Open access • Proceedings Article • DOI:10.1109/IVS.2017.7995854

Semantically aware multilateral filter for depth upsampling in automotive LiDAR point clouds — Source link

Martin Dimitrievski, Peter Veelaert, Wilfried Philips

Institutions: Colorado State University, Ghent University

Published on: 11 Jun 2017 - IEEE Intelligent Vehicles Symposium

Topics: Upsampling, Image processing, Point cloud, Image segmentation and Pixel

Semantically aware multilateral filter for depth upsampling in automotive LiDAR point clouds

Martin Dimitrievski, Peter Veelaert, and Wilfried Philips

Abstract— **We present a novel technique for fast and accurate reconstruction of depth images from 3D point clouds acquired in urban and rural driving environments. Our approach focuses entirely on the sparse distance and reflectance measurements generated by a LiDAR sensor. The main contribution of this paper is a combined segmentation and upsampling technique that preserves the important semantical structure of the scene. Data from the point cloud is segmented and projected onto a virtual camera image where a series of image processing steps are applied in order to reconstruct a fully sampled depth image. We achieve this by means of a multilateral filter that is guided into regions of distinct objects in the segmented point cloud. Thus, the gains of the proposed approach are two-fold: measurement noise in the original data is suppressed and missing depth values are reconstructed to arbitrary resolution. Objective evaluation in an automotive application shows state-of-the-art accuracy of our reconstructed depth images. Finally, we show the qualitative value of our images by training and evaluating a RGBD pedestrian detection system. By reinforcing the RGB pixels with our reconstructed depth values in the learning stage, a significant increase in detection rates can be realized while the model complexity remains comparable to the baseline.**

I. INTRODUCTION

Depth perception as a visual ability allows us to perceive the structure of the world in three dimensions. During driving, the human brain-eye loop is constantly active and adjusts to the ongoing traffic situation. However, the human organs have physiological limits such as accumulation of eye strain, blinding by glare and low light sensitivity, which can all pose difficulties to the inexperienced driver. Analysis of current trends in road traffic accidents confirm the notion that the weakest link and the most unreliable part of a vehicle is indeed the driver [1]. A significant part of the driver's reasoning is performed using prior experience of the situation and context. Using spatio-temporal tracking, the human brain is able to interpolate any missing visual cues that might occur due to occlusions or loss of concentration. This process is difficult to accurately model since it is highly subjective, but in all cases the brain relies on a decent set of sensory inputs over long periods of time. Thus, we suspect that the combination of accurate depth and visual imagery plays an important role in the driving reasoning.

* The work was financially supported by IWT through the Flanders Make ICON project 140647 : "Environmental Modeling for automated Driving and Active Safety (EMDAS)"

In the past decade camera technologies have advanced to a point where we can safely assume that the visible light spectrum can be densely and accurately imaged at low sensor cost. Visual object detection algorithms have also been actively studied and currently there are many, now considered classical [2], applications that can be run in a real-world environment. A recent and comprehensive overview of pedestrian detectors has been presented in [3]. However, since the popularization of deep learning, applications of pedestrian detection in RGB images have received an increase in interest producing new and promising algorithms [4],[5],[6],[7].

Depth information, on the other hand, has been mainly perceived by extracting disparity information using stereo image processing. The evaluation page on the KITTI Stereo 2012 benchmark, [8] lists more than 50 algorithms and a review of the topic is outside the scope of this paper. We note that most of the shortcomings when processing stereo images arise from noise in estimating the disparity which is inversely proportional to depth. Small disparity errors may give large depth errors, especially for distant objects. Authors in the literature have used other depth sensing methods, such as time-of-flight cameras, where they successfully applied up-sampling and de-noising to the raw data. Authors in [9] were able to recover dense, de-noised depth images without the use of a camera reference image. However, time-of-flight cameras have limited operating range, especially in bright light conditions, and as such are not in the immediate focus for autonomous vehicles research. Another viable ranging solution is the Light Detection and Ranging (LiDAR) sensor which scans the environment by shining infra-red laser beams and measuring the reflection delay in order to determine the correct distance of objects. These sensors can operate reliably in outdoor environments and have usable ranges at up to 80m. This performance, however, comes at a price of reduced sampling density and to this date, steeper sensor cost.

The topic of obtaining a dense depth map and interpolation from automotive LiDAR point clouds has been sparsely researched with attempts such as [10] and recently [11]. The former proposed a d-dimensional reformulation of the bi-lateral filter in order to use a calibrated RGB image as guidance for the depth up-sampling. The latter uses interpolation techniques only on data from time-synchronized LiDAR point clouds to produce dense depth maps. We found out, however, that even though these resulting images look appealing to the eye, the actual values around object boundaries are far from their correct values. An adaptation to the bi-lateral filter approach has been suggested [12] where points within each local neighbourhood are clustered and only points belonging to the cluster with the closest distance

to the local point are considered for up-sampling. This technique has promising results, however it uses a very small clustering window which can miss the underlying semantics and the object structure. Another disadvantage is that it only clusters points into two groups (foreground / background) which isn't descriptive enough in situations where several objects lie in the same line of sight and/or objects have small size relative to the sampling resolution.

Our main contribution is a semantically aware algorithm for generation of dense depth maps using only sparse LiDAR point cloud data. We propose a two-step approach by first segmenting the input point cloud into disjoint objects and then applying our novel guided multilateral filtering technique on the depth projection image. We use a reformulated multilateral filter to work on projected sparse images of LiDAR distance and reflectance values, filling-in missing measurements, preserving the object edges and suppressing the measurement noise. Our dense depth maps will later be used to reinforce the RGB input of a state-of-the-art pedestrian detection algorithm. The rest of the paper is organized as follows: in Chapter 2 we will describe how we segment the point cloud how it is projected on a virtual camera viewport. Chapter 3 describes our depth map up-sampling using a novel reformulation of the multilateral filter. The experimental setup and results confirming our hypothesis are presented in Chapter 4 and in Chapter 5 we discuss the fail cases and the opportunities for further improvement.

II. Point Cloud Segmentation

The data coming from the HDL-64E automotive ranging sensor from the company VelodyneLidar is formed by a fixed pattern of 64 rotating laser beams. The sampling rate of each laser can be programmed and generally is in the range of 2000 samples per full rotation. This rotational nature of the sensor head results in three dimensional point cloud which represents the environment as a sparse and non-uniform sample. As the distance from the head increases, the scanning density decreases proportionally. A representation of such 3D points in a Cartesian coordinate system poses a challenge in many applications where algorithms assume uniform spatial data distribution. Lately, authors have started proposing novel techniques that cope with the varying sample densities [13] by transforming the point clouds into uniformly sampled lattices. The initially added cost of transformation, at the end, results in benefits such as increased system flexibility and the possibility to use theoretically proven algorithms which are otherwise not applicable. In this analysis we will further explore this idea by segmenting the environment into disjoint objects using fast neighborhood indexing based segmentation. We refer to the work in [14] for a more detailed survey of 3D point cloud segmentation algorithms.

Assuming that the world consists of separate objects that are not physically connected to each other, we propose a segmentation algorithm for finding disjoint objects based on the region growing paradigm. The way objects are segmented also depends on the scene. In a traffic environment any vertical standing structure that is protruding above the local road surface or ground plane is a potential collision threat. Objects that are high enough to be included in this categorization can be frequently observed in the KITTI dataset: roadside barriers, traffic signs/lights, natural objects, vegetation and trees, animals, buildings, bridges, fences and other road users (vehicles, pedestrians, cyclists, etc.). Object boundaries are therefore very important and can be simply defined as the limits of free space that spans around the vehicle. We propose to segment the environment by representing it as a two dimensional occupancy grid. In this representation the ground plane has zero occupancy and anything that protrudes above this plane has a correspondingly higher probability of occupancy attached to the underlying grid cell. It has already been suggested that modeling objects as vertical lines, [15] is well suited as a common basis for scene understanding tasks of driver assistance and autonomous systems.

A. Ground plane removal and projection

Object segmentation of a driving environment should consist of the ground plane as a background segment and all potential collision threats as separate foreground segments. The first step is to remove points representing the local ground plane, or otherwise laying in close proximity. We use our RANSAC based ground estimation technique [16] to fit a plane to the data scanned in front of the vehicle. Later, we compute the occupancy grid (map), as the most as the most widely used environmental model, using our GPU implementation, as discussed in our previous work. In most cases the ground can be accurately segmented using plane fitting algorithms followed by a simple threshold. The produced occupancy map consists of two layers, a model of the probability of occupancy of each cell p, and the number of times a cell has been scanned, t:

$$\begin{aligned} &p\left(m_{i,j} \mid z\right) \mid i,j \in F \\ &t\left(m_{i,j} \mid z\right) \mid i,j \in F, \end{aligned} \tag{1}$$

where z is the point cloud data, and F is the frustum of the LiDAR. A visualization of one such map can be seen on the bottom left image in Fig.1. The map is a 2D matrix that resembles a regular grid of rectangular cells. Therefore, using the maps in (1) we can quickly segment points into foreground or background (FG/BG) by thresholding, using ground threshold τ :

$$O_{i,j} = \begin{cases} \text{strong FG}, p\left(m_{i,j}\right) \geq \tau, t\left(m_{i,j}\right) > 1 \\ \textit{weak}\ \text{FG}\ , p\left(m_{i,j}\right) \geq \tau, t\left(m_{i,j}\right) = 1 \\ BG \qquad\qquad\qquad\qquad , \textit{otherwise}. \end{cases} \tag{2}$$

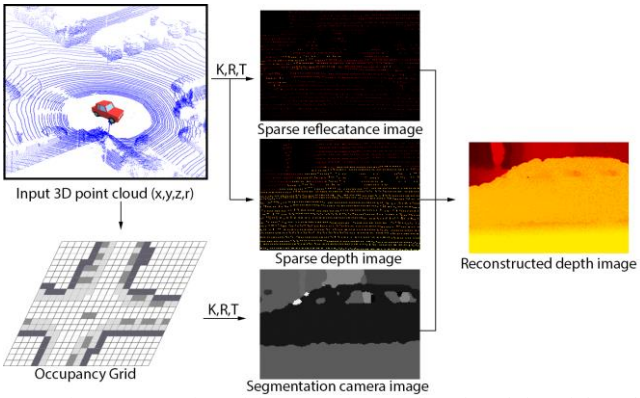

Figure 1. Overview of the processing pipeline, from left to right and from top to bottom: 3D visualization of the input data; the intermediate 2D occupancy from equation (1); camera projections of reflectance, depth and segmentation data; output: dense depth image

The choice of the value of τ can be adapted to be above the expected unevenness of the local road surface. In practice we chose a relatively "safe" value of $\tau = 0.25$ in order to segment most road users on the street and on the sidewalk. Due to the scanning sparsity, the obtained grid in (2) is suffering from noise and discontinuities especially in objects that are distant from the sensor. We label these sparsely scanned points as weak foreground (FG) objects so that they can be treated accordingly in the following steps. An intermediate morphological processing step is then applied to clean up the object boundaries in order to avoid over-segmentation.

B. Gap fling and clustering

Two distinct degradations in our resulting grid (2) can be observed: objects in the scene that have parts lying below the threshold, might get split up into several disjoint blobs, and second, parts of objects that are far away and only sparsely scanned, usually appear as disconnected noise. In order to keep all object parts within their respective blobs, and at the same time not merge too many blobs together, we propose a strategy of iterative weak object fusion. Weak object blobs in the threshold grid (2) are appended to any strong object that might be in the neighborhood *B*. These unified objects are now treated as the new strong objects and the procedure is repeated until it converges. Formally, at iteration *k,* we define this procedure as the morphological sequence:

$$O^{k}_{strongFG} = O^{k-1}_{strongFG} \cup \left\lfloor dilate\left(O^{k-1}_{strongFG}, B\right) \cap O_{weak} \right\rfloor . \quad (3)$$

In practice, the algorithm usually converges in less than 5 iterations depending on scene complexity and the type of LiDAR being used. Each unique object of interest must be physically separated from other objects, i.e. to be completely surrounded by at least one cell of empty space, thus, we treat each disjoint blob in our grid (3) as a unique object. The actual computation of disjoint blobs can be performed by any fast connected components labeling algorithm:

$$O_{idx} = \text{connected components}\left(O^{k}_{strongFG}\right). \quad (4)$$

3D points from the point cloud will be labeled with the respective object index from the blobs in (4). By varying the grid resolution and threshold parameters, we can tune our segmentation to separate specific objects such as pedestrians, cyclists, cars, buses, etc.

III. Depth Projection and Upsampling

Point cloud data generated by automotive LiDAR is semi-structured in a sense that the exact position of the LiDAR head can be computed for each 3D point. However, oftentimes the intelligent vehicle will be equipped with several LiDAR sensors that might not have the same specifications and can be poorly synchronized. In such realistic scenarios we must treat the agglomerated point cloud data as a single unstructured source where any assumption about the structure of this data is false. We offer an elegant solution to this challenge by transforming the point cloud from a single or multi-LiDAR sensors to a 2D image projection with arbitrary resolution.

A. Calibration and projection

We will assume a virtual pinhole camera positioned at the origin of the LiDAR coordinate system. The camera is pointed in the direction of travel of the vehicle and has a set of intrinsic parameters such as focal length and sensor size. 3D points which lie in R^3, are projected onto this virtual camera image by using the pinhole camera projection model:

$$\begin{aligned} \mathrm{d}_i &= P\mathrm{x}_i \\ \mathrm{x} &= (x, y, z, 1)^T \\ \mathrm{d} &= (u, v, 1)^T \\ P = KRT, K &\in \mathrm{R}^{3\mathrm{x}4}, R \in \mathrm{R}^{4\mathrm{x}4}, T \in \mathrm{R}^{4\mathrm{x}4}. \end{aligned} \quad (5)$$

Each LiDAR point ***x*** is mapped to a point ***d*** in camera coordinate system $\prod^{(u,v)}$ where the pixel value is our target quantity, which is the measured Euclidean distance or the reflectance index. The matrix K contains the intrinsic camera parameters, R is a camera rectification matrix and T holds the extrinsic camera parameters with respect to the LiDAR. Object segmentation indices and reflectance values are projected using the same camera viewport to the respective segmentation and reflectance images. When using reasonable intrinsic parameters, K, the depth and reflectance images are sparse. An example of such a projection can be seen in the middle column of Fig.1. Thus, we propose an up-sampling technique with a three-fold task: fill-in missing depth samples, preserve object boundaries and suppress the measurement noise.

B. Multilateral filter definition

Laboratory measurements on static objects with known distances show that the LiDAR noise follows a Gaussian distribution with a variance +-3cm. In turn, a filter with a Gaussian impulse response function will be a good candidate for noise suppression. Measuring objects in the real world, produces depth values which cover both flat and rough or edge regions. It can also be expected that the empty depth pixels contain the same or similar structure that needs to be accurately reconstructed. Object edges can be seen as a discontinuity in the depth function and flat regions have smoothly varying values. Thus it is natural that we do the reconstruction of the depth image using a form of an edge preserving filter. This task has been performed with lot of success in the RGB image domain using the original

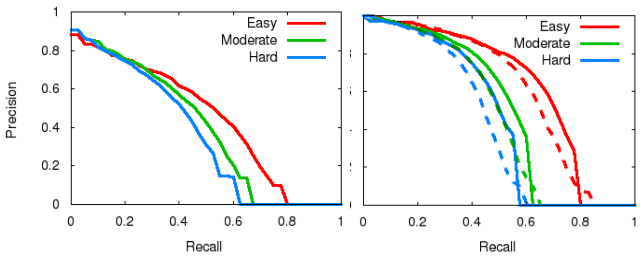

Figure 2. Left: performance of the baseline pedestrian detector [20] on KITTI, right: comparison between our RGB+D pedestrian detector (solid lines) and the RGB+D detector in [12] (dashed lines)

formulation of the bi-lateral filter [17]. It follows the paradigm of locally varying filter coefficients that filter the image in two directions simultaneously. Two functions measuring geometric closeness and photometric similarity are adapting the coefficients to the local image patches. The resulting filter is optimized to suppress noise while preserving details around sharp edges. Formally we can write a discrete version of the bi-lateral filter as:

$$d'_{u,v} = \frac{1}{w}\sum_{k,l}^{S} f(k,l)\,g\left(\left|d_{u,v} - d_{k,l}\right|\right)d_{u,v}\,, \qquad (6)$$

where the output $d'_{u,v}$ is a weighted average of the input value $d_{u,v}$ and the product of the functions $f()$ and $g()$ over the local neighborhood S. The first function measures the inverse Euclidean distance of pixel positions S, and the second function measures the distance in luminance, usually following a radial basis function.

We propose a filter that is loosely inspired by this formulation. The novelty in our approach lies in a formulation that works on both depth and IR reflectance values while extracting semantical information using the segmentation image for object association or guidance. Lastly, we use supervised learning to optimize the model parameters in order to achieve maximum reconstruction accuracy. Our method relies on two important assumptions about the nature of the input signal. Firstly, depth is a smoothly varying function except at object boundaries where the derivative is infinite and secondly, properties of IR reflectance image can be approximated with properties of natural light images i.e. smooth local variations and sharp object edges. In practice, it is very difficult to model the IR reflectance image since it is product of angle of incidence of the LiDAR light and the surface material properties of the object. This form of active light reflectance contrasts with RGB imaging where the reflection is often Lambertian in nature. However, our later experiments show that filtering the depth by including the IR reflectance image as a factor in the filter coefficients significantly improves reconstruction fidelity. Formally our upsampling and filtering function is a multilateral extension of (6) defined as:

$$d'_{u,v} = \frac{1}{w}\sum_{k,l}^{S} f_1(k,l)\,f_2\left(d_{u,v},d_{k,l}\right)f_3\left(r_{u,v},r_{k,l}\right)f_4\left(o^S,k,l\right)d_{u,v} \qquad (7)$$

where d, r and o are the projected camera images containing the LiDAR depth, reflectance and segmentation object index values as defined in (5). The weight factor w represents the sum of coefficients computed by the functions $f_{1..4}$ and S is a small interpolation window:

$$\begin{aligned} f_1(k,l) &= e^{-\alpha\|k,l\|^2} \\ f_2\left(d_{u,v},d_{k,l}\right) &= e^{-\beta\left\|d_{u,v}-d_{k,l}\right\|^2} \\ f_3\left(r_{u,v},r_{k,l}\right) &= e^{-\rho\left\|r_{u,v}-r_{k,l}\right\|^2} \\ f_4\left(o^S,k,l\right) &= \begin{cases} \gamma & ,if\ o^S_{k,l} = o^S_{dominant} \\ 1-\gamma & ,otherwise \end{cases}. \end{aligned} \qquad (8)$$

Functions $f_{1..3}$ compute vector distances in the three directions: pixel position space, depth space and IR reflectance space. The function f_4 operates on the local image patch o^S as follows: find the locally dominant object o^S_{dominant} by computing the statistical mode of the categorical object indices in the patch S, then find whether the object index of the current pixel is the same as the dominant object index. In practice this function allows interpolation to be performed only with measured values from the locally dominant object in the patch. Each time the patch moves through the image, a different dominant object is found and thus each pixel will be reconstructed by the data from the most frequently occurring object in its own neighborhood. The parameter γ controls how strict our rule is applied and its value usually stays close to 1. Empty areas of the image are assigned with object labels based on a local k-NN clustering, with the value of k varying based on the LiDAR model, usually k=3. On Fig.1, right, we show one typical example of the density and reconstruction quality that can be obtained by our proposed method.

C. Parameter optimization

Important part of our analysis is finding the optimal values for parameters α,β and ρ. The need for optimality can easily be deducted from the fact that in (7) we fuse image modalities with broadly different nature and value ranges. Pixel distance range is defined by the chosen camera focal length and sensor size, depth data is measured in meters with range 0m to 80m and IR reflectance is pre-processed internally by the LiDAR and is a relative dimensionless variable with values in the range of 0 to 1.

We define the model vector $\theta = [\alpha,\beta,\gamma,\rho,S,...]^T_P$ containing P parameters that we want to optimize over a set of N ground truth images $\{d^i_{GT}\}_{i=0..N-1}$ using a set of M input images triplets $\{d^i,r^i,o^i\}_{i=0..M-1}$ containing depth, IR reflectance and object indices as defined in the previous section. The function projecting the later set into the former is the one we formulated in (7) and depends on θ. We will use supervised learning technique to learn the optimal parameters that minimize the projection error defined by the following function:

$$E\left(\theta,\{d^i_{GT}\},\{d^i,r^i,o^i\}\right) = \sum_{i=0}^{N-1} PSNR\left(d^i_{GT},d^i_{reconstructed}\right), \qquad (9)$$

where the objective function is the peak signal to noise ratio between the reconstructed image and the ground truth depth map. Both the objective function and our proposed filtering function are non-linear, thus a closed form analytical solution for the minimum is not possible. We resort to an iterative optimization algorithm, namely steepest-gradient descent with an update step defined by:

$$\theta^k = \theta^{k-1} + \mu \frac{\partial E}{\partial \theta}\bigg|_{\theta^{k-1}}, \qquad (10)$$

where μ controls the inertia for the update of our solution. In practice we compute the derivative in (10) by computing the objective function at $E\left(\theta^{k-1} + \Delta\theta^{k-1}, \left\{d^i_{GT}\right\}, \left\{d^i, r^i, o^i\right\}\right)$ for combinations of small $\Delta\theta^{k-1}$. This operation is computationally expensive since the objective function has to be applied to a set of *N* images, so we tend to use the minimal training set size that produces a generally acceptable solution θ over a larger set of validation images. The optimal parameter set at convergence is able to even out the differences in data ranges of the different data modalities making each term in (8) to have a meaningful impact on the result.

IV. Experimental results

In the first section we will focus on measuring the absolute numerical accuracy of the reconstructed depth images and in the second section we will measure how much an object detection algorithm can benefit from dense depth information as an additional modality.

A. Quantitative analysis

Currently there seems to be no consensus on how to quantitatively evaluate the reconstruction of dense depth images coming from a LiDAR point cloud in automotive scenarios [12]. One possibly relevant dataset is the KITTI Stereo 2012 and Stereo 2015 [8] and [18] which uses dense depth maps from LiDAR as the ground truth to compare depth map reconstructions of various stereo algorithms. The ground truth of this dataset is built by registering a set of consecutive LiDAR frames (5 before and 5 after the frame of interest) using the iterative closest point algorithm. Accumulated point clouds are projected onto the camera image and then all ambiguous image regions such as windows and fences are manually removed. Using an exhaustive search through the provided raw data, [12] have found a practical sub-set of the Stereo 2015 ground truth images that we will also use to quantitatively measure our dense depth map reconstructions against. The dataset consists of 100 original point clouds and 100 corresponding dense point clouds considered as ground truth.

In our experiments we will be using the provided data as follows: initially, we project the sparse (input) LiDAR point clouds on a local occupancy grid with cell size of 0.125x0.125m. Ground plane estimation is performed as described in chapter *II.A* and the object segmentation is performed by the connected components algorithm implemented in the Quasar programming language [19]. Our proposed filtering and up-sampling works on local image patches with dimensions 17x30 pixels. Using supervised learning we found that the optimal values for the parameters $\{\alpha, \beta, \gamma, \rho\}$ are {0.129, 0.011, 0.999, 56.23} respectively. We used a random sub-sample of 30% to search for the optimal parameters and the remaining set was used for validation. In the end, depth images are converted to disparity images (KITTI Stereo 2015 baseline=0.537m) in order to compare to the ground truth format. The measured accuracies are given in Table I. We note that the parameter search was relatively invariant to the selected subset for training, thus we present the accuracy for the entire set of 100 point clouds in the dataset. Accuracy is measured by means of the metric *D-all:%* which represents the percent of outlier image pixels averaged over all ground truth pixels. Here outliers are pixels with disparity error greater than 3 pixels. We conducted the measurements using the provided MATLAB scripts in KITTI Stereo 2015 and obtained the accuracies which we compare to the work discussed in [12]. We are able to outperform the best algorithm in [12] by a significant margin. Interestingly, when we switch off the term f_3 in (8), again, we observe top performance. This shows that the fast segmentation technique we described in section *II* is better at capturing semantical information by preserving the object structure in the scene. Several examples of the reconstructed depth images, reflectance images and segmentation images can be seen on our project page [21].

B. Qualitative analysis

To show the potential effectiveness of our reconstructed depth images we re-trained a baseline pedestrian detection model using depth as additional modality. We expect that the addition of dense depth information to the available RGB data can boost both accuracy and robustness. The KITTI [8] object detection benchmark is currently the most relevant dataset for evaluation and ranking of object detection algorithms in the domain of autonomous and intelligent vehicles. It covers different urban scenarios, from university campus to downtown and residential areas.

The Aggregated Channel Feature (ACF) object detector [20] is currently one of the best performing algorithms that has a publicly available real-time implementation. We expanded the original formulation of the ACF architecture by adding our up-sampled depth maps to the processing pipeline and re-trained a pedestrian detection model. The classifier we chose is a multi-stage cascade of weak decision trees. Once the models are trained, the detection of pedestrians is

TABLE I. Accuracy of Depth Image Reconstruction, % of outlier pixels

Method	Reconstruction error,
Proposed	**2.75%**
Proposed[a]	**3.13%**
Premebida[12]	3.35%
Minimum	4.63%
Bilateral	4.77%
Delaunay	5.55%
Median	6.88%
IDW[12]	7.14%
KRI[12]	7.25%
Average	7.56%
Maximum	17.76%

performed on the combined RGB and our reconstructed depth images. In order to measure and compare our results to the literature, we uploaded the detected bounding boxes to the KITTI evaluation server where we obtained the results presented on a short extract in Table II. In this straight forward implementation we observe more than 10% improvement over the original ACF algorithm [20], and more than 4% improvement over the comparable RGB+D method in [11]. Otherwise, our proposed detector is the fastest and most accurate non-neural network based algorithm until the time of writing this paper. Fig.2 provides a more accurate overview of the precision/recall rates for the discussed models. Finally, we note that our depth map up-sampling is running on a moderately powerful GPU (gtx770) at 15fps and the pedestrian detector runs on 6 CPU cores in parallel at 30fps. Without any specific optimization both algorithms are keeping up with the sampling rate of the input data.

V. Conclusion

We tested the hypothesis that depth information plays an important role for the task of object detection in a driving scenario. Current trends indicate that the future of autonomous sensing systems will likely consist of a ranging system coupled with a multispectral video camera. Thus the availability of a semi-structured and sparse 3D point cloud can be used to generate a virtual depth camera whose images can improve the detection rates of the computer vision algorithms. Off-line tests on a publicly available pedestrian detection dataset showed the potential application of our up-sampled depth images. We discovered that even in perfectly well-lit situations, the pedestrian detection system can benefit greatly from additional depth information. We suspect that the boost in accuracy will be much greater in situations where the information in the visible light image is lacking, such as, during the night, under bridges or in tunnels, in harsh light conditions (sunrise/sunset), in fog or mist etc. Our main direction of future research is exploring these situations of bad weather in more detail in order to automatically reinforce the visible light image with the up-sampled depth image. We are also looking into the possibility to directly apply our technique to various automotive grade LiDAR sensors with lower specifications.

The main characteristic our approach is that it relies on accurate segmentation of the input image. The proposed segmentation technique can cope well with nearly flat roads, however our projection on a 2D occupancy grid is sensitive to changing road gradient. In traffic situations where the vehicle is approaching a ramp or a steep incline, parts of the road which are higher than the current surface will be segmented as separate objects. In a worst case scenario this will cause our segmentation algorithm to under-segment the occupancy map by connecting objects to the sloping road surface. This is not a catastrophic failure since the performance of the up-sampling would then be similar to the performance of the classical bi-lateral filter. On the other hand, a more accurate and robust segmentation technique can expected to yield depth images with higher fidelity.

TABLE II. Pedestrian detection results

Method	Accuracy	Execution
Proposed	**50.91%**	**0.03s**
Fusion DPM[12]	46.67%	30s
ACF[20]	39.81%	0.2s

References

[1] D. J. Fagnant, K. M. Kockelman, “Preparing a Nation for Autonomous Vehicles: Opportunities, Barriers and Policy Recommendations”, Eno Foundation (www.enotrans.org); at www.enotrans.org/wp-content/uploads/wpsc/downloadables/AV-paper.pdf

[2] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in CVPR. IEEE, 2005, pp. 886–893.

[3] R. Benenson, M. Omran, J. Hosang, B. Schiele, “Ten Years of Pedestrian Detection, What Have We Learned?”, ECCV 2014 CVRSUAD workshop proceedings

[4] A. Angelova, A. Krizhevsky, V. Vanhoucke, A. Ogale, D. Ferguson, “Real-Time Pedestrian Detection With Deep Network Cascades”, Proceedings of BMVC 2015

[5] X. Chen, K. Kundu, Y. Zhu, A. Berneshawi, H. Ma, S. Fidler and R. Urtasun, “3D Object Proposals for Accurate Object Class Detection”, NIPS 2015.

[6] F. Yang, W. Choi and Y. Lin, “Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers”, CVPR 2016.

[7] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler and R. Urtasun, ”Monocular 3D Object Detection for Autonomous Driving”, CVPR 2016.

[8] A. Geiger, P. Lenz, R. Urtasun, ”Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite”, CVPR 2012

[9] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. High-quality scanning using time-of-flight depth superresolution. In Time of Flight Camera based Computer Vision, 2008

[10] J. Dolson, J. Baek, C. Plagemann, and S. Thrun, “Upsampling range data in dynamic environments,” in CVPR. IEEE, 2010.

[11] C. Premebida, J. Carreira, J. Batista and U. Nunes, “Pedestrian Detection Combining RGB and Dense LIDAR Data”, 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

[12] C. Premebida, L. Garrote, A. Asvadi, A. Pedro Ribeiro, U. Nunes, “High-resolution LIDAR-based Depth Mapping using Bilateral Filter”, 2016 IEEE ITSC

[13] Michiel Vlaminc, Quang Luong, Werner Goeman, Peter Veelaert and Wilfried Philips, “Towards online mobile mapping using inhomogeneous lidar data”, (2016) Proceedings of IEEE Intelligent Vehicles Symposium.

[14] Anh Nguyen, Bac Le, “3D point cloud segmentation: A survey”, 2013 6th International Conference on Robotics, Automation and Mechatronics (RAM)

[15] D. Pfeiffer, U. Franke, “Modeling dynamic 3D environments by means of the stixel world”, IEEE Intelligent Transportation Systems Magazine, Vol: 3, Issue: 3, 2011

[16] M. Dimitrievski, D. Van Hamme, P. Veelaert and W. Philips, “Robust matching of occupancy maps for odometry in autonomous vehicles”, (2016) Proceedings of 2016 VISAPP. p.626-633

[17] C. Tomasi, R. Manduchi, “Bilateral Filtering for Gray and Color Images”, Proceedings of the 1998 IEEE International Conference on Computer Vision

[18] M. Menze, A. Geiger, “Object Scene Flow for Autonomous Vehicles”, Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR) 2015

[19] B. Goossens, J. De Vylder, S. Donné, W. Philips, “Quasar - a new programming framework for real-time image/video processing on GPU and CPU”, (2015) 9th International Conference on Distributed Smart Cameras. p.205-206

[20] P. Dollar, R. Appel, S. Belongie, and P. Perona, “Fast Feature Pyramids for Object Detection”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014,Vol. 36, Issue 8, (pages 1532 – 1545)

[21] http://telin.ugent.be/~mdimitri/depth.html