Road Detection Using High Resolution LIDAR

R. Fernandes*, C. Premebida*, P. Peixoto*, D. Wolf †, U. Nunes*
* Institute of Systems and Robotics, Electrical and Computer Engineering Department,
University of Coimbra, Portugal.
{rfernandes, cpremebida, peixoto, urbano}@isr.uc.pt
† Mobile Robotic Laboratory, ICMC, University of São Paulo, Brazil.
denis@icmc.usp.br

***Abstract*—This paper proposes a road detection approach based solely on dense 3D-LIDAR data. The approach is built up of four stages: (1) 3D-LIDAR points are projected to a 2D reference plane; then, (2) dense height maps are computed using an upsampling method; (3) applying a sliding-window technique in the upsampled maps, probability distributions of neighboring regions are compared according to a similarity measure; finally, (4) morphological operations are used to enhance performance against disturbances. Our detection approach does not depend on road marks, thus it is suitable for applications on rural areas and inner-city with unmarked roads. Experiments have been carried out in a wide variety of scenarios using the recent KITTI-ROAD benchmark [1], obtaining promising results when compared to other state-of-art approaches.**

I. Introduction

Detection of road regions ahead of a vehicle in real-world conditions is an important research problem for vehicular and mobile robotics applications. Considering safety requirements, high reliability demands, and the large diversity in realistic conditions, road detection becomes a very challenging task. Road terrains can vary significantly, for instance the pavement can be made of different materials or have different patterns. Besides, there can be changes due to weather conditions, light conditions or shadows. It is also common that road boundaries are not detectable because they either don't exist or are occluded by objects or cars on the road. All these factors make road detection a very difficult recognition problem.

Several road detection systems have been developed over the last decade. A survey on recent progress in road and lane detection is given in [2]. According to [2], and considering usual sensing technologies, most road detection systems are based on camera (vision), LIDAR, or a fusion of both sensors. Vision based methods are the most popular, mainly due to the presence of visual cues and landmarks, such as [3], [4]; in particular, Kuhnl *et al.* [5] use spatial ray features (SPRAY), extracted from three confidence maps, as the input to a boosting classifier. A large number of approaches explore the longitudinal patterns of roads: line markings, wheel tracks, and the road edge [6], [7]. However, most of these approaches do not deal with occlusions, such as a vehicle in front of the camera, neither provide a solution for unmarked roads.

High resolution 3D-LIDAR such as the Velodyne HDL-64E enables accurate dense depth measures in real-time, with ranges upwards of 50 meters, being effective under most operating conditions (namely surface textures, shadows and different light conditions), thus making this a promising sensor for road detection. Most LIDAR-based road detection approaches rely on a more or less complex model of the road, as in [8]. Other approaches require a precise localization system obtained using a high-performance Global Positioning System (GPS) and/or an Inertial Measurement Unit (IMU) [9] and [10]. On the other hand, solutions using a combination of LIDAR and vision are reported in [8] and [11]. Finally, in [12] the authors study the possibility of replacing a laserscanner by a stereo camera; they conclude that this approach has unsatisfactory performance when the surface has very low texture – as is the case of most road surfaces. Besides, stereo generally has lower precision in long range than LIDAR.

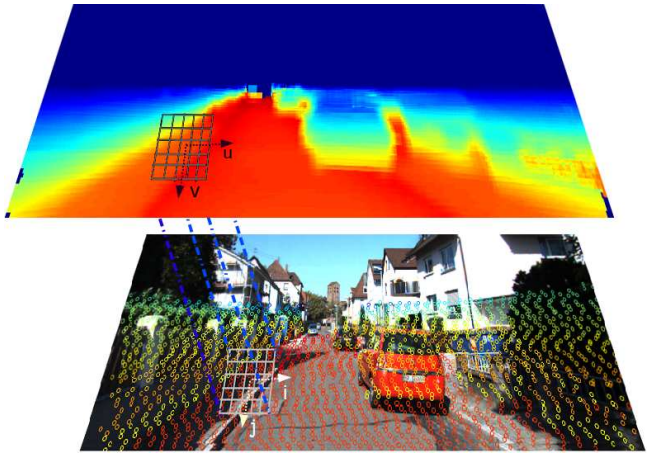


Fig. 1. Illustrative representation of a dense-height map, depicted in the top, obtained from upsampling LIDAR points. Bellow, it is a sparse set of points projected to the image plane. The grid represents a neighborhood mask (kernel) used in the proposed upsampling method. Here, colors represent different heights: red and orange are regions with low height, flowed by yellow, and so on and so forth.

Using 3D-LIDAR data only, and assuming the LIDAR sensor is calibrated with respect to a 2D camera-reference plane, an upsampling method was developed in order to create a dense height-map (elevation-map) out of the sparse and noisy 3D point-cloud (see Fig.1). Based solely on such high-resolution height map, a new road detection approach is presented in this paper. Experimental tests have been carried out in a wide variety of scenarios using the KITTI-ROAD benchmark [1], where the performance of our method is compared with state-of-the-art methods for road detection. In Fig. 1, an example of a RGB image (bottom) from KITTI database, and a high-resolution elevation map (top) generated using our upsampling method are shown; the sparse point-cloud, projected in the image plane, were colored as function of the height.

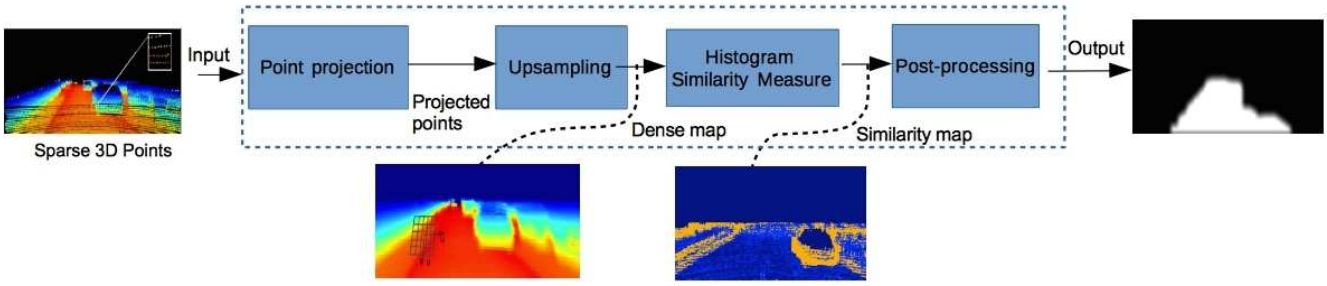


Fig. 2. Overview of the proposed road detection approach. Four main stages are involved: (1) projection of 3D-LIDAR points into a 2D reference plane; (2) upsampling the sparse points to a dense (high resolution) height map; (3) sliding-window technique that uses a similarity measure over neighboring regions in the height map; and (4) threshold and final morphological operations.

The structure of the paper is as follows: section II details our approach, including: a briefly explanation of the projection of 3D points to a 2D plane; followed by a description of our upsampling method; the sliding-window technique, presented in section II-C, uses height probability density functions of neighboring regions which are compared according to a similarity measure; finally, in section II-D the post-processing stage is described. Experimental results on the KITTI-ROAD benchmark and discussions can be found in section III, before the paper concludes in section IV.

II. Road Area Estimation

The overall flow of the road detection approach is depicted in Fig. 2, which is composed by four processing stages: (1) projection of 3D-LIDAR points into a 2D reference plane; (2) upsampling the sparse points to a dense (high resolution) height map; (3) sliding-window technique that uses a similarity measure over neighboring regions in the height map; and (4) final morphological operations.

A. Point projection

Taking advantage of the calibrated data provided in the KITTI benchmark, our approach uses the spatial-relationship between 3D points projected to a camera plane. The Velodyne HDL-64E S2, used in the benchmark, has 0.09 degree angular resolution, 2 cm distance accuracy, collecting around 1.3 million points/second. Scans are stored as floating points with $[x; y; z]$ coordinates (x = forward, y = left, z = up) [1]. The rigid body transformation from the Velodyne coordinates to camera coordinates is expressed by:

$$T_{velo}^{cam} = \begin{pmatrix} R_{cam}^{velo} & t_{cam}^{velo} \\ 0 & 1 \end{pmatrix} \quad (1)$$

where R_{cam}^{velo} and t_{cam}^{velo} are the rotation and translation matrices, respectively. Detailed information regarding LIDAR and camera calibration, data alignment, the calibration matrices in (1), and intrinsic and extrinsic parameters are given in [13]. A 3D point $X_v = (x, y, z, 1)^T$ in the LIDAR coordinates system gets projected to a point in the camera plane $X_c = (x, y, z, 1)^T$ according to:

$$X_c = T_{velo}^{cam} X_v. \quad (2)$$

Every point X_c is then rectified to match the image plane using a rectification matrix T_{rec}

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = T_{rec} X_c. \quad (3)$$

Considering the projected LIDAR points in pixel coordinates (u, v), as given by (3), some operations are performed in advance to the upsampling stage (described in the next section). Namely, the points outside the camera plane are discarded and the remaining points are sorted according to its position in pixel units, in order to speed up the search process. Finally, the points are rearranged to a new space that combines the coordinates in pixel units (u, v), the range r, and the height z, such that a point $\mathbf{P}$ is represented by $\mathbf{P} = (u, v, r, z)$.

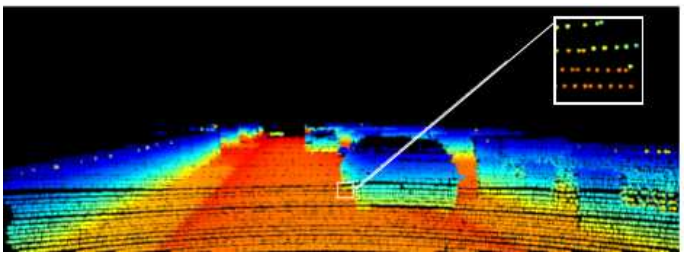


Fig. 3. Example of an image showing sparse and noisy LIDAR points from a Velodyne HDL-64E. The zoom highlights the sparsity of the LIDAR points projected into a high-resolution (1242 $\times$ 375 pixels) image plane.

B. Upsampling Sparse Range Data

Since LIDAR point clouds are sparse and noisy (as shown in Fig. 3), an upsampling method is applied in order to obtain a smooth (filtered) and dense map. For upsampling LIDAR range inputs we first used the method proposed by Dolson *et al.* [14] and, after some experimental evaluations, we decided to develop a method with the goal of obtaining high-resolution elevation maps conditioned on range data only. The Dolson's method, whose implementation code is available at [15], rely on the assumption that areas of similar texture in the camera image will have alike depth/range value. Moreover, most methods, such [14], are designed to solve the upsampling problem of sparse 3D points jointly using information from

intensity images. On the other hand, the method explained in this paper, which resembles in some way the method proposed in [14], uses only data from the 3D LIDAR. Furthermore, assuming the height (z-axis) values of the LIDAR points in the road are roughly constant in contrast to the depth values, we decided to create dense maps from the height z information instead of range r.

Let $\mathbf{P} = (u, v, r, z)$ denotes a calibrated set of 3D sparse LIDAR points projected to a camera plane as explained in sec. II-A. The value of the target dense map H, in a given position (u, v), is estimated by the weighted combination of the height values z of the sparse points $\mathbf{P}$ in a neighborhood, as follows:

$$H_{(u,v)} = \frac{1}{\alpha} \sum_{k \in \mathcal{N}(m)} w_k \cdot z_k \tag{4}$$

where the neighborhood $\mathcal{N}(m)$ is defined by the limited region within a mask m: with size 11×11 (in our case), and centered in position (u, v). In (4), α is a normalizing factor that ensures weights sum to one, *i.e.*, $\alpha = \sum w_k$.

Similarly to the bilateral filter, which was first described in [16] and then used in [17] to upsample low resolution images, each weight w_k is determined by two factors:
- a pixel distance function $f()$ that considers the difference in position between the mask central point $\mathbf{Q}(u, v)$ and the points $\mathbf{P}(i, j)$ within the neighborhood $\mathcal{N}(m)$;
- and a confidence weighting term $g(r)$. In our case, $g(r)$ is calculated as a function of the measured range distance r, and normalized by the maximum range.

Thus, a 2D-spatial neighborhood filter is formulated as:

$$H_{(u,v)} = \frac{1}{\alpha} \sum_{(i,j) \in \mathcal{N}(M)} f(\mid P_{(i,j)} - Q_{(u,v)} \mid) \cdot g(r) \cdot \ z_{(i,j)} \tag{5}$$

where the distance function $f()$ is assumed to be the Euclidean distance between the coordinates in pixel units:

$$f(|P_{(i,j)} - Q_{(u,v)}|) = \sqrt{(i-u)^2 + (j-v)^2} \tag{6}$$

Knowing that LIDAR points are not error-free, namely the Velodyne HDL-64E S2 has 2.5 cm RMSE range accuracy and average 0.002 rad beam divergence which causes inherent uncertainty in the sensor returns [18], we have considered these uncertainties as function of the distance thus, the further the object is from the LIDAR, the greater is the error in the measured points $\mathbf{P}$. Having this effect in mind, the value of the range factor $g(r)$, in (5), decreases proportional with the distance, penalizing points as function of their distance from the LIDAR:

$$g(r) = \frac{1}{r/m_r} \tag{7}$$

where m_r represents the maximum range of LIDAR. Note that the number of elements inside the mask is not constant and depends on the 3D-clouds sparsity, and the pixel-positions (u, v) of $\mathbf{P}$ are non-integer values, as shown in Fig. 3.

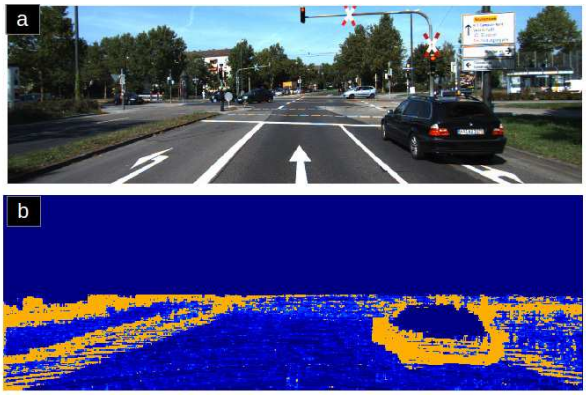


Fig. 4. (a) An example of an image from the KITTI-dataset. (b) The output image, where the objects are detected using the Bhattacharyya distance over the dense map. Blue represents values near 0 (*i.e.*, high similarity between neighbor regions), and orange values near 1 (*i.e.*, low similarity between neighbor regions).

C. Histogram Similarity Measure

It is assumed that the road is a smooth surface where two neighbor regions have small variation in height. Relying on this assumption, road segments and road-edge areas are identified using a measure of similarity between the height probability distribution of two neighbor regions. The values of height of those regions (or patches) are extracted from the dense map, obtained from (5).

A similarity measure mathematically determines the shortest distance between two observations in a high dimensional space. Various similarity/dissimilarity measures have been formulated throughout the years, each with its own strengths and weaknesses [19]. Among them, the Bhattacharyya distance is widely used and has been found to be more accurate for general purposes than a number of other measures [20]. Let v represent a given feature, p and q two discrete probability distributions over the same domain X. Then, the discrete form of Bhattacharyya coefficient is:

$$B[p_v(x), q_v(x)] = \sum_{x \in X} \sqrt{p_v(x) \cdot q_v(x)}. \tag{8}$$

Its metric form, as proposed in [21], is given by

$$D = \sqrt{1 - B[p_v(x), q_v(x)]} \tag{9}$$

where, in case of a complete mismatch (9) yields a value of 1, while maximum match yields 0.

The task of computing the height similarity between two neighbor regions can be described as follows: first, a sliding window of fixed size [$n \times m$] scans across the complete dense map (left-to-right-top-to-bottom), according to a step size Δ. For each patch extracted in the sliding window, a normalized histogram is computed over the flattened array of height values. Then, an empirical estimate of the probability density functions (*pdf*) is produced by dividing each bin of the histogram by the number of elements in each bin. Finally, the similarity between the actual *pdf* and the *pdf* of the previous patch is computed using (9).

(a) (b) (c) (d)

Fig. 5. In (a) we have an example of a road without obstacles, while in (b) and (c) there are vehicles on the road. Finally, in (d) there is a difficult case where the road in the left part is considered ground-truth, which we had a complete missing (red color).

The output is a map, as shown in Fig. 4, with values between 0 and 1, where values near 0 represent a region with smooth variations of height and values near 1 represent regions with much variation in height–likely an obstacle or a curb.

D. Post-processing

In this latest processing stage, a threshold is used to distinguish between objects, road-edges and estimate the road region delimited by them. This threshold is chosen in order to maximize the average precision of the global category in the training set. Furthermore, morphological operations [22] are performed in order to improve the quality of the segmentation. More specifically, morphological erosion with a structuring element of size 5×5 is used to remove small objects, followed by a morphological dilatation with a structuring element of size 7×7 connects regions that are close to each other.

III. Experiments

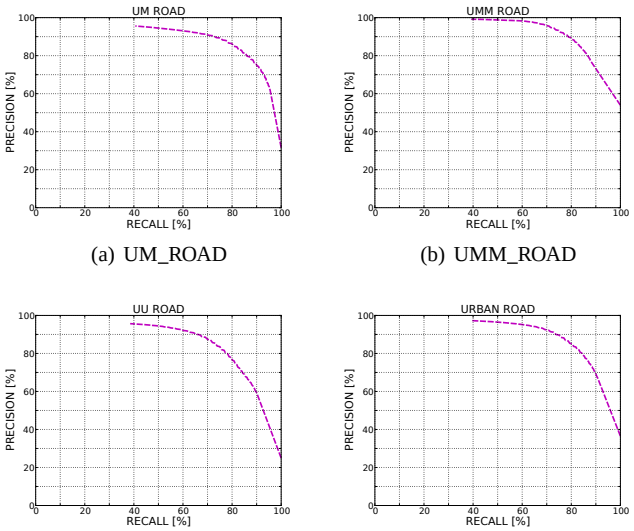
The performance of the road-detection approach was assessed using the KITTI-ROAD Benchmark Suite [1][1] which consists of 579 frames (rectified images with average spatial resolution of 1242 $\times$ 375 px), corresponding to 289 training frames and 290 testing frames. The dataset comprises three different categories of road scenes, as well as a global category combining all scenes. Table I summarizes the categories and the number of frames in each dataset.

TABLE I. KITTI-ROAD DATASET SUMMARY.

Scene	N.Train	N.Test	Short description
UM_ROAD	95	96	Urban Marked two-way road
UMM_ROAD	96	99	Urban Marked Multi-lane road
UU_ROAD	98	100	Urban Unmarked road
URBAN_ROAD	289	290	all urban scenes combined

Images and 3D-LIDAR scans of the KITTI-ROAD dataset were recorded from five different days on inner city (urban) roads. The training set comprises hand-labeled ground-truth annotations, while the testing set is evaluated using the online

[1]http://www.cvlibs.net/datasets/kitti/eval_road.php (Road)



(a) UM_ROAD (b) UMM_ROAD

(c) UU_ROAD (d) URBAN_ROAD

Fig. 6. Precision-Recall curves from the KITTI evaluation server, per urban scenes, obtained in the testing set.

KITTI evaluation server. Following the benchmark evaluation methodology, performance assessment is carried out in terms of the following measures: **MaxF** (Maximum value of F-measure), **AP** (Average Precision), **PRE** (Precision), **REC** (Recall), **FPR** (False positives rate) and **FNR** (False negatives rate). Further information of the dataset and details regarding the performance methodology are presented in [1], [13].

Results on the testing set for each urban scene are shown in Fig. 6, in terms of Precision-Recall, and summarized in Table II with percentage values of the performance measures. The reported results, obtained directly from the evaluation server, are consistent with other state-of-art methods (even though we use LIDAR data only) whose results are publicly available on the KITTI-ROAD website.

Figure 5 illustrates the performance of the method qualitatively on a set of test images. Detection errors occurred mainly

TABLE II. PERFORMANCE ON THE TESTING SET (FROM EVALUATION SERVER).

Scene	MaxF	AP	PRE	REC	FPR	FNR
UM	83.40 %	86.61 %	83.45 %	83.35 %	7.63 %	16.65 %
UMM	84.49 %	89.57 %	88.24 %	81.04 %	12.63 %	18.96 %
UU	79.34 %	80.04 %	82.25 %	76.63 %	5.50 %	23.37 %
URBAN	82.72 %	87.58 %	85.44 %	80.17 %	7.87 %	19.83 %

due to the following factors:

- The morphological operations used in post-processing stage may detach some areas to the road, while adding others that do not belong to the road;
- As already mentioned, the LIDAR measurements uncertainties grow as function of the distance. Therefore, road regions that are far from the vehicle can be hard to detect;
- Moreover, the benchmark performance criteria considered that, in some cases, road areas separated from the main road by a barrier (rail road, garden, etc.) should also be detected (see Fig.5(d)). However, our approach intends to detect only the road ahead of the vehicle, which increases the number of false negatives;
- Finally, since our detection approach does not depend on road marks, it may fail to detect some roads delimited only by marks.

IV. CONCLUSION

In this paper, we propose a road detection approach based on 3D-LIDAR data. We also propose an upsampling method, to create dense maps, that takes into account the uncertainty of the LIDAR readings as function of measured distances. Furthermore, our road detection solution relies on a similarity measure between neighbor regions on height dense map. Since the detection is based on region features, our detection method is robust against some variations over the road, such as unknown number of lanes or slopes.

The reported experiments in the KITTI-ROAD dataset show that LIDAR data may be very useful on road detection, even on unmarked roads. As future work, we plan to explore the LIDAR reflectivity information in order to detect lane markings. We also intend to use processing-time optimization techniques (such as GPU implementation).

ACKNOWLEDGMENT

This work was supported by project "ProjB-Diagnosis and Assisted Mobility", grant QREN Centro-07-ST24-FEDER-002028, and Portuguese Foundation for Science and Technology (FCT), and COMPETE program, under grant PDCS10:PTDC/EEA-AUT/113818/2009.

REFERENCES

[1] 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.

[2] A. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," in *Machine Vision and Applications*, vol. 25, April 2014, pp. 727–745.

[3] M. Felisa and P. Zani, "Robust monocular lane detection in urban environments," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, June 2010, pp. 591–596.

[4] Z. He, T. Wu, Z. Xiao, and H. He, "Robust road detection from a single image using road shape prior," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, Sept 2013, pp. 2757–2761.

[5] T. Kuhnl, F. Kummert, and J. Fritsch, "Spatial ray features for real-time ego-lane extraction," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, Sept 2012, pp. 288–293.

[6] M. Bertozzi and A. Broggi, "Gold: a parallel real-time stereo vision system for generic obstacle and lane detection," *Image Processing, IEEE Transactions on*, vol. 7, no. 1, pp. 62–81, Jan 1998.

[7] M. Darms, M. Komar, and S. Lueke, "Map based road boundary estimation," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, June 2010, pp. 609–614.

[8] W. Zhang, "Lidar-based road and road-edge detection," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, June 2010, pp. 845–848.

[9] J. Yoon and C. Crane, "Ladar based obstacle detection in an urban environment and its application in the darpa urban challenge," in *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, Oct 2008, pp. 581–585.

[10] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, "Junior: The stanford entry in the urban challenge," *Journal of Field Robotics*, 2008.

[11] Q. Li, L. Chen, M. Li, S.-L. Shaw, and A. Nuchter, "A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios," *Vehicular Technology, IEEE Transactions on*, vol. 63, no. 2, pp. 540–555, Feb 2014.

[12] M. Antunes, J. Barreto, C. Premebida, and U. Nunes, "Can stereo vision replace a laser rangefinder?" in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct 2012, pp. 5183–5190.

[13] 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.

[14] 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.

[15] J. Dolson, J. B., C. Plagemann, and S. Thrun, "Upsampling range data in dynamic environments (CPU code)," http://graphics.stanford.edu/papers/upsampling_cvpr10/ (accessed in July/2013).

[16] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*, Jan 1998, pp. 839–846.

[17] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Trans. Graph.*, vol. 26, no. 3, July 2007. [Online]. Available: http://doi.acm.org/10.1145/1276377.1276497

[18] C. Glennie and D. D. Lichti, "Temporal stability of the Velodyne HDL-64E S2 scanner for high accuracy scanning applications," *Remote Sensing*, vol. 3, no. 3, pp. 539–553, 2011. [Online]. Available: http://www.mdpi.com/2072-4292/3/3/539

[19] S.-H. Cha and S. Srihari, "Distance between histograms of angular measurements and its application to handwritten character similarity," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 2, 2000, pp. 21–24 vol.2.

[20] N. Thacker, F. Aherne, and P. Rockett, "The bhattacharyya metric as an absolute similarity measure for frequency coded data," in *Kybernetika*, vol. 32, June 1997, pp. 1–7.

[21] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, 2000, pp. 142–149 vol.2.

[22] J. Serra, *Image analysis and mathematical morphology: Theoretical advances*, ser. Image Analysis and Mathematical Morphology. Academic Press, 1988.