proportion is correct, and plausibility is defined as the maximum possible belief of a given proportion (or 1 minus the negated proportion). (The difference between plausibility and support is called *ignorance*.) The classification process is illustrated in Figure 3.

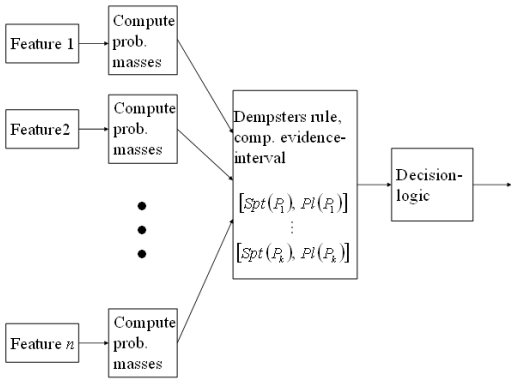

Figure 3 Illustration of the Dempster-Shafer method.

Each feature is treated in turn, i.e. probability masses are calculated, and supports and plausabilities updated. Initially, having no prior information, support for P1 and P2 is "small" and P3 "high". In our case, the decision logic is simply a thresholding; if the support for MBT is sufficiently large (> 0.95), the corresponding cluster is assigned to MBT.

Features

The resolution in the dataset is relatively coarse. Thus, any features which need a high resolution must be discarded. Only relatively simple features are applicable. The applied features are listed below. As can be seen, they all are quite easy to compute.

- Length and width. These are determined from the cluster's bounding box.
- Turret diameter (if any turret). The mean of maximum height and maximum-mode height is calculated. Samples being higher than this mean is assumed to be part of turret, gun etc. A Hough based approach is implemented for fitting a disc to these samples. The diameter of this disc is used as an estimation of the turret diameter. The disc position is utilized in the following features.
- Distances between turret position and cluster border. These four features are simply the distance between the turret (if any) position and each of the bounding box edges.

Probability masses.

Examples of probability masses are shown in Figure 4.

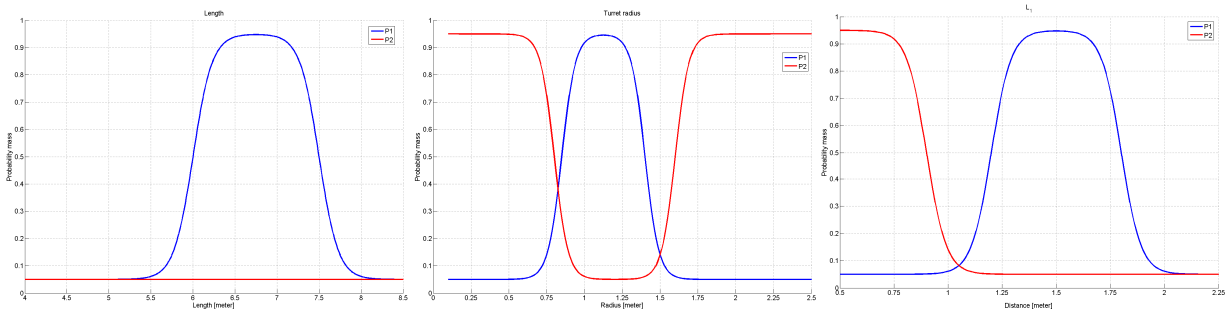

Figure 4 Illustration of probability masses. Blue curves are probability masses for P1 and red curves for P2. The plot to the left is the probability masses for the length (the probability masses for the width are qualitatively similar), in the middle for the turret diameter, and to the right for the distance between the turret position and the nearest cluster border (the probability masses for the other distances are qualitatively similar).

Large probability masses results in an increase in support for the corresponding proportion. In case both probability masses are small, the supports remain constant.

Model-based classifier

The model-based classifier is still under development. So far, only a relatively simple strategy has been implemented. It is based on comparing range distributions, and it comprises the following steps:

1. A 3D model of an object candidate is chosen.
2. A range image of the model is simulated [23] based on imaging geometry between the sensor and the object estimated from the LADAR image.
3. A range histogram with a predefined number of bins is generated for both the cluster samples and the simulated model samples. Each histogram is normalized by subtracting the median range [23].
4. Comparison of range histograms of the given model and each cluster. In the presented examples we measure the Euclidian distance between the model and cluster-histogram vectors, and find objects with a high score of value (1 – Euclidian distance).

4. EXPERIMENTS

Experiments have been carried out for all kind of scenes in our dataset (field, small riverbed, camp, a forest). The results are presented in the subsequent sections.

4.1 Segmentation (and detection)

Images from all available scenes have been used. The average distance between sensor and vehicles have been between 900 and 1300 meter. Three examples are shown in Figure 5. The camp and field scenes (left and right, respectively) are relatively flat. The left third of the forest scene (in the middle) is fairly flat. The trees, however, are located on a small hill. Its top is approximately 10 meter above the flat area to the left, which makes the scene rather challenging.

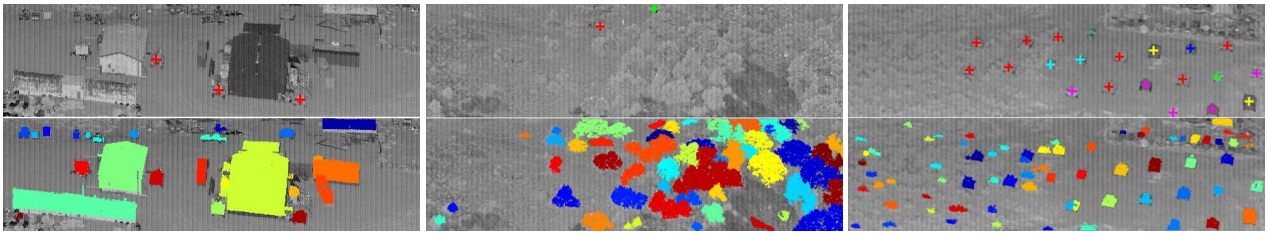

Figure 5 Segmentation results for three different scenes. The upper part shows the position of vehicles (red – M60, red – M35, blue – ZSU-23-4, yellow – M53/55, cyan – M113, magenta – other vehicles (M110A2, M47, bridge layer, tank retriever, M728). The lower part shows clusters larger than 1.5x1.5 meter.

The detection rate depends on the scene. For "easy" scenes (relatively flat scenes) it is more than 95% (typically 98-99%). For more difficult scenes like the forest scene, the detection rate drops down to around 85%, and for even more difficult scenes like scenes where the vehicles are located in a relatively small riverbed, it drops further down – typically to around 70-75%. In both these cases, this is due to difficulties in estimating the ground level.

4.2 Clutter removal

As can be seen from Figure 5, vehicle-sized, natural clutter will of course be extracted. In the field and riverbed scenes, it is usually 30-40 bushes and small trees which are detected. In some of scans of the field scenes, there are also some objects in a scrapyard. We have no ground truth of what kind of objects which are there.

The dataset consists of 105 M-60 clusters, 26 M-47 clusters, 138 clusters of other military vehicles, 36 clusters of objects in the scrapyard, and 253 clutter clusters.

The probability of classifying the various kinds of objects as clutter (natural or man-made) is shown in Figure 6. The probabilities are estimated by applying the leave-one-out procedure [24].

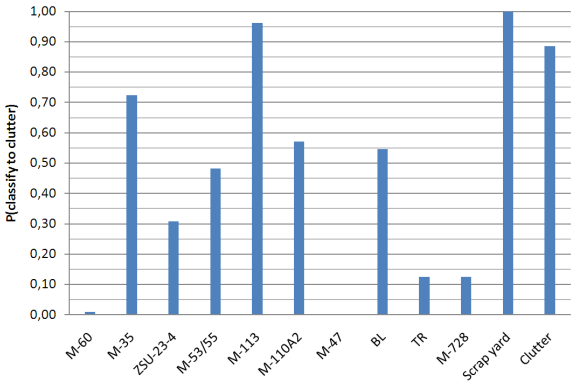

Figure 6 The probability of assigning a cluster to clutter.

Three of 105 M-60 and none of the 26 M-47 were detected as clutter. All these three M60 were parked near physical clutter, and these clutters were included in the object cluster. This is most likely the reason for the mistakes. Concerning M-113s, all but one of them are detected as clutter. This does not surprise as the difference between maximum height and height of maximum mode is used as a feature. For the other vehicle classes, there may be more complex reasons for the detection results. As long as they all are vehicles, the detection results are not very important. However, what is important is the ability to detect clutter, whether it is natural vegetation or various clutter in the scrap yard. We see that around 90% of the clutter is detected.

4.3 Classification

Dempster-Shafer classification

The same dataset as used for the clutter detection is also used for the classification experiments. A question to be asked is how the results from the clutter rejection should be utilized. There are two obvious options. One is simply to say that what is rejected is rejected and thus removed from the dataset. The other one is to use the output from the clutter rejection to initialize the classification. The advantage with the first one is that detected clutter is removed and will not cause

any more trouble. However, the disadvantage is that a misclassified target is forever lost. In the second option, the advantage and disadvantage more or less swaps. The advantage is that a misclassified target can still be detected if the provided evidence is sufficiently good (i.e. if the extracted features are sufficiently close to the expected ones). The disadvantage is that clutter may be misinterpreted as targets.

Figure 7 shows the results for the first options; i.e. the clutter rejection, clutter removal, and a following classification on the rest of the dataset. Figure 8 shows the classification result only (i.e. after clutter removal).

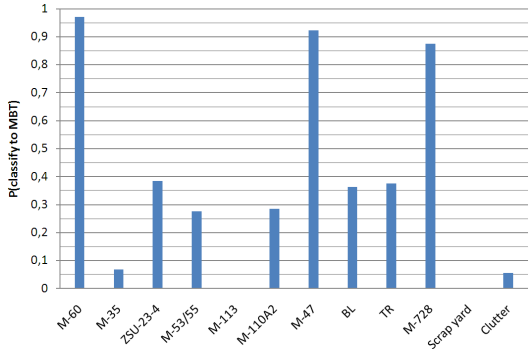

Figure 7 Probability of assigning a cluster as MBT. The processing consists of the clutter removal followed by the Dempster-Shafer classification

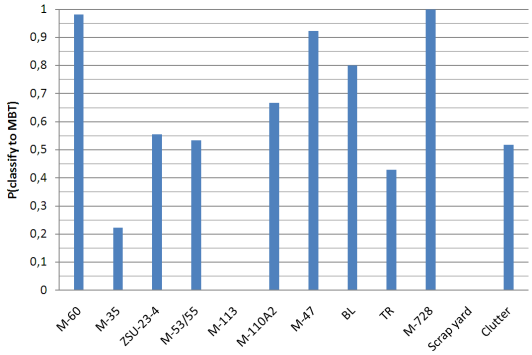

Figure 8 Classification results only. Probability of assigning a cluster as MBT using the dataset were (assumed) clutter is removed.

Figure 9 shows the results for the second option where the clutter rejection is only used for initializing the belief in the classification. For the clutter samples, 0.95 is assigned to proportion 2 (clutter) and 0.05 to proportion 3 (the union of P1 and P2).

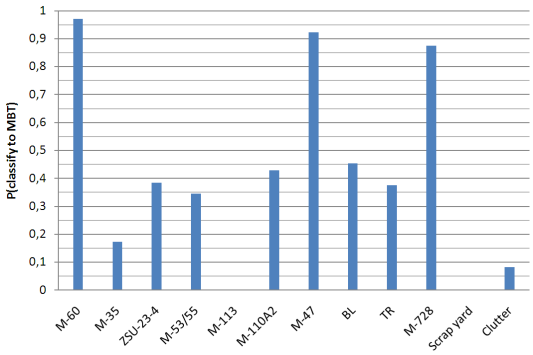

Figure 9 Classification results. Probability of assigning a cluster as MBT. Clutter is not removed; it is only used for initializing the Dempster-Shafer classifier.

For the first option we see that slightly more than 96% of the MBTs (M-60 and M-47) are treated as targets. For all other military vehicles but the M-113, they are classified as MBTs if they are seen from positions where the features derived from the corresponding clusters are sufficiently close to the MBT features. We notice that all M-728 are detected as MBT. That does not surprise because the M-728 is a special utility vehicle based on the M-60 chassis. Nevertheless, using very simple features, very many (in average 70%) of the non-MBT military vehicles are discarded. If a higher rejection rate is desirable, more "detailed" shape-describing features must be utilized (or other techniques like model matching). It is also interesting to see that all objects/vehicles in the scrapyard are treated as clutter. The clutter detection itself is able to detect around 90% of the clutter, and the classification process removes half of the remaining clutter.

For the second option we see that slightly more clutter has been classified as MBTs. This is to be expected. The reason is simply because their size has been sufficiently close to MBT size. The consequence has been that the initial clutter detection as been overruled.

Model-based classification

Figure 10 shows examples of histogram signatures for different vehicles, retrieved from a scan.

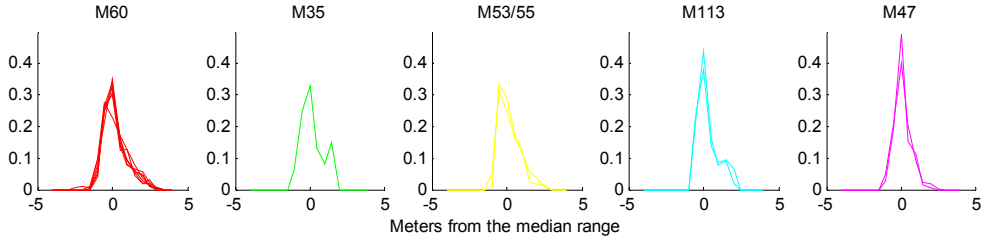

Figure 10 Examples of histogram vehicles for different vehicles.

A basic way to compare a range histogram of an observed object and that of the simulated model object is to simply measure the Euclidean distance between the histograms, as vectors. The object model used in the simulator throughout is an M-60.

Figure 11 shows the scene of our first example, and Figure 12 the corresponding results. This scene has 22 objects: categorized vehicles and detected clutter.

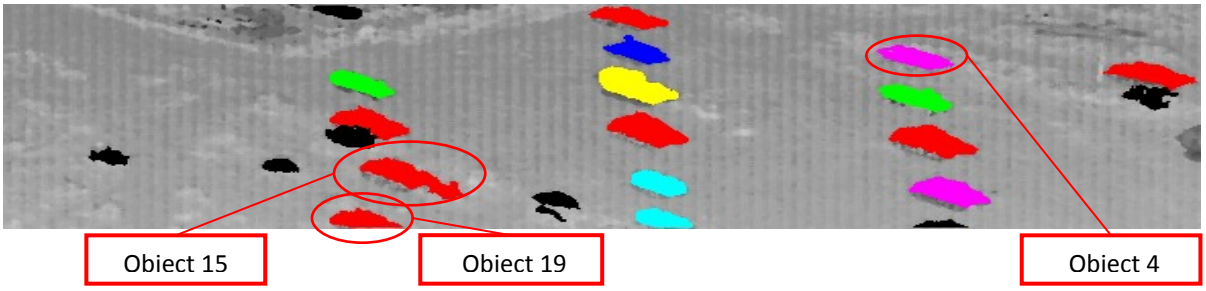

Figure 11 The scene in the first example

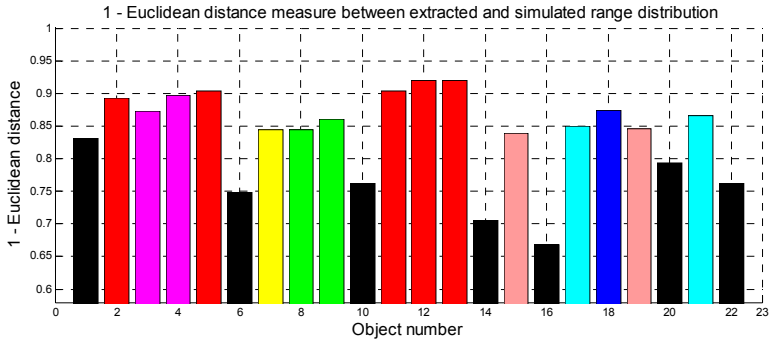

Figure 12 Similarity values between an M-60 model and the clusters in scene.

Objects 2, 5, 11, 12, 13 (red) as well as 15 and 19 are M-60. For a given threshold we single out all the M-60-s except object 15 and 19, as well as object 4, a different vehicle also receiving a high score.

As can be seen, object 15 is an M-60 with clutter attached, and object 19 is an M-60 at the edge of the image, with part of it missing. Thus, it is not unexpected that these receive a lower comparison score with respect to the range distribution of the model vehicle.

The second scene and the corresponding results are shown in Figure 13 and Figure 14 respectively. The scene has 27 objects, categorized vehicles and detected clutter.

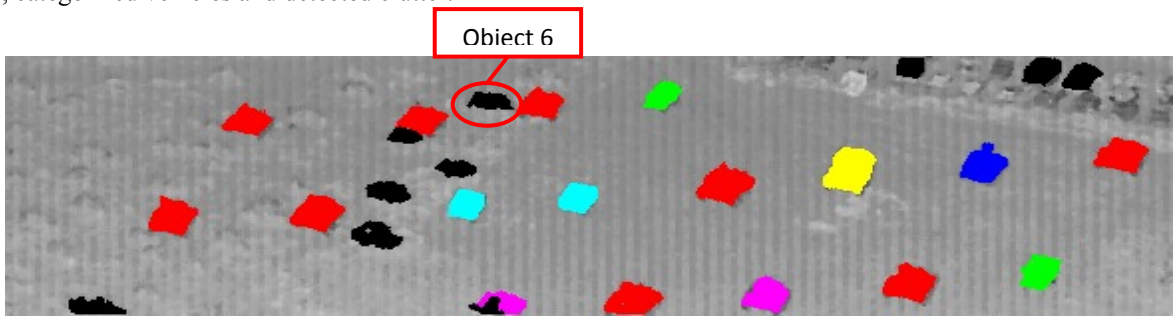

Figure 13 The scene in the second example.

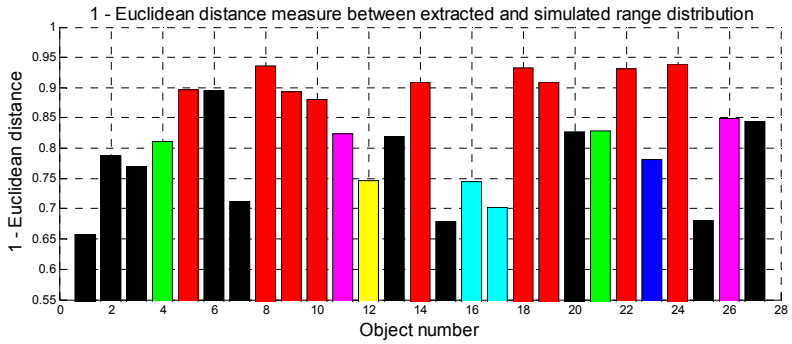

Figure 14 Similarity values between an M-60 model and the clusters in the second scene.

Objects 5, 8, 9, 10, 14, 18, 19, 22 and 24 are M-60. For a given threshold we single out all the M-60-s, as well as object 6, which is vegetation clutter also receiving a high score.

5. SUMMARY AND CONCLUSION

In this paper we have presented our work on a dataset generated with an airborne LADAR sensor. The dataset comprises targets like main battle tanks, armored personnel carriers, bridge layers, tank retrievers, self-propelled artillery, and trucks. Algorithms for all parts in the processing chain have been presented and evaluated.

Segmentation: In flat terrain, we have found that almost all objects are extracted. In difficult scenarios where the ground is far from planar in a neighborhood of the object, the extraction rate drops to around 60%, which still has to be considered as quite good.

Clutter removal: The clutter-removal algorithm seems to be quite effective. Around 90% of the clutter and all objects in the scrapyard are in fact detected. At the same time, only 3 of 131 MBTs are considered as clutter. Many of the other non-interesting vehicles are also considered as clutter. This shows that features discriminate well, and it would be interesting in a future study to investigate more sophisticated detectors than cascade detectors and Mahalanobis-based detectors.

Dempster-Shafer classification: First of all, we found that 96% of the MBTs were classified as MBT, which has to be considered as quite high. Furthermore, an additional 50% reduction (to the clutter removal) of clutter was achieved. Thus, after clutter removal and Dempster-Shafer classification, only 5% of the clutter is left. Considering the other vehicles but M-728 in the scenes, an additional reduction in treating them as MBTs was achieved. This was an interesting

finding because the applied features are relatively simple, and all of the vehicles are (with respect to such features) quite similar; rectangular with a flat "roof" and some structure on this "roof". Considering the M-728, all of these were classified to MBT. This is not a surprise since they are built upon the M-60 chassis.

Model-based classification: The model-based classification is still under development. So far, it seems that even a simple strategy which does not utilize any geometrical connection between cluster and model is able to discriminate between M-60 clusters and clusters from other classes. To further explore the use of range signatures for classification, investigating choices of histogram building methods as well as examining ways of measuring similarity of range histograms suggest ways towards refinement of the method.

So to conclude, we have found that relatively simple algorithms are able to detect and classify vehicles at a distance of more than one kilometer. We haven't so far put any effort in real-time processing, but all algorithms have such a potential.

In an on-going study the model-based classification algorithm is developed further. Topics for this study are both other distance metrics and algorithms for classification in camera perspective as well as algorithms for use in a top-down perspective. These strategies will then be compared with the Dempster-Shafer classifier.

REFERENCES

[1] Wang, H. and Suter, D., "A model-based range image segmentation algorithm using a novel robust estimator," Proc. 3rd International Workshop on Statistical and Computational Theories on Vision, 1-21 (2003).

[2] Stevens, M. R., Snorrason, M., Stouch, D. W. and Amphay S, "Estimating the ground plane in ladar 3-dimentional imagery for target detection," Proc. SPIE 5094, 1-9 (2003)

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

[4] Felip, R. L. I., Ferrandans, S. and Diaz-Caro, J., "Target detection in ladar data using robust statistics," Proc. SPIE 5988, 1-11 (2005).

[5] Chevalier, T., Andersson, P., Grönvall C. and Tolt, G., "Methods for ground target detection and recognition in 3D laser data," FOI-R 2150-SE, Totalförsvarets forskningsinstitut (2006).

[6] Garten, H., Tal, Y., Swirski, Y. and Imber, A., "Recognition of tanks in laser radar (LADAR) images," Proc. SPIE 5613, 166-176 (2004).

[7] Roy, S. and Maheux, J., "Baseline processing pipeline for fast automatic target detection and recognition in airborne 3D ladar imagery," Proc. SPIE 8049, 1-11 (2011).

[8] Palm, H. C., Haavardsholm, T. H. and Ajer, H., "Detection of military objects in ladar images," FFI/RAPPORT-2007/02472, Forsvarets forskningsinstitutt (2007).

[9] Umasuthan, M. and Wallace, A. M., "Outlier removal and discontinuity preserving smoothing of range data," IEE Proceedings of Vision, image and signal processing 143 ,191-200 (1996).

[10] Grönvall, C., Tolt, G., Chevalier, T. and Larsson, H., "Spatial filtering for detection of partly occluded targets," Optical engineering 50(4), (2011).

[11] Palm, H. C., "Detection of vehicle-sized objects in LADAR-images in scenes with large objects or clutter using local operators," FFI/NOTAT-2008/00838, Forsvarets forskningsinstitutt (2008).

[12] Palm, H. C., "Estimation of cluster orientation," FFI/NOTAT-2009/0150, Forsvarets forskningsinstitutt (2009).

[13] Grönvall C., Gustafsson, F. and Millnert, M., "Ground target recognition using rectangle estimation," IEEE Transactions on Image Processing 15(11), 3401-3409 (2006).

[14] Neulist, J. and Armbruster, W., "Segmentation, classification and pose estimation of military vehicles in low resolution laser radar images," Proc. SPIE 5791, 218-225 (2005).

[15] Astola, J., Haavisto, P. and neuvo, Y., "Vector median filters," Proceedings of IEEE 78(5), 648-689 (1990).

[16] Wang, H. and Suter, D., "MDPE: A very robust estimator for model fitting and range image segmentation," International Journal of Computer Vision 59(2), 139-166 (2004).

[17] Fischler, M. A. and Bolles, R. C., "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM 24(6), 381-395 (1981).

[18] Alkoot, F. and Kittler, J., "Experimental evaluation of expert fusion strategies," Pattern Recognition Letters 20, 1361-1369 (1999).

[19] Kuncheva, L. I., "A theoretical study on six classifier fusion strategies," IEEE Transactions on Pattern Analysis and Machine Intelligence 24(2), 282-286 (2001).
[20] Palm, H. C., "Fusion of information from perimeter surveillance sensors (in Norwegian)," FFI/RAPPORT-2004/001721, Forsvarets forskningsinstitutt (2004).
[21] Hall, D., L., [Mathematical techniques in multisensory data fusion], Artech House Inc, Norwood, MA, (1992).
[22] Shafer G, [A mathematical Theory of Evidence], Princeton University Press, Princeton, NJ, (1976).
[23] Haavardsholm, T. V., "A simple Ladar simulator based on ray tracing," FFI/RAPPORT-2009/01481, Forsvarets forskningsinstitutt (2009).
[24] Duda, R. O., Hart, P. E., [Pattern classification and scene analysis], John Wiley and Sons, NY, 75-76 (1973).