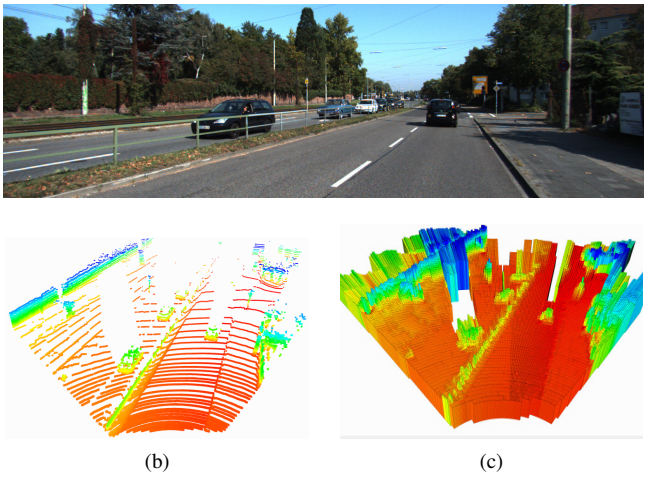
Polar-grid representation and Kriging-based 2.5D interpolation for urban environment modelling

Cristiano Premebida*†, João Sousa†, Luis Garrote*†, and Urbano Nunes*†
*Department of Electrical and Computer Engineering (DEEC)
†Institute of Systems and Robotics (ISR)
University of Coimbra, Portugal.
Emails: {cpremebida,joaosousa,garrote,urbano}@isr.uc.pt

***Abstract*—In this paper a spatial interpolation approach, based on polar-grid representation and Kriging predictor, is proposed for 3D point cloud sampling. Discrete grid representation is a widely used technique because of its simplicity and capacity of providing an efficient and compact representation, allowing subsequent applications such as artificial perception and autonomous navigation. Two-dimensional occupancy grid representations have been studied extensively in the past two decades, and recently 2.5D and 3D grid-based approaches dominate current applications. A key challenge in perception systems for vehicular applications is to balance low computational complexity and reliable data interpretation. To this end, this paper contributes with a discrete 2.5D polar-grid that upsamples the input data, *i.e.* sparse 3D point cloud, by means of a deformable Kriging-based interpolation strategy. Experiments carried out on the KITTI dataset, using data from a LIDAR, demonstrate that the approach proposed in this work allows a proper representation of urban environments.**



(b) (c)

Fig. 1. Image-frame from KITTI database (first row), the respective high-resolution 3D point cloud from Velodyne (b), and interpolated 2.5D grid (c).

I. Introduction

Spatial modelling of the surrounding environment, using data from sensors mounted onboard an intelligent vehicle, is a research topic that impacts on many applications, such as collision prediction, object detection, obstacle avoidance, road detection, and pedestrian protection systems. Environment modelling for mobile robotics and intelligent vehicle applications has been addressed by many research groups [1][2][3][4], and the approaches differ in many aspects such as: 1) spatio or spatio-temporal representations; 2) 2D, 2.5D or 3D models; 3) the type of sensory-data being processed; 4) and estimation techniques employed.

Essentially, this work deals with discrete spatial representation for environment modelling based on data from a high-resolution LIDAR sensor. More specifically, an instrumented robotic-vehicle collecting high resolution 3D data, see Fig. 1(b), and driving in urban traffic conditions is the case of interest in this paper, where the data comes from a Velodyne HDL-64 mounted on the roof of the vehicle [5][6]. Although many of the research works related to 3D range data interpolation have the goal of combining range data with texture information from a camera [7][8], this paper proposes a spatial representation and interpolation approach that relies on only LIDAR data. For that purpose, a dense 2.5D polar-grid is used, where each cell represents an estimated elevation (height) obtained by a Kriging-based interpolation technique applied to the input 3D-data as shown in Fig. 1(c).

Most of the related work on using LIDAR-based grid representations aim to solve the segmentation/clustering problem, as in [9][10][11]. This is justified by the fact that many applications in autonomous robotic-vehicles depend, directly or indirectly, on a segmentation approach, *e.g.* object detection [11], road detection [12], and detection and tracking of moving objects (DATMO) [13]. In common, LIDAR-based grids have the purpose of obtaining a compact and efficient representation to support real-world tasks, and therefore it is important to guarantee low computational complexity; among other requirements.

This work in particular contributes with a method to obtain a 'dense' 2.5D grid, *i.e.* the information in the discrete grid is enhanced (upsampled), in the sense that each cell is updated not only by the points that fall within the cell, but by a sliding-window technique where a deformable local-region gathers data from a set of neighbors cells (within the region). As consequence, the grid is upsampled and smoothed simultaneously. Furthermore, an outlier detection algorithm is applied in such local-region to improve the resulting grid.

The remainder of this paper is organized as follows, in the next section the polar grid representation is described, Section III briefly details the ordinary Kriging technique, while in Section III-A the proposed deformable polar grid approach is presented, followed by a description of an outlier detection method in III-B. Experimental results are reported in Section IV regarding different implementations of 2.5D grids for urban

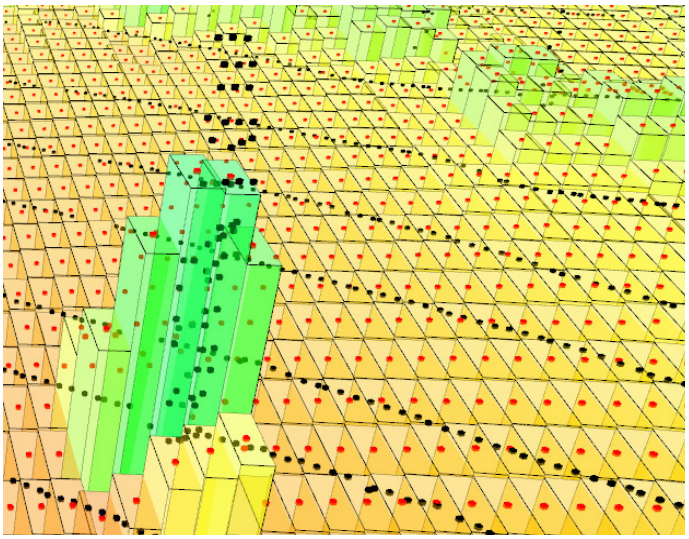


Fig. 2. Polar-grid construction for 2.5D representation, where the estimated elevation value (red) of a given cell is obtained by interpolating the 'height' (z_j) components of the 3D points (black).

scenarios representations. Finally, Section V concludes this paper.

II. Polar Grid

In this section, an approach to generate a 2.5D (or elevation map as in [14]) polar-grid for environment representation is introduced. Such environment representation models the relationships between the 3D-LIDAR measurements and the surrounding. As the requirements made by the ITS/IV industry increase, more precise environment representations (such as the ones that represent 3D environments) are required to better represent the physical world as well as to provide a solid base for advanced tasks an IV can execute. However, constraints related to computational complexity and task execution time need to be met, creating a tradeoff between the model quality and its computational complexity. Both 2.5D and 3D models can be employed to represent the vehicle's outdoor environment but 2.5D models provide a smaller computational burden.

Multiple representations have been proposed to address the problem of environment representation in 2.5D models (also known as height maps), with Cartesian-grids [1], [15] being one of the most common representations. The advent, in the last years, of 3D high-end sensors such as the Velodyne HDL-64E have demanded new solutions to address the problem of dense 3D-based environment modeling. The measurements provided by such sensors, *i.e.* dense 3D point clouds, have fixed vertical angular resolution and hence radial dispersion is significant; this aspect becomes more evident as the distance from the LIDAR increases. Consequently, converting a 3D point cloud into a Cartesian-grid produces a sparse grid representation. On the other hand, polar and/or log-polar (logarithmic range resolution) grids [16][17][18] can be applied to mitigate this problem as they can adapt to the sensor's radial dispersion and, as a consequence, more suitable representations of the environment are likely to be attained.

The polar-grid proposed in this paper is represented by a two-dimensional grid composed by $m_r \times m_\theta$ cells, where each

Algorithm 1 Conversion of 3D point-cloud to a polar-grid.

Input: Point Cloud: **P**
Output: Polar Grid: **G**
1: Polar Grid Dimensions (m_r,m_θ)
2: **for all** $(\mathbf{p} = (x,y,z)^T \in \mathbf{P})$ **do**
3: $(r,\theta) \leftarrow$ **pointToPolar**(x,y)
4: $(r^i,\theta^i) \leftarrow$ **polarToIndex**(r,θ)
5: $G(r^i,\theta^i) \leftarrow G(r^i,\theta^i) \cup \{r,\theta,z\}$
6: **end for**

cell is defined by a radial (r) and angular (θ) components. Given a set of 3D Cartesian points $(\mathbf{p}_i = (x_i,y_i,z_i)^T \quad i = 1,\cdots,n)$ the value of a given *cell* is obtained by an estimation of the 'height' points $z_j, j \in cell$, as illustrated in Fig. 2.

Each point $\mathbf{p}_i$ in the 3D point cloud (**P**) is converted to polar coordinates (r,θ) in a straightforward process (**pointToPolar** in Algorithm 1), where $(r,\theta) = (\sqrt{x^2+y^2}, arctan(\frac{y}{x}))$. To build a polar-grid, a correspondence between a polar value (r,θ) and a grid index (**polarToIndex** in Algorithm 1) is given by:

$$r_i(r) = \begin{cases} round(\frac{r-r_{min}}{\Delta r}) & \text{if } r_{min} \leq r \leq r_{max} \\ -1 & \text{otherwise} \end{cases} \quad (1)$$

with r_{min} the minimum radius, Δr the step factor, and r_{max} the maximum radius, usually up to the maximum LIDAR's range. The angular variable is calculated as:

$$\theta_i(\theta) = \begin{cases} round(\frac{\theta-\theta_{min}}{\Delta\theta}) & \text{if } \theta_{min} \leq \theta \leq \theta_{max} \\ -1 & \text{otherwise} \end{cases} \quad (2)$$

where θ_{min} is the minimum angle, $\Delta\theta$ the step factor and θ_{max} the maximum angle allowed.

Conversely, the process of converting the indices back to the polar value w.r.t. the cell center (r_c,θ_c **indexToCenterCell**[1]) is given by:

$$r_c(i) = \begin{cases} r_{min} + i\Delta r - \frac{\Delta r}{2} & \text{if } 0 \leq i \leq m_r \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$\theta_c(i) = \begin{cases} \theta_{min} + i\Delta\theta - \frac{\Delta\theta}{2} & \text{if } 0 \leq i \leq m_\theta \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Basically, Algorithm 1 is used to convert the Cartesian coordinates (x,y) of **P** to Polar coordinates (r,θ) and to obtain a direct correspondence between the indices of the polar-cells and the coordinates (r,θ). During this process, the intensity (height) value of the cells are not estimated. The height estimation algorithm and the final 2.5D polar-grid method are presented in the next sections.

III. Ordinary Kriging

In order to estimate the heights' values of the cells $\in \mathbf{G}$, ordinary Kriging is here used as local estimator where samples within a target area (or neighborhood region $\mathcal{N}$), are weighted to estimate the value of a given cell centered in such region. The Kriging estimator is an optimal linear predictor, *i.e.* BLUP

[1] See Algorithm 2.

(best linear unbiased predictor) [19], that predicts the value of a random function $Z = Z(x)$ at the location of interest x_0 from the samples (measured observations) $z(x_i)$ of $Z(x)$, where $x_i \in \mathcal{N}$, *i.e.* the considered sample points x_i used to estimate $Z(x_0)$ are within the local region $\mathcal{N}$. The points x_i are weighted according to a spatial structure, modeled via a covariance function or the equivalent variogram [20].

The ordinary Kriging predicts the value $Z^*(x_0)$ by a linear combination of $Z(x)$ evaluated at the samples $x_i \in \mathcal{N}, i = 1, \cdots, n$ as:

$$Z^*(x_0) = \sum_{i=1}^{n} \omega_i Z(x_i) = \boldsymbol{\omega}^T \mathbf{Z}, \tag{5}$$

where $\boldsymbol{\omega} = (\omega_1, \cdots, \omega_n)^T$ are the weights that dictate the influence of the variables $Z(x_1, \cdots, x_n)$ in the estimation of $Z(x_0)$. The unbiasedness condition of the Kriging predictor is ensured by $\sum \omega_i = 1$ and by the assumption that the mean is intrinsically stationary, *i.e.* $\mathbb{E}[Z(x_i) - Z(x_0)] = 0, \forall i$.

Ordinary Kriging assumes that the mean of $Z(x)$ is unknown but constant at the local neighborhood $\mathcal{N}$ and the covariance function $C(h)$, or the variogram function $\gamma(h)$, is known given a lag distance h. The goal of Kriging is to calculated the weights $\boldsymbol{\omega}$ such that the error variance of the estimator σ_E^2 is minimized s.t. $\boldsymbol{\omega}^T \mathbb{1} = 1$:

$$\sigma_E^2 = Var(Z^*(x_0) - Z(x_0)) = \mathbb{E}[(\textstyle\sum_{i=1}^{n} \omega_i Z(x_i) - Z(x_0))^2] =$$
$$Var(Z(x_0)) + Var(\textstyle\sum_{i=1}^{n} \omega_i Z(x_i)) - 2Cov(\sum_{i=1}^{n} Z(x_i), Z(x_0)) \tag{6}$$

To simplify the notation, $C_{ij} := Cov(Z(x_i), Z(x_j))$, and in accordance with [20], (6) becomes:

$$\sigma_E^2 = \sigma_0^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i \omega_j C_{ij} - 2 \sum_{i=1}^{n} \omega_i C_{i0}, \tag{7}$$

where $C_{i0} = Cov(Z(x_i), Z(x_0))$. The Lagrange multipliers method is used to minimize (7) s.t. $\sum_i^n \omega_i = 1$. Making $\mathscr{L} = Var(Z^*(x_0) - Z(x_0)) + 2\lambda(\sum \omega_i - 1)$, where λ is the Lagrange multiplier, the minimization problem becomes:

$$\underset{\omega}{\text{minimize}}\ \mathscr{L}, \text{ s.t.} \sum_{i}^{n} \omega_i = 1 \tag{8}$$

where $\mathscr{L}$ can be expressed more conveniently by:

$$\mathscr{L} = \sigma_0^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i \omega_j C_{ij} - 2 \sum_{i=1}^{n} \omega_i C_{i0} + \underbrace{2\lambda(\sum_{i}^{n} \omega_i - 1)}_{0}. \tag{9}$$

Taking the partial derivatives of $\mathscr{L}$ w.r.t. the weights and setting these derivatives to zero results in:

$$\begin{pmatrix} \sum_{i=1}^{n} \omega_i C_{1i} + \lambda = C_{10} \\ \sum_{i=1}^{n} \omega_i C_{2i} + \lambda = C_{20} \\ \vdots \\ \sum_{i=1}^{n} \omega_i C_{ni} + \lambda = C_{n0} \\ \sum_{i=1}^{n} \omega_i \cdot 1 + 0 = 1 \end{pmatrix} \tag{10}$$

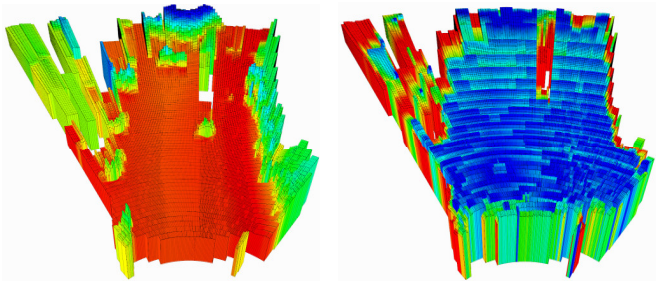


Fig. 3. Interpolated polar-grid (left), and estimated error variance σ_E^2, per cell, in the right.

that can be rearranged in terms of a system of linear equations,

$$\begin{pmatrix} C_{11} & \cdots & C_{1n} & 1 \\ C_{21} & \cdots & C_{2n} & 1 \\ \vdots & \cdots & \vdots & \vdots \\ C_{n1} & \cdots & C_{nn} & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \\ \lambda \end{pmatrix} = \begin{pmatrix} C_{10} \\ C_{20} \\ \vdots \\ C_{n0} \\ 1 \end{pmatrix}. \tag{11}$$

Finally, the weights $\boldsymbol{\omega}$ can be calculated from:

$$\mathbf{C} \cdot \mathbf{w} = \mathbf{D} \Rightarrow \mathbf{w} = \mathbf{C}^{-1}\mathbf{D}, \tag{12}$$

where $\mathbf{w} = [\omega_1, \cdots, \omega_n | \lambda]^T = [\boldsymbol{\omega} | \lambda]^T$, $\mathbf{C}$ is the leftmost matrix and $\mathbf{D}$ is the vector of covariances of the second member of (11).

The values of $\boldsymbol{\omega}$, calculated by solving (12), are then used in (5) to obtain an estimated value of the variable of interest $Z(x_0)$ with the minimum error variance σ_E^2. In this work, the desirable variables to be estimated are the values of the cells in the polar-grid. As an example, Fig.3 (right) shows the error variances calculated for a given grid (left), where dark blue cells manifest a higher level of confidence (lower errors), and red designates cells with higher error variances. Basically, zones in the grid with less measurement points tend to increase the uncertainty. We can anticipate that the polar-grid using Kriging provides a more suitable representation due to the robustness of the Kriging technique, which is an unbiased (zero mean error) and best (in the sense that the error variance is minimized) linear method.

A. *Interpolation of the polar-grid*

The estimation of the cell's height values in the polar-grid described in Section II is performed using the Kriging method discussed above. Considering a 3D point cloud ($\mathbf{P}$) as input, a polar grid $\mathbf{G}$ is calculated using Algorithm 1, where $\mathbf{G}$ contains the correspondences between the points in $\mathbf{P}$ and the cells in the grid. In order to estimate the cell's height values in the polar-grid, the Kriging method requires that a region of interest (ROI), *i.e.* a neighborhood area, to be defined. In this work, a variable-sized sliding-window approach is adopted to compute the ROI in a such way that it takes into consideration the geometrical model that governs, in a simplified manner, the LIDAR-scans as they reflect on the ground (as represented in Fig. 4). Furthermore, the parameters that defines a ROI vary according to the range position in the grid.

Figure 4 depicts a simplified representation of the LIDAR behavior because the ground is assumed flat and non-

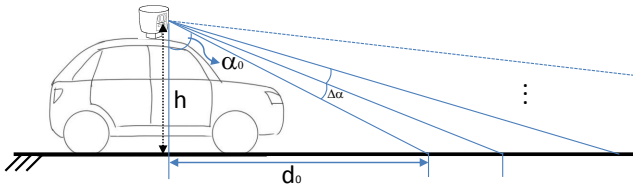


Fig. 4. Illustration of the distance d_0 that presumably the first laser-scan would hit a flat ground, where $d_0 = h \cdot tan(\alpha_0)$.

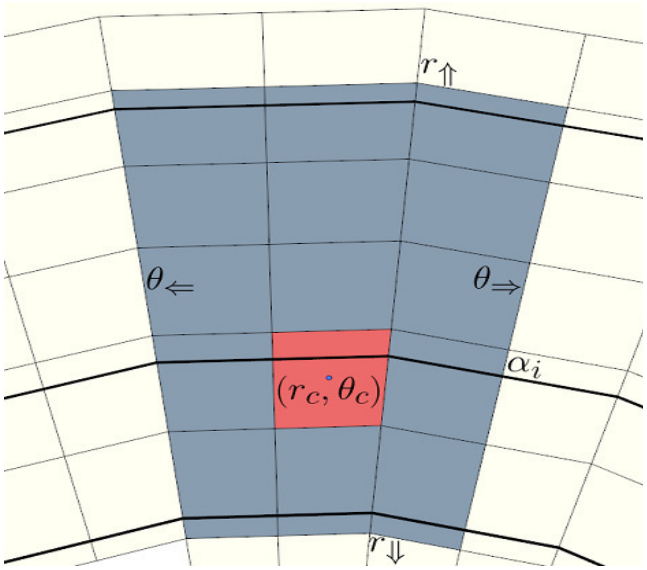


Fig. 5. ROI for an arbitrary cell (r_c, θ_c). The boundaries of ROI vary according to the parameters $(\theta\Leftarrow, \theta_{\Rightarrow}, r_{\Downarrow}, r_{\Uparrow})$; see text for details.

deformable, oscillations in the vehicle and in the sensor-platform are disregarded, and the measurement errors are assumed negligible. Based on these assumptions, the radial component increases as function of $tan(\cdot)$ in order to compensate the radial dispersion of the LIDAR-scans which follows a tangent function law, *e.g.* $d_0 = h \cdot tan(\alpha_0)$ as illustrated in Fig. 4. On the other hand, the angular size of the ROI is defined constant (left and right boundaries $(\theta\Leftarrow, \theta_{\Rightarrow})$) regardless the position in the polar-grid.

For a given cell-indexes $\{i, j\}$ and corresponding polar space coordinates (r_c, θ_c), according to the parameters depicted in Fig. 4, the corresponding α_r is given by:

$$\alpha_r = arctan(\frac{r_c}{h}), \tag{13}$$

where h is is the height of the LIDAR to the terrain. The corresponding scan's angular-identifier $(\alpha_i, i = 1, \cdots, 64)$ is given by:

$$\alpha_i = round(\frac{\alpha_r - \alpha_0}{\Delta\alpha}). \tag{14}$$

From the radial dispersion of the LIDAR scans, the upper and lower limits of the ROI (see Fig. 5) are given by:

$$\begin{aligned} r_{\Downarrow} &= h \cdot tan(\alpha_0 + (\alpha_i - N + 1)\Delta\alpha) \\ r_{\Uparrow} &= h \cdot tan(\alpha_0 + (\alpha_i + N)\Delta\alpha) \end{aligned} \tag{15}$$

Algorithm 2 Proposed 2.5D polar-grid interpolation using Kriging.

Input: Point Cloud: **P**
Output: Interpolated Polar Grid: **G**

```
Polar Grid Dimensions (m_r, m_θ)
G ← PolarGrid(P, m_r, m_θ)
G ← OutlierRemover(G)
for all (cell_{i,j} ∈ G) do
    (r^i_min, r^i_max, θ^j_min, θ^j_max) ← ROI(cell_{i,j})
    Z ← (r^i_min, r^i_max, θ^j_min, θ^j_max)
    if Z = ∅ then
        cell_{i,j} ← ∅
    else
        (r_c, θ_c) ← indexToCenterCell(cell_{i,j})
        cell_{i,j} ← Kriging(Z, r_c, θ_c)
    end if
end for
```

where N is the number of laser scans $\in$ ROI, $\Delta\alpha \approx 0.4°$, and $h \approx 1.73$m. Finally, the intensity value of each cell, *i.e.* its height, is estimated by Kriging the LIDAR-points **Z** according to (5). The proposed algorithm is summarized in Algorithm 2.

B. Outlier removal

In Algorithm 2, the function **OutlierRemover** uses the interquartile range (IQR) method to detect outliers that may occur in the LIDAR measurement points. Denoting by $\mathbf{g}_j = (r_j, \theta_j, z_j)^T$, with $j = 1, \cdots, m$, the set of LIDAR-points inside a given cell in the grid $G(r^i, \theta^i)$, the first quartile ($Q_{25\%}$) was calculated by taking the median of the lower half of $z \in \mathbf{g}_j$, while the third quartile ($Q_{75\%}$) is the median of the upper half. It is assumed that $\mathbf{g}$ is ordered in terms of z, and the median value $M(z)$ was considered in the calculations of the quartiles. $M(z)$, which divides $\mathbf{g}$ in two halves, was calculated in terms of the z-value of the LIDAR-points. In our implementations, if $m < 5$ the IQR method was not applied.

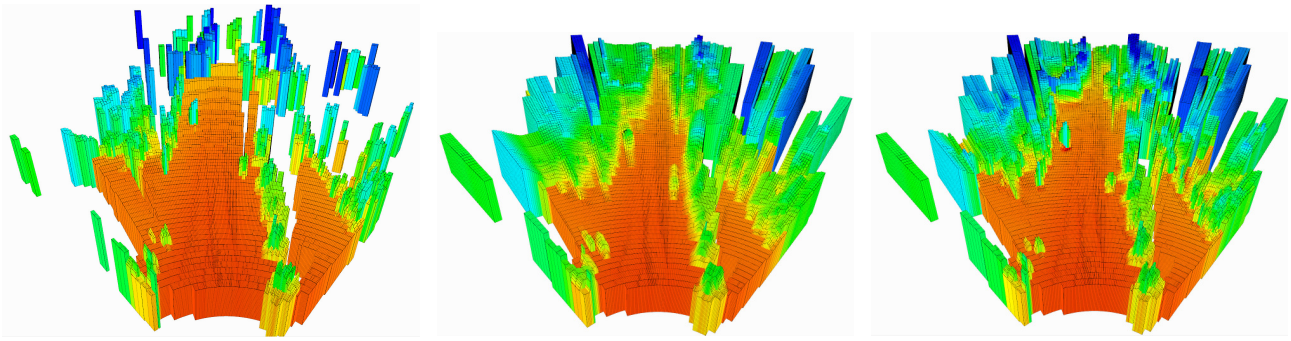
Knowing that $IQR = Q_{75\%} - Q_{25\%}$, the z elements of $\mathbf{g}$ that fall below $Q_{25\%} - 1.5 \cdot IQR$ or above $Q_{75\%} + 1.5 \cdot IQR$ are considered outliers and then are removed. Some experiments using the 3σ-rule were carried out, but the results obtained with IQR were better than the 3σ-rule.

IV. Experiments and Discussion

Experiments on real-world data, from the KITTI dataset [5], are presented in the sequel. The KITTI dataset was recorded in urban-like scenarios using a multi-sensor platform mounted onboard an instrumented-vehicle. In this work, only point clouds from a LIDAR (Velodyne HDL-64E) are considered. Using such point clouds as input, four approaches were studied for discrete 2.5D environment representation: Cartesian grid (CG), polar-grid (PG), polar-grid using IDW[2] interpolation (PG_{IDW}), and polar-grid using Kriging (PG_{KRI}). Except in PG_{IDW}, Kriging-based interpolation was employed on all the aforementioned approaches. The experiments reported in this section were performed with C++ implementations, while Point Cloud Library (PCL[3]) was used for clustering.

[2]IDW denotes Inverse Distance Weighting.
[3]http://www.pointclouds.org/documentation/tutorials/cluster_extraction.php

(a) Frame 000085 from KITTI (b) 3D Point cloud (c) *CG*: Cartesian grid



(d) *PG*: with non-deformable ROI (e) PG_{IDW}: using IDW method (f) PG_{KRI}: using Kriging

Fig. 6. Results using different algorithms for 2.5D grid representation and interpolation. For the frame 000085 from the KITTI dataset, with image from camera_2 (a) and corresponding 3D point cloud from Velodyne (b), four cases are shown: a Cartesian grid using fixed-size ROI and Kriging (c), a polar-grid using fixed-size ROI and Kriging (d), a polar-grid using deformable ROI and IDW interpolation (e), and finally a polar-grid using deformable ROI and Kriging.

A. *Qualitative results*

Figure 6 shows, for a given scenario, the resulting 2.5D environment representation using *CG* (c), *PG* (d), PG_{IDW} (e), and PG_{KRI} (f). An image of the scene (a) is provided to facilitate the understanding of the environment, while the input 3D point cloud is shown in (b). Notice that the polar-grids are shown in Cartesian coordinates, to make the analysis easier to follow. Although *CG* is the most common approach in mobile robotics and IV/ITS applications, the referred Cartesian grid has inconsistencies. Assuming that the LIDAR measurements follow discrete spherical coordinates, when the scans hit the terrain the resulting points follow a Polar representation (as discussed in sect. II). Therefore, the resulting polar-grids are more appropriate than the *CG*.

The *PG*, in Fig.6(d), obtained by interpolating the point-cloud with a fixed-size ROI performs quite similar as *CG*. However, when the proposed 'deformable' ROI is used, the output polar-grids are more reliable, *e.g.* in Figs. 6(e) and 6(f). Such 'non-fixed size' ROI technique achieved very good results because it captures the underlying behavior of the LIDAR scans as function of the distance, which is incorporated in the parameters $(r_{\Downarrow}, r_{\Uparrow})$ of the ROI, as described in Subsection III-A.

An additional study was performed to compare the Kriging with the IDW interpolation algorithm. In short, the implemented IDW algorithm weights the z components of the 3D points within the ROI as function of the inverse of the Euclidean distance to the center of the target-cell. Quantitative comparison between PG_{IDW} and PG_{KRI} is reported in the next section, where both interpolation methods were implemented considering a polar-grid with deformable ROI.

B. *Quantitative results*

As discussed earlier, grid-based representation is useful in many applications in robotics and IV/ITS. In this paper, however, 2.5D polar-grid will be evaluated in terms of object/obstacle detection, because it is a prerequisite in collision avoidance, protection systems, DATMO, and other tasks. For this end, a PCL-based clustering was used to segment the data in the 2.5D grids. The output segments are then compared against ground-truth objects from the KITTI *object detection dataset*. Although KITTI database does not provide a general object/obstacle detection benchmark, we decided to consider all object categories in our experiment, *i.e.* "Pedestrian", "Person_sitting", "Cyclist", "Misc", "Car", "Truck", "Tram", "Van". For obvious reasons, objects labeled as "DontCare" were ignored. Denoting by O a given labeled object, and S a segment/cluster, precision (*Pre*), recall (*Rec*) and Jaccard (*Jac*) performance measures are calculated as: $Rec = \frac{O\ capS}{|O|}$, $Pre = \frac{O \cap S}{|S|}$, and $Jac = \frac{O \cap S}{O \cup S}$, where $(O \cap S)$ is the overlap area between O and S, $(O \cup S)$ is the area of union, and $|\cdot|$ is the restrict area of a given element. *Rec*, *Pre* and *Jac* were normalized by the number of objects.

It is important to note that these performance-measures were defined in terms of pixel bounding boxes; thus, the segments extracted from the 2.5D grid locations are transformed to pixel coordinates using the calibration parameters between camera_2 and the LIDAR, as detailed in [5]. For the results shown in Fig.7, with threshold values $[\frac{thr}{10} | thr \in \mathbb{N}, thr \leq 10]$, we considered 209 frames and 1042 objects from the KITTI training-set. These results were restricted to the PG_{IDW} and PG_{KRI} approaches because they achieved better representations. For $thr = 0.5$, which is the usual threshold value adopted in this field, the *IDW* interpolation approach achieved 42.9%

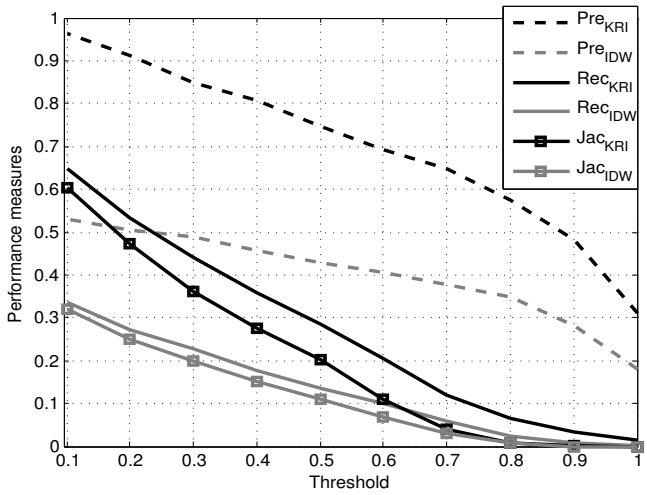


Fig. 7. Precision, recall and Jaccard curves, per-threshold value, for PG_{KRI} (black) and PG_{IDW} (grey).

precision, 13.6% recall and 10.9% Jaccard. The polar-grid-Kriging performance is better, with 74.6%, 28.4% and 20.3% respectively. Overall, we found best performance by using PG_{KRI} to represent the evaluated urban scenarios.

V. Conclusion

Discrete representation of urban scenarios is a problem of interest in many real-world applications, for instance collision avoidance, path planning, protection systems, and ADAS in general. Here, this problem is particularly addressed in terms of 2.5D grid-based representations for environment modelling. To this end, and considering measurements (in the form of a point cloud) from a 3D LIDAR mounted on the roof of an instrumented vehicle, a deformable polar-grid approach is proposed. Firstly, a discrete grid is generated in Polar coordinates, then the input 3D point cloud is projected on the grid, followed by an outlier removal algorithm, and finally a Kriging-based interpolation method is used to filter and to predict the values (height) of the cells on the grid. The interpolation process is locally constrained by a auto-adjust region. Experiments on point clouds from a state-of-the-art dataset, the KITTI database, are reported using four solutions: an usual Cartesian grid, a polar-grid, a polar-grid interpolated with IDW, and the proposed Kriging-based method. The algorithms were implemented in C++, and all the grids were interpolated, *i.e.* 2.5D 'dense' grids were considered in the experiments. Based on the reported results, we found the proposed solution to be the most suitable approach for environment modelling. As future work we plan to extend our work to consider higher-level applications, *e.g.* object detection and collision avoidance.

Acknowledgment

This work has been supported by the FCT project "AMSHMI2012-RECI/EEIAUT/0181/2012" and project "ProjBDiagnosis and Assisted Mobility - Centro-07-ST24-FEDER-002028" with FEDER funding, programs QREN and COMPETE. We also thank the reviewers for their valuable comments and suggestions.

References

[1] R. Triebel, P. Pfaff, and W. Burgard, "Multi-level surface maps for outdoor terrain mapping and loop closing," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Oct 2006.

[2] A. Broggi, E. Cardarelli, S. Cattani, and M. Sabbatelli, "Terrain mapping for off-road autonomous ground vehicles using rational B-spline surfaces and stereo vision," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 648–653.

[3] R. Pascoal, V. Santos, C. Premebida, and U. Nunes, "Simultaneous segmentation and superquadrics fitting in laser-range data," *Vehicular Technology, IEEE Transactions on*, vol. 64, no. 2, pp. 441–452, Feb 2015.

[4] J. Moras, V. Cherfaoui, and P. Bonnifait, "Credibilist occupancy grids for vehicle perception in dynamic environments," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 84–89.

[5] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[6] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[7] J. Dolson, J. Baek, C. Plagemann, and S. Thrun, "Upsampling range data in dynamic environments," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 1141–1148.

[8] H. Andreasson, R. Triebel, and A. Lilienthal, "Non-iterative vision-based interpolation of 3D laser scans," in *Autonomous Robots and Agents*, ser. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2007, vol. 76, pp. 83–90.

[9] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion," in *Intelligent Vehicles Symposium, 2009 IEEE*, June 2009, pp. 215–220.

[10] M. Himmelsbach, F. v Hundelshausen, and H. Wuensche, "Fast segmentation of 3D point clouds for ground vehicles," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, June 2010, pp. 560–565.

[11] M. Himmelsbach, A. Mueller, T. Luettel, and H.-J. Wuensche, "LIDAR-based 3D object perception," in *Proceedings of 1st International Workshop on Cognition for Technical Systems*, Munich, Oct. 2008.

[12] F. Homm, N. Kaempchen, J. Ota, and D. Burschka, "Efficient occupancy grid computation on the GPU with lidar and radar for road boundary detection," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, June 2010, pp. 1006–1013.

[13] A. Azim and O. Aycard, "Detection, classification and tracking of moving objects in a 3D environment," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June 2012, pp. 802–807.

[14] J.-S. Gutmann, M. Fukuchi, and M. Fujita, "3D perception and environment map generation for humanoid robot navigation," *The International Journal of Robotics Research*, vol. 27, no. 10, pp. 1117–1134, 2008.

[15] P. Pfaff, R. Triebel, and W. Burgard, "An efficient extension to elevation maps for outdoor terrain mapping and loop closing," *The International Journal of Robotics Research*, vol. 26, no. 2, pp. 217–230, feb 2007.

[16] C. Tongtong, D. Bin, L. Daxue, Z. Bo, and L. Qixu, "3D LIDAR-based ground segmentation," in *Pattern Recognition (ACPR), 2011 First Asian Conference on*, Nov 2011, pp. 446–450.

[17] P. Sermanet, P. Sermanet, R. Hadsell, M. Scoffier, M. Scoffier, U. Muller, and Y. LeCun, "Mapping and planning under uncertainty in mobile robots with long-range perception," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept 2008, pp. 2525–2530.

[18] Y. Yi, Y. Guang, Z. Hao, F. Meng-Yin, and W. Mei-ling, "Moving object detection under dynamic background in 3D range data," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, June 2014, pp. 394–399.

[19] A. Lichtenstern, "Kriging methods in spatial statistics," Master's thesis, Technische Universitat Munchen, Department of Mathematics, 2013.

[20] E. H. Isaaks and R. M. Srivastava, *Applied Geostatistics*. Oxford University Press, 1989.