# High-resolution LIDAR-based Depth Mapping using Bilateral Filter

Cristiano Premebida, Luis Garrote, Alireza Asvadi, A. Pedro Ribeiro and Urbano Nunes

*Abstract*— High resolution depth-maps, obtained by upsampling sparse range data from a 3D-LIDAR, find applications in many fields ranging from sensory perception to semantic segmentation and object detection. Upsampling is often based on combining data from a monocular camera to compensate the low-resolution of a LIDAR. This paper, on the other hand, introduces a novel framework to obtain dense depth-map solely from a single LIDAR point cloud; which is a research direction that has been barely explored. The formulation behind the proposed depth-mapping process relies on local spatial interpolation, using sliding-window (mask) technique, and on the Bilateral Filter (BF) where the variable of interest, the distance from the sensor, is considered in the interpolation problem. In particular, the BF is conveniently modified to perform depth-map upsampling such that the edges (foreground-background discontinuities) are better preserved by means of a proposed method which influences the range-based weighting term. Other methods for spatial upsampling are discussed, evaluated and compared in terms of different error measures. This paper also researches the role of the mask's size in the performance of the implemented methods. Quantitative and qualitative results from experiments on the KITTI Database, using LIDAR point clouds only, show very satisfactory performance of the approach introduced in this work.

Fig. 1. Example of an urban scene where the LIDAR point-cloud is shown in pixel coordinates (bottom). The sparseness of the LIDAR is emphasized in the zoom view at the bottom-right. This frame is part of the KITTI-dataset, where the LIDAR is mounted on the roof of a vehicle and the LIDAR covers a field-of-view (FOV) limited by a vertical angle - approximated by a line (red dashed). This line will be hereafter called horizon line.

## I. INTRODUCTION

This paper deals with the problem of obtaining a depth map, in pixel coordinates, from a single 3D point-cloud generated by a LIDAR mounted on-board an instrumented vehicle. Assuming the LIDAR is calibrated wrt a monocular camera, the transformation of the 3D point-cloud to the image plane generates a sparse distribution of points, as shown in Fig. 1 (zoom at the bottom-right), where the majority (more than 90%) of pixels are unsampled. Some difficulties arise in obtaining a consistent and dense (high resolution) depth map, where consistency has to do with obtaining a good estimation of depth values in edges and smooth regions of the depth-map. On the other hand, density is related to the spatial resolution of the map where the problem is caused by sparse and incomplete data. Upsampling sparse 3D point-clouds to obtain a high resolution depth image can be explored in the context of artificial sensory-perception for intelligent/autonomous vehicles and ADAS applications, such as: road and obstacle detection [1], curb detection [2], vehicle detection [3].

The majority of the works in this research area use, in conjunction with range data, information from monocular camera as in [4]–[7]. This work, instead, differs from existing solutions because the proposed approach uses only data from a LIDAR, and therefore monocular camera is considered just

The authors are with the Institute of Systems and Robotics, Dept. of Electrical and Computer Engineering, University of Coimbra, Portugal. {cpremebida,garrote,asvadi,urbano}@isr.uc.pt

for calibration and visualization purposes. More specifically, a Bilateral Filter (BF) based framework is described which generates, from single 3D point-cloud delivered by a Velodyne HDL-64, a high resolution depth-map. Upsampling 3D point-clouds to produce a dense depth map based solely on data provided by a LIDAR i.e., color/texture information from camera is not been used, is a research area with very few scientific literature. The recent work of Miksik *et al.* [7] shares some common points with the approach described in this paper but, data from a stereo-image system is used in their algorithm. Conversely, methods that combine data from LIDAR and camera i.e., the depth upsampling strategy is enhanced by color and texture information, are much more common as evidenced by a number of interesting scientific works, such as [4], [5], [8]. Besides the problem of the low number of sampled points provided by a single point-cloud, which corresponds to less than 10% of the region within the FOV (see Fig. 1), particular attention has to be given to the ambiguity problem of foreground vs background, that occurs in the areas of discontinuities (edges) between objects (such as a pedestrian or a vehicle) and far away objects (background). This is an aspect that will be addressed in detail in Sect III and is one of the main reasons to use edge-preserving methods like the BF. This work utilizes the structure of edge-preserving filters, namely the BF, which means that the sampled points in the local window are weighted by combining a neighborhood distance function and a range based function. The first weighting function does not contribute significantly to the performance of the system, while the second function plays a key role on the final results. These aspects will be further discussed and demonstrated.

Edge-preserving filters are very popular in the computer

vision community, and they are used in many applications such as noise reduction, stereo matching, image deconvolution, image upsampling. Examples of this type of filter are the Anisotropic Diffusion filter [9], the Bilateral filter [10], [4], [11], and the recent Guided filter [12].

In terms of contributions, this paper proposes an upsampling framework for high-resolution depth mapping based on the BF, using an original approach to process the range values, within a local mask, where the discontinuities (edges) are meaningfully preserved. The proposed approach is particularly suitable for LIDAR only depth maps. Moreover, this work provides a thorough experimental validation of the framework, considering qualitative and quantitative criteria, and reports experimental results and comparative evaluation with several other techniques.

## II. DEPTH-MAP UPSAMPLING FORMULATION

Given a 3D point-cloud $PC^t \subset \mathbb{R}^3$ produced by a LIDAR at a given time-stamp $t$ (from now on $t$-index will be omitted to simplify the notations), and assuming the LIDAR is calibrated wrt a monocular camera with image-plane denoted by $\Pi_2$, the formulation starts by considering the set of points $P \in PC$ that lie within $\Pi_2$ i.e., $P$ is obtained after the transformation from $\mathbb{R}^3$ to the camera coordinate system and then to the image-plane using the calibration matrix. Given the set of points $P \in \Pi_2$, with $P$ having non-integer pixel coordinates and being sparse, the goal here is to obtain a high resolution depth-map $DM$ (high density of points) limited by the size of $\Pi_2$ ($mu \times mv$) and by the horizon line. In terms of locations, $DM$ is restricted to positive-integer values i.e., pixel locations, in the range $[0, 1, \cdots, mv]$ and $[0, 1, \cdots, mu]$ respectively for the horizontal and vertical positions.

The elements of $P$ are the points $\{\mathbf{p_1}, \cdots, \mathbf{p_n}\}$, where each point $\mathbf{p_i} = (u, v, r)_i$ is represented by the position in pixel coordinates $(u, v)_i$ and by the range value $r_i$ as measured by the LIDAR. However, since $u$ and $v$ are finite real numbers, the position coordinates of $P$ can be rounded to integer values in $DM$ for purpose of computational efficiency. At this stage i.e., without further processing, three cases may occur: (case1) locations of $DM$ with just one point; (case2) locations with more than one point from $P$; and (case3) locations of $DM$ without corresponding point ("empty"). In this paper, which deals with the problem of obtaining depth-maps from a Velodyne HDL-64 sensor, the case1 occurs in approximately 6.7% of pixel locations of the $DM$, in the case2 we have about 0.1% of locations with more than a single point and finally, for the case3, about 93.2% of positions in $DM$ are empty (unsampled). Those percentages were calculated from experimental data using 100 frames of the KITTI dataset (more details in Sect. V-A). Although negligible for one pixel-size (let's say a mask of $1 \times 1$), the case2 deserves particular attention as the number of sample points rapidly increases as the size of the area of interest, the local window, increases to $3 \times 3$, $5 \times 5$, and so on. This situation is better understood after viewing Fig. 2 where, due to the LIDAR's perspective viewing, range points pertaining to the cyclist, located in the foreground, and points belonging
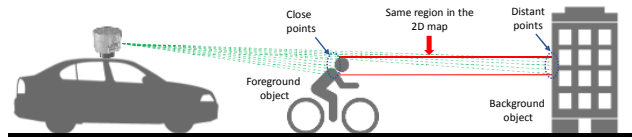


Fig. 2. Illustration of the superposition between close (foreground) and far (background) objects, whose measurement points will be located in the same local (vicinity) region of the depth-map.

to the background are situated in the same $DM$'s location (the zoomed view in the bottom row of Fig. 1 illustrates this problem.). This yields large deviations for the average range value at a local window and is the main cause of errors in the depth map. As explained before, the set of measurement points from a LIDAR sensor, in the form of the point-cloud $PC$, are converted to discrete locations in a two-dimensional map $DM$, in pixel coordinates, and is represented by $P$. The main difference from other upsampling problems is that $DM$ has a very low density of points: of the order of 6.8%. Therefore, the goal is to find a technique to estimate the value of $r$ in unsampled locations of $DM$ and keeping those estimated values of depth (range-distance to the LIDAR) consistent through the depth-map.

The discussion above allows us to concentrate on solving the *case3* thus, we need a solution of estimating the desirable variable $r$ of the unsampled positions in $DM$ and, at the same time, the errors due to the boundaries regions (edges) should be minimized. Carrying out a solution of estimating depth values in locations without measurement points will result in a depth-map with higher resolution than the input; this is known as "upsampling" and can be formulated as a spatial estimation problem as detailed in the next sections.

### A. Defining the region of interest

Spatial data interpolation, or estimation, is typically performed under the assumption that a local region of interest $R$ is previously defined and the desired point-value to be estimated is located within $R$; this is known as **local interpolation**. In a more general case, any polygon shape can be used to define $R$, however, in the problem considered here the most usual solutions are square regions (usually called mask or window). Nevertheless, for purpose of completeness, solutions based on Delaunay-triangles will also be considered. Delaunay-triangulation is effective in obtaining depth-maps with close to 100% of density i.e., all locations with unknown depth-values are estimated, because this method interpolates all points regardless the distance between the points of a triangle. Except for Delaunay based methods, all the interpolation methods discussed in this paper are applied to estimate the range value of locations centred at the local and square window $R$ with size defined by $mr \times mr$ (in pixel units). The value of $mr$ is not simple to decide and has direct impact on the number of available points to estimate the variable of interest and consequently on the computational effort. Moreover, an important aspect to take into account is the minimum number of points in $R$

necessary to guarantee consistency, statistical significance and efficiency of the estimator. This paper will address this issue by experiments, investigating the spatial-resolution of the *DM* (i.e., its density) and some statistics for increasing values of *mr*. The implementation consists in 'moving' *R* through the locations of *DM* using the sliding window technique; then, all the points within *R* are considered for estimating, locally, the depth value of the centre point.

### B. Local interpolation algorithms

The role of an interpolation method is to estimate values of range measurements, from a LIDAR, in both sampled and unsampled (empty) locations of the depth-map (*DM*). The locations in *DM* are in pixel coordinates and the interpolation is performed locally i.e., restricted to a local region *R*. Let $\mathbf{x}_0 = (u,v)_0$ be the location of interest, which is the centre of *R*, and let $r_0^*$ be the variable to be estimated, that is, the range-distance *r* at $x_0$. Given a finite set of measured points $\mathbf{p}_i = (\mathbf{x}_i, r_i), i = 1, 2, \cdots, n$, where $\mathbf{x}_i \in R \subset DM$, spatial interpolation can be formulated in terms of a function that weights the depth values $r_i$ of the points $\mathbf{p}_i$ according to spatial (position) based parameters $\Theta(\mathbf{x}_i)$ thus, $r_0^* = f(r_i, \Theta(\mathbf{x}_i))$. There are many possibilities for $f(\cdot)$ to solve the problem but, for obvious reasons, we will address a limited number of methods. Besides, the following basic operators will be considered: sample average, minimum, maximum, median, nearest neighbor. The following three classes of interpolation techniques will be considered in this work: 1) inverse distance weighting (*IDW*), the simplest variant of the Shepards Method; 2) the Ordinary Kriging (*KRI*) and 3) a Polygon based method using Delaunay triangles (*DEL*).

The *IDW* can be expressed as $r_0^* = \sum_{i=1}^n \omega_i(\mathbf{x})r_i / \sum_{i=1}^n \omega_i(\mathbf{x})$, where $\omega_i(\mathbf{x}) = d_i^{-p}$, $d = ||\mathbf{x}_0 - \mathbf{x}_i||$ is a given distance function (a metric), and *p* is a power parameter (positive real number). In a general form, the weights can be generalized to an arbitrary kernel function $K(\cdot)$ yielding, for Kernel-based methods, the expression: $r_0^* = \sum_{i=1}^n K(\mathbf{x}_0, \mathbf{x}_i)r_i$. *KRI* is an optimal linear estimator that estimates the value of a random function at the location of interest, $\mathbf{x}_0$, from samples $\mathbf{x}_i$ located in the local region of interest. The points $\mathbf{x}_i$ are weighted according to a covariance function or the equivalent semivariogram $\gamma(h)$. The parameters $\Theta$ in this method are the values of *nugget*, *sill* and *range*; where *h* represents the lag distance i.e., a distance measure between points. In the *DEL* approach, the value of the unsampled point of interest, which lies within the triangle, can be estimated by different techniques. In this work, we use the Matlab classes *delaunayTriangulation* and *scatteredInterpolant* to interpolate the three points of a given triangle; the available techniques in *scatteredInterpolant* are: linear, nearest neighbor, and natural neighbor interpolation.

In common with all those interpolation methods, the weighting is performed as function of the position of the sampled points $\mathbf{x}_i$, and the variable of interest ($r_i$) is not considered in the problem formulation. Conversely, the BF [10] allows one of its weighting terms to be dependent of the variable of interest, in our case the range distance ($r_i$). For this reason, and due to the successful performance of Bilateral filtering in edge-preserving applications [13], we propose a modified version of BF to upsample depth maps.

### III. EDGE-PRESERVING ON DEPTH MAP UPSAMPLING

In this section we briefly review the BF and propose a new range-weighting technique for upsampling depth-maps using BF. For a detailed review on edge-preserving filters, in the domain of image processing, please see [12].

### A. Bilateral filter

Following the notations in Sect. II, Bilateral filtering [10] [13] can be expressed as follows:

$$r_0^* = \frac{1}{W} \sum_{\mathbf{x}_i \in R} G_{\sigma_s}(||\mathbf{x}_0 - \mathbf{x}_i||) G_{\sigma_r}(|r_0 - r_i|) r_i \qquad (1)$$

where $G_{\sigma_s}$ weights the points $\mathbf{x}_i$ inversely to their distance to the position of interest $\mathbf{x}_0$, $G_{\sigma_r}$ controls the influence of the sampled points as function of their range values $r_i$, and finally *W* is a normalization factor that ensures weights sum to one. In (1), we set $G_{\sigma_s}$ to be inversely proportional to the Euclidean distance between the center of the mask *R* and the sampled locations $\mathbf{x}_i$, yielding $G_{\sigma_s} = \frac{1}{1+(||\mathbf{x}_0 - \mathbf{x}_i||)}$. The influence of $G_{\sigma_s}$ is not very significant because the problem of jump discontinuities is caused by differences in range. On the other hand, $G_{\sigma_r}$ is the key component to be explored in order to provide improvement in the estimation of $r_0^*$, under the influence of discontinuities between foreground and background. A common form of weighting the range values, as in [11], is given by $G_{\sigma_r} = \frac{1}{1+(|r_0 - r_i|)}$. However, as mentioned in Sect. II, the average percentage of centred pixel $\in R$ with range values is (only) 6.8% therefore, the nearest value $r_0 = \min(r_i), \forall r_i \in R$ has been chosen at an unsampled location $\mathbf{x}_0$.

### B. Modified Bilateral filter

We propose a modification in the BF, henceforth indicated as *BF*\*, by expressing the weighting element $G_{\sigma_r}$ as a function of the 'dispersion' of *r* in the mask *R*. Assuming that an edge is characterized by a discontinuity in the range values, we propose to use a clustering algorithm to detect discontinuities and, if it is the case, the number of clusters (*nc*) will be at least two; therefore, an edge (or a discontinuity) occurs if $nc > 1$ (as shown in Fig. 3). The algorithm used to perform clustering is based on the popular DBSCAN [14], which is a simple and effective algorithm that depends on two parameters, $\varepsilon$ and *minPts*. The implementation of the DBSCAN should take into consideration six definitions, as detailed in [14], and a distance function between points. In this work, we consider the distance function (*DF*) as given by $DF = |\frac{r_k - r_{k+1}}{r_k + r_{k+1}}|$, $k = 1, \cdots, nR$, where *nR* is the number of points $\in R$. If $DF > \varepsilon$ i.e., a discontinuity has been detected, the occurrence of more than one cluster is likely true. A cluster ($s_i$) is accepted only if $minPts > 1$. Ideally, the number of clusters corresponding to a window where an edge occurs should be $nc = 2$ i.e., one clustered

Fig. 3. The first row is an example-frame from the KITTI dataset and serves for visualization purpose; in the row below, the pixels in light-grey indicate the center of masks where $nc > 1$ hence, corresponding to discontinuities/edges between foreground and background.

set of points belonging to the foreground and another to the background. We conducted experiments using the DBSCAN algorithm, with $\varepsilon = 0.08$ and $minPts = 2$, and found out that $nc$ is equal to two in the majority of the regions where an edge occurs. Figure 3 provides a visual display where the light-grey pixels correspond to regions with $nc > 1$. In some circumstances, however, $nc$ is greater than 2 and it is not easy to detect the 'optimal' boundary between foreground vs background. Whenever $nc \geq 2$, the approach presented in this section considers at most two clusters; conversely, if $nc = 1$ equ. (1) is applied to all the points in $R$. When $nc > 1$ the $BF^*$ uses a ratio ($\lambda$) between the number of points of, at most, two clusters.

Let $np_1$ and $np_2$ be the number of points belonging to the clusters $s_1$ and $s_2$, where $\lambda = np_1/np_2$ is the ratio of interest. The variable $np_1$ corresponds to the cluster that has the closest average distance to the LIDAR (denoted cluster $s_1$); this holds regardless of the number of remaining clusters. However, if $nc > 2$ and excluding the cluster $s_1$, then a decision process is carried out, to select $s_2$, according to the following rule: $s_2$ is chosen as the cluster with more points and, in case of clusters with the same number of points, then $s_2$ corresponds to the one with the closest average distance. Once the pair $(s_1, s_2)$ has been selected, then a threshold-based rule is applied to the ratio $\lambda \geq Thr$ in order to penalize $s_1$ or $s_2$. This rule is exclusive in the sense that one of the clusters will be excluded, therefore if $\lambda \geq Thr$, then the $BF^*$ will be run on the points belonging to $s_1$ else, only the points in $s_2$ will be considered. The key idea is to strongly penalize, based on the ratio $\lambda$, one of two clusters and, as consequence, only points belonging to one of the clusters will be considered in (1).

## IV. Dataset and evaluation methodology

Publicly datasets for purpose of LIDAR-based depth map evaluation and performance assessment are not available at the time of this writing. Therefore, to provide quantitative evaluation, we resort to the *KITTI Stereo 2015* which provides groundtruth for disparity maps evaluation and benchmark. But, because that dataset was built to evaluate stereo systems, we had to find a solution that makes the evaluation of LIDAR-based depth maps possible; this is explained in the following section.



Fig. 4. An example of groundtruth point cloud, superimposed on a color image, with some objects (vehicles) having better resolution than the majority regions in the groundtruth. Those well delineated vehicles, that represent foreground objects, are a challenging situation for a solution using single LIDAR-based point cloud.

### A. KITTI dataset for depth-map evaluation

The present *KITTI Stereo 2015* comprises a total of 400 frames from stereo images, 200 for training with their corresponding disparity maps (the groundtruth), and 200 for benchmarking purposes, where each frame has two associated images: one from each camera of the stereo pair. The *KITTI Stereo 2015* is part of the *KITTI Vision Benchmark Suite*, being the latter composed of thousands of frames from different sensors: a Velodyne LIDAR, cameras, and GPS/IMU. The groundtruth for disparity evaluation was created using a process that incorporates a set of consecutive scans from the LIDAR (5 before and 5 after the actual frame of interest), where this sequence of point clouds were conveniently merged by a ICP technique as reported in [15], followed by a manually correction step to rectify eventual ambiguities. Nevertheless, the actual *KITTI Stereo 2015* dataset provides a more challenging and accurate groundtruth, described in [16], where "objects" (vehicles) in the scenes were recovered by fitting detailed CAD models to the point clouds. The consequence is a set of groundtruth frames where some objects are very well delineated as shown in Fig. 4. To make possible the evaluation of depth-maps generated solely by LIDAR, a new dataset - using data from KITTI - had to be built. First, the groundtruth, originally in the form of disparity maps, has to be converted to depth maps by known geometry. To calculate depth $Y_E$ of a disparity map $Y_I$, it is required to know the values of the baseline $B$ and the focal length $f_c$, and the conversion is given by $Y_E = Bf_c/Y_I$. Once the *KITTI Stereo 2015* made available image-frames and the corresponding disparity maps, it is necessary to find the LIDAR scans that match the groundtruth frames. In this work, we performed a non-exhaustive search in the "Raw Data" recordings of the *KITTI Vision Benchmark Suite* and established the correspondence between 100 LIDAR scans and their counterpart in the groundtruth set in *KITTI Stereo 2015*. Henceforth, the experiments presented in the next sections were carried out using this set of 100 scans.

### B. Methodology

The evaluation methodology adopted in this work is similar to the one used in the *KITTI Stereo 2015* benchmark, excepting the following: the total number of frames used in the evaluation is 100, and the separation of training and testing set is not carried out once our approach does not depend on a learning strategy. In short, we adapted the C++ codes of the development kit package, from KITTI website,

to run the evaluation routine on the dataset described above. Four performance measures, detailed in [16], are calculated: *D-bg*, *D-fg*, *D-all*, and *Density*. Although the groundtruth, as already mentioned, has been obtained by a combination of 10 point clouds and by complementary object CAD-models, the number of unsampled pixels in the groundtruth depth-maps is still significant (see Fig. 4). So, the groundtruth depth maps are not 100% dense i.e., the value of *Density* is calculated considering only the sampled pixels [16].

## V. Experiments and evaluation

This section describes the experiments conducted on the dataset detailed in Sect. IV-A, with the goals of assessing the performance of the local-region ($R$) and evaluating the approaches discussed in Sect. II-B and Sect. III. The reported results are from the non-occluded set (*Disp_noc_0*). An appropriate solution to mitigate the errors caused by discontinuities between foreground and background will depend, primary, on the detection of the occurrence of such discontinuities. As mentioned in Sect. III-A, a BF-modified version, $BF^*$, was proposed to obtain consistent depth-map from solely LIDAR data.

### A. The role of the region of interest

The size of $R$, defined by $mr \times mr$ in pixel units as described in Sect.II-A, controls the number of points (within $R$) to be used by a given local-interpolation approach. Table I provides the results, averaged over the 100 frames of the dataset, for increasing values of $mr$. The average number of points in $R$ is denoted by $N_{ave}$, while $N_{max}$ indicates the maximum number of points. These values were obtained by applying a sliding-window strategy with step of 1 pixel. The density of the map, denoted by $D_{ens}$ and given in percentage, is calculated considering the region covered by the LIDAR's field-of-view (FOV). As shown in Fig. 1, the LIDAR measurements cover a limited vertical FOV i.e., the upper part of the image plane is empty, and hence only the pixels below a certain 'horizon line' (see Fig. 1) are used to compute $D_{ens}$. A density of 100% means that the window $R_{mr \times mr}$ had, in all locations of the sparse map, at least 1 point inside $R$. The 'horizon line' was calculated by averaging, for each column of $DM$, the points of $P$ that have the smallest value in vertical-axis.

The KITTI Database provides the MATLAB/C++ utility package used in the evaluation of the algorithms. Density is evaluated against the KITTI groundtruth and, therefore, it follows a methodology which is not the same as the above. For that reason, the values of density (denoted by $Den^*$) as calculated by the KITTI evaluation package, shown in the last row of Table I, are not the same of $D_{ens}$.

### B. Evaluation of local-spatial interpolation algorithms

The first experiments conducted in this work involve methods that do not depend on the size of $R$, which are the cases of the Nearest ($NEA_{nei}$) operator and the Delaunay-based techniques using linear ($DEL_{lin}$), nearest neighbor ($DEL_{nei}$), and natural neighbor ($DEL_{nat}$) interpolation. In terms of error

### TABLE I
STATISTICS FOR INCREASING SIZE OF $R$.

| $mr$ | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 |
|---|---|---|---|---|---|---|---|---|---|
| $N_{max}$ | 6 | 11 | 17 | 23 | 36 | 42 | 55 | 64 | 75 |
| $N_{ave}$ | 0.6 | 1.7 | 3.3 | 5.5 | 8.2 | 11.4 | 15.1 | 19.4 | 24.2 |
| $D_{ens}$ | 48.9 | 81.1 | 93.5 | 96.6 | 97.5 | 97.9 | 98.3 | 98.6 | 98.8 |
| $Den^*$ | 58.2 | 87.2 | 96.6 | 98.7 | 99.3 | 99.5 | 99.6 | 99.7 | 99.8 |

performance, as shown in Table II, $NEA_{nei}$ and $DEL_{nea}$ achieved comparable results, however the main difference resides in the fact that the density resulted from $NEA_{nei}$ varies with the window size ($mr$), while the Delaunay-based approaches attain the same value of density, being equal to 99.96%. The methods *IDW* and *KRI*, the operators sample average (*AVE*), minimum (*MIN*), maximum (*MAX*) and median (*MED*), and also the BF and $BF^*$ have different performances for different values of $mr$. For this reason, and to show the relationship between the errors and $mr$, Figure 5 provides the values of the error measures (*D1-bg*, *D1-fg* and *D1-all*) for increasing values of window size.

### TABLE II
EVALUATION RESULTS USING METHODS *NEA* AND *DEL*.

| Method | D1-fg | D1-bg | D1-all |
|---|---|---|---|
| $NEA_{nei}$ | 17.47 % | 3.78 % | 5.53 % |
| $DEL_{lin}$ | 22.05 % | 4.15 % | 6.48 % |
| $DEL_{nea}$ | 17.10 % | 3.82 % | 5.55 % |
| $DEL_{nat}$ | 23.54 % | 4.21 % | 6.73 % |

### TABLE III
EVALUATION RESULTS FOR $mr = 13$.

| Method | D1-fg | D1-bg | D1-all |
|---|---|---|---|
| $BF^*$ | 8.23 % | 2.63 % | 3.35 % |
| *MIN* | 7.57 % | 4.20 % | 4.63 % |
| *BF* | 14.64 % | 3.32 % | 4.77 % |
| *MED* | 20.37 % | 4.91 % | 6.88 % |
| *IDW* | 25.84 % | 4.41 % | 7.14 % |
| *KRI* | 25.77 % | 4.54 % | 7.25 % |
| *AVE* | 26.67 % | 4.77 % | 7.56 % |
| *MAX* | 34.12 % | 15.37 % | 17.76 % |

### C. Discussion

Considering the results shown in Fig. 5 the error (*D1-fg*) on the foreground objects (Fig. 5(a)) demonstrated to be the most challenging case, particularly for the operators *MAX*, *AVE*, and methods *KRI*, *IDW*, where the error increases monotonically with $mr$. For the median operator (*MED*), the situation is also not favorable. The *MIN* shows a good behavior up to $mr = 15$, while *BF* and $BF^*$ are relatively robust for all values of $mr$, although the proposed $BF^*$ attained the best results. In terms of background-errors (*D1-bg*), depicted in Fig. 5(b), most of the methods are generally satisfactory except, clearly, the operators *MAX*, *MIN* and *MED*. Finally, and as consequence of the combined errors (*D1-all = D1-fg + D1-bg*), from Fig. 5(c) it is possible to conclude that $BF^*$ achieved the lowest error among the implemented methods. Based on the values of density, provided
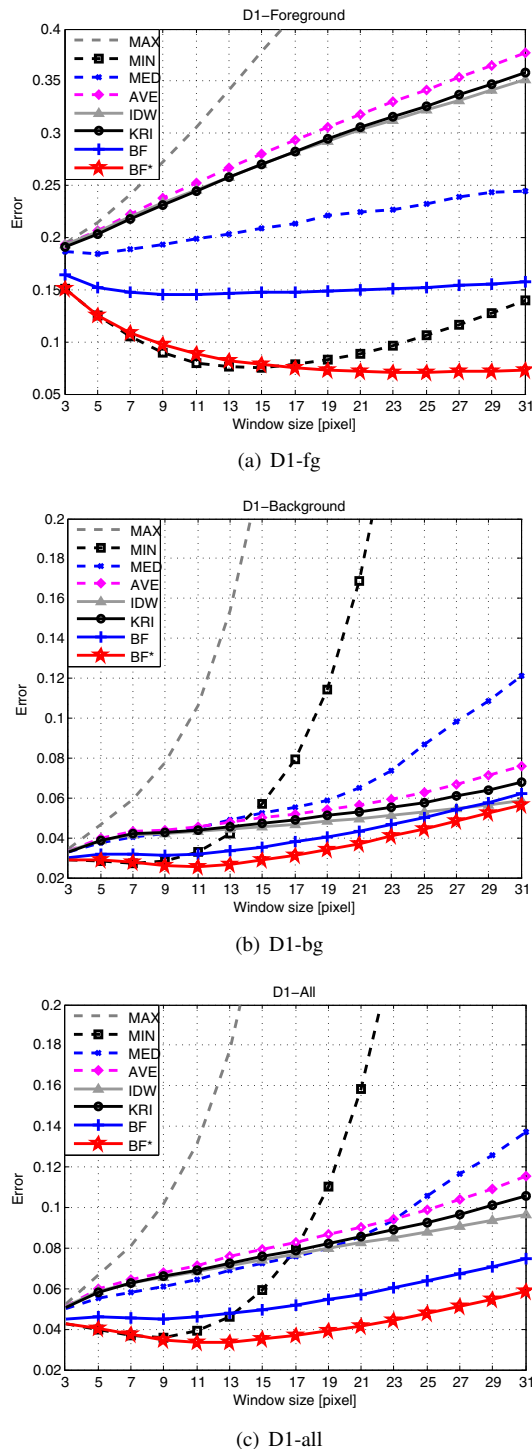
(a) D1-fg



(b) D1-bg



(c) D1-all

Fig. 5. Error curves, as function of the window size *mr*, for the methods and techniques addressed in this work. The curves for the Delaunay-based techniques and the nearest-neighbor are omitted because *mr* does not influence their error performances.

in Table I, and considering the error plots in Fig. 5, the 'optimum' value of the window size is $mr = 13$. Therefore, Table III shows the values of error, in percentage and for a window size of $13 \times 13$, which facilitates the comparison of results among the methods and techniques, including those reported in Table II. Finally, further qualitative results

from the techniques and methods discussed in this paper are provided as supplementary material [1].

## VI. CONCLUSION

A high-resolution, LIDAR-based only, depth mapping approach is presented in this work as a promising solution to be used in sensory perception systems, as part of applications such as: road detection, object recognition and tracking, environment modeling, cooperative perception. The approach is based on the Bilateral Filter (BF) and contributes with a technique that influences the weighting range-term of the BF. Experiments using the KITTI database were carried out to assess the performance of the proposed approach as well as other usual interpolation techniques, namely: Kriging, IDW, Delaunay interpolation, Median, Nearest neighbor, and others. From the experimental results reported in this paper, the proposed approach, denoted by $BF^*$, attained the best results among the methods and techniques tested.

## REFERENCES

[1] P. Y. Shinzato, D. F. Wolf, and C. Stiller, "Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion," in *IEEE IV*, June 2014, pp. 687–692.

[2] C. Fernandez, D. F. Llorca, C. Stiller, and M. A. Sotelo, "Curvature-based curb detection method in urban environments using stereo and laser," in *IEEE Intelligent Vehicles Symposium (IV)*, 2015.

[3] F. Zhang, D. Clarke, and A. Knoll, "Vehicle detection based on LiDAR and camera fusion," in *IEEE ITSC*, Oct 2014, pp. 1620–1625.

[4] J. Dolson, J. Baek, C. Plagemann, and S. Thrun, "Upsampling range data in dynamic environments," in *IEEE CVPR*, 2010.

[5] H. Andreasson, R. Triebel, and A. Lilienthal, "Non-iterative vision-based interpolation of 3D laser scans," in *Autonomous Robots and Agents*, vol. 76.

[6] Y. He, L. Chen, and M. Li, "Sparse depth map upsampling with rgb image and anisotropic diffusion tensor," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*, June 2015, pp. 205–210.

[7] O. Miksik, Y. Amar, V. Vineet, P. Perez, and P. Torr, "Incremental dense multi-modal 3d scene reconstruction," in *IEEE/RSJ IROS*, 2015.

[8] A. Gonzalez, G. Villalonga, J. Xu, D. Vazquez, J. Amores, and A. M. Lopez, "Multiview random forest of local experts combining RGB and LIDAR data for pedestrian detection," in *IEEE IV*, 2015.

[9] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, Jul 1990.

[10] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *IEEE CVPR*, 1998.

[11] C. Premebida, J. Carreira, J. Batista, and U. Nunes, "Pedestrian detection combining RGB and dense LIDAR data," in *IROS*, 2014.

[12] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE TPAMI*, vol. 35, no. 6, pp. 1397–1409, 2013.

[13] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, "Bilateral filtering: Theory and applications," *Foundations and Trends in Computer Graphics and Vision*, vol. 4, no. 1, pp. 1–73, 2009.

[14] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications," *Data Min. Knowl. Discov.*, vol. 2, no. 2, 1998.

[15] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *IEEE CVPR*, 2012.

[16] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *IEEE CVPR*, 2015.

[1]Supplemental material is published on IEEE Digital Repository.