

Calibration and multi-sensor fusion for on-road obstacle detection

Luis Alberto Rosero¹ and Fernando Santos Osório²

Abstract—This article proposes a method for sensor calibration and obstacle detection in urban environments. The radar and 3D LIDAR were calibrated with relation to a stereo camera. Data from the radar, 3D LIDAR and stereo camera sensors were fused for the detection of obstacles and determination of their shapes. The data collected using both camera and 3D LIDAR are a point cloud that is segmented taking the radar detections as obstacle hypotheses. This sensor fusion approach decreases the processing, improves the obstacle detection and reduces false positives.

I. INTRODUCTION

Autonomous cars must know the position and dimension of obstacles for making decisions. Several methods as those that use point cloud segmentation have been proposed for on-road obstacle detection. Such clouds can be captured by 3D sensors, as LIDARs or stereo cameras. However these approaches require to process millions or at least thousands of points. This paper proposes a calibration method for sensors alignment and fusion that decreases the processing of on-road obstacle detection in urban environments. Two sensor pairs (radar-camera stereo and radar-3D LIDAR) were used. In our method, the radar generates obstacle hypotheses to accelerate and improve the segmentation process of point cloud generated by a 3D sensors (stereo camera or 3D LIDAR).

The advantages of automotive radars over other sensors include detection of obstacles remotely in its field of view, support to vibrations and operation under different lighting and weather conditions. Radars do not have moving parts and are not fine mechanical parts dependents. However, they neither report precise obstacle dimensions nor provide information only on position and velocity. An entire car is usually represented by a single point in the radar field of view.

Because of the limited determination of obstacle dimensions imposed by radar, we propose a radar-stereo camera or radar-LIDAR data fusion that estimates the dimensions and contours of the obstacles detected by the radar. The method improves detections and eliminates false positives generated by the use of a single sensor.

The sensors are calibrated for their alignment and achievement of data fusion for on-road obstacle detection. To calibrate sensors, we found the radar or a 3D LIDAR pose

relative to a stereo camera. Our sensor calibration method is computer-vision-based and uses an estimation method to solve the Perspective-n-Point problem (PnP) [1], which consist of n points in the 3D sensor coordinate system and their n correspondences in the 2D image plane. The solution for the PnP problem finds an $[R|t]$ matrix that maps points in the sensor coordinate system (radar or 3D LIDAR) to the camera coordinate system.

Once the sensors are aligned, the obstacles are segmented. Small point clouds are cut around each radar detection and for each small point cloud a Difference of Normals (DoN) [2] filter is used for the removal of ground. Our method considers point clouds located in the radar range, captured by a stereo camera or a 3D LIDAR and segmented using radar detections as seeds for the *min-cut* segmentation algorithm [3]. Multiple objects can appear in the detection therefore (*min-cut*) is used to isolating the object detected by radar.

A commercial bi-mode MMW radar (76.5 GHz Delphi ESR), a LIDAR 3D (Velodyne HDL-32) and a stereo camera (Pointgrey BumblebeeXB3) are used to detect obstacles and their boundaries. The radar is provided with simultaneous long (174 m.) and mid (60 m.) range functionality. The radar azimuth angle is 90 deg in the mid range, and 20 deg in the long range. The stereo camera adopted has a 60 deg Field of View (FOV) and its image resolution is 1280 X 960 pixels.

We use the CaRINA II platform [4] to collect data for evaluation of both calibration and obstacle detection methods.

II. RELATED WORK

Many approaches for obstacle detection based on the use of sound, images, LIDAR and radar sensors have been proposed. Others perform the task of obstacle detection using two or more sensors as 3D LIDAR, radar or cameras and merge features of these sensors to improve detections. Some approaches use two or more sensors to complement each other, e.g. radar with infrared [5], radar with camera [6] [7] [8] [9] [10] and radar with 3D LIDAR [11]. In [5] was used MMW radar to find hypotheses of cars in the 3D world, such hypotheses were projected in an image for its classification. The pose of the camera in relation to radar is estimated by manual measurement.

III. SENSOR CALIBRATION

This section describes the methodology followed to fuse three sensors (radar, 3D LIDAR and stereo camera) in the same coordinate system. First a general procedure was develop to align sensors. The camera coordinate system was taken as reference for the transformation of the other sensors

¹Luis Alberto Rosero is with Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, Av Trabalhador São-carlense, 400, Brazil lrosero@usp.br

²Fernando Santos Osório is with Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, Av Trabalhador São-carlense, 400, Brazil fosorio@usp.br

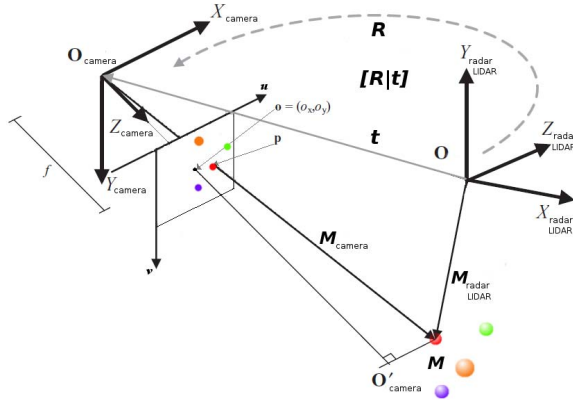


Fig. 1: Pin-hole camera model

coordinate systems. For this case we show the configuration specified for fusing camera with radar and 3D LIDAR respectively. In this way, all data (camera, radar and 3D LIDAR) are in a single coordinate system (camera coordinate system). The sensors were calibrated finding the camera pose related to the sensor (radar and 3D LIDAR). Our sensor calibration method is computer-vision-based and uses the pin-hole camera model implemented in the OpenCV library [12]. This camera model is based on [13] and represented by Eq. (1).

$$S \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}_K \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}_{[R|t]} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (1)$$

Fig. 1 shows the transformation performed by the $[R|t]$ matrix that maps points $((X_w, Y_w, Z_w) \in \mathbb{R}^3)$ coming from the w sensor (radar or 3D LIDAR) coordinate system to the 3D camera coordinate system. The intrinsic parameter matrix ($K \in \mathbb{R}^{3 \times 3}$) transforms points from the three-dimensional camera coordinate system to the final image coordinate system $((u, v) \in \mathbb{R}^2)$, where f_x, f_y are the focal lengths in pixel units and (o_x, o_y) is the principal point, usually located in the center of the image. To estimate the parameters of the K matrix we use the OpenCV library method for camera calibration using a chessboard.

The camera pose (position + orientation) is O_{camera} and the sensors pose (radar/LIDAR) is O . Through the $[R|t]$ matrix is performed a transformation of points related to the origin O of the radar/LIDAR for the frame of the camera defined by O_{camera} .

The camera pose estimation uses n 3D-to-2D correspondences. This is a well-known problem in computer vision and requires the estimation of the six degrees of freedom of the camera pose and camera parameters for its solution. A first approach to solve the problem is the use of a minimum of 6 pairs of correspondences between 3D and 2D world points and the Direct Linear Transform (DLT) algorithm [14]. However, the DLT finds a projection matrix P that projects

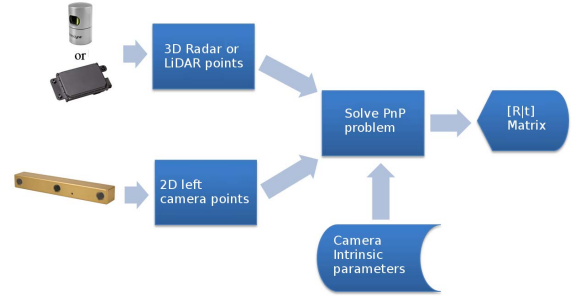
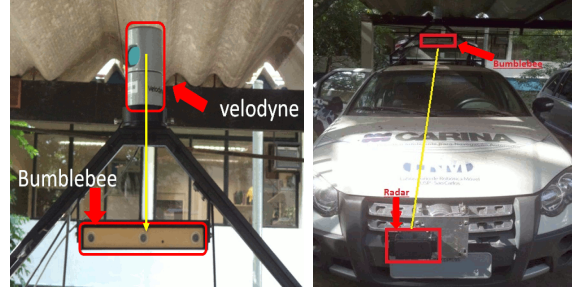


Fig. 2: Block diagram of the proposed calibration process



(a) Camera 3D LIDAR setup (b) Camera radar setup

Fig. 3: Setup for experimental test using CaRINA 2 platform

the 3D points directly on the image plane and requires intrinsic camera parameters.

A simplification of the problem is known as Perspective-n-Point (PnP). It assumes the intrinsic parameters of the camera are known and only need to calculate the $[R|t]$ matrix. The methods for solving the PnP problem are commonly used in augmented reality [15].

To solve the PnP problem and retrieve the camera pose, only three point correspondences may become necessary [16]. There are approaches that use iterative solutions to the problem with $n > 3$ points [17] [18] [19]. However, non-iterative solutions are more commonly used because of their lower computational complexity and high precision ($O(n^3)$ [20], $O(n^2)$ [21]). One of the most efficient methods is Efficient Perspective-n-Point camera pose estimation (EPnP) with complexity $O(n)$ [1].

Fig. 2 is a block diagram of the calibration process that allows to find the transformations for each sensor (radar and 3D LIDAR) individually to map points coming from radar or 3D LIDAR coordinate system to the camera coordinate system and image plane. This transformation is performed by the $([R|t])$ matrix. Points in the coordinate system of the sensors (radar, 3D LIDAR) and their correspondences in the image must be collected. For this task we use the CaRINA 2 platform. Fig. 3 shows the real pose and setup for the three sensors. The camera must be calibrated (intrinsic parameters) and a method that solves the PnP problem and finds the camera pose in relation to the sensors must be employed. The EPnP method was used to calculate the camera pose $([R|t])$ with relation to radar or 3D LIDAR sensors. The sensors were calibrated separately.



Fig. 4: Trihedral corner reflector



Fig. 5: Target segmentation in the image plane

A. Camera-radar calibration setup

A square trihedral corner reflector was constructed (Fig. 4) and placed over a traffic cone. Sequentially, M points in the radar coordinate system were collected. The reflector was constructed with three aluminum plates of $L = 30\text{ cm}$, 3 mm thickness and area of 30 cm^2 . The radar cross section (RCS) of the reflector is $19,873.6\text{ m}^2$. This value was calculated using Eq. (2), this equation is based on [22]. This reflector ensures a continuous target detection for one of the M 3D points in the radar coordinate system and serves as input for the method that solves the PnP problem.

$$\text{Triangular Trihedral RCS} = \frac{4\pi L^4}{\lambda^2} \quad (2)$$

To capture the 2D image points corresponding with the each point detected by the radar was performed an automatic method for target detection using color segmentation with threshold for the blue color in the HSV color space. The method segments the blue object and its edges; the blue area moment is calculated and yields the center of mass of the target (Fig. 5), which is the 2D point in the image corresponding to the 3D point detected by the radar (red dots). Fig. 5 shows the $n = 17$ points captured in the radar coordinate system and their correspondences in the image plane.

B. Camera-3D LIDAR calibration setup

For camera-3D LIDAR calibration a Styrofoam rectangle painted in blue (calibration pattern) is used as target with dimensions $100\text{ cm} \times 70\text{ cm}$. The target is placed in different positions in front of the camera. The four corners of the rectangle were used as 3D points for the calibration method.

The point cloud collected in a place with flat floor and a single obstacle in the front (target) was segmented for the detection of the rectangle. The floor was removed by the Difference of Normals (DoN) [2], which localizes regions

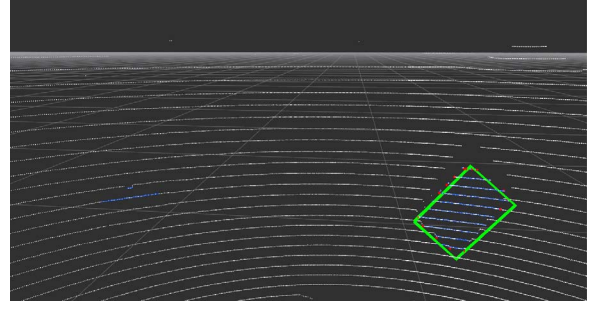


Fig. 6: Rectangle segmentation and corner detection on 3D point cloud

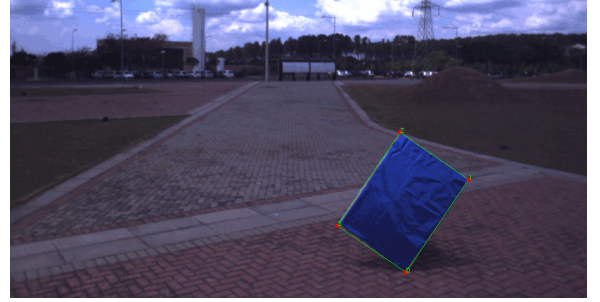


Fig. 7: Rectangle segmentation and corner detection on the image

of similarity in normal vectors at point clouds. A filter was then applied for the removal of outliers from the resulting target segmentation and the points of the 3D LIDAR that hit the rectangle were obtained.

To find the four 3D coordinates of the corners, we find the rectangle that best fits the segmented point cloud in the previous step. To do this, we first adjust the point cloud to a plane $ax + by + cz + d = 0$. After we find the convex hull (set of points that is necessary to form a polygon of minimum area that includes all the points of the plane) of the cloud of points of the target. The Rotating Calipers algorithm [23] is used with the convex hull points as input [24]. It fits the minimum rectangle that contains all the points of the set. Fig. 6 shows an example of the detection of the rectangle and its corners by DoN filter, removing outliers, setting the plane, convex hull and applying Rotating Calipers.

Similar to what was done to find the target in the image for radar-camera calibration, here we also use a target detector based on the blue color segmentation and its edges. Having the blue area segmented from the board the Rotating Calipers 2D algorithm is used to find the rectangle in the image. The four corners of the rectangle belong to the set of n 2D points used as the input of the method for solving the PnP problem. Fig. 7 shows an example of corner and rectangle detection in the image captured by the camera and Fig. 8 displays the corner detection for 19 targets placed sequentially, for a total of $n = 76$ 3D points in the LIDAR coordinate system (8b) and their correspondences in the plane of the 2D image (8a).

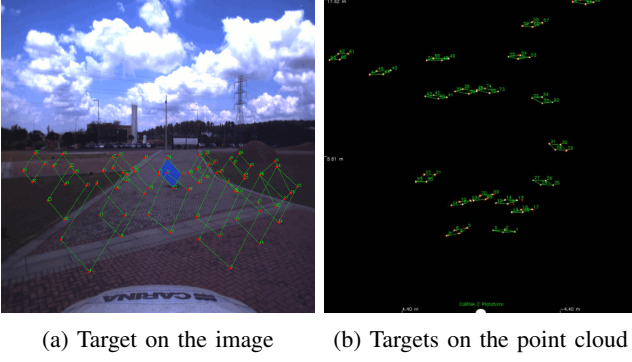


Fig. 8: Automatic points detection for 3D LIDAR and camera calibration

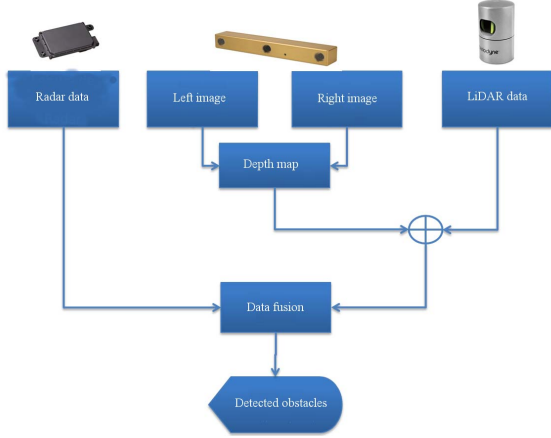


Fig. 9: Block diagram for the sensor fusion system proposed

IV. OBSTACLE DETECTION

Fig 9 shows a diagram of our sensor fusion proposal. Data from the three sensors (radar, LIDAR and stereo camera) were captured. Subsequently, the data captured by the radar served as a hypotheses for the obstacle detection performed in the fusion stage. One of the point clouds is chosen, i.e., either the one generated by the 3D LIDAR, or the one generated by the stereo camera. Each radar detection is used as a seed for the formation of a cluster around each obstacle and segments the point cloud.

A. Point cloud segmentation

To reduce the amount of data to be processed, we cut the point cloud around of the camera coordinate system. The point cloud was limited to 2 meters in the positive Y axis, which is slightly above the height of the car (CaRINA 2 platform), 0.5 meters in the negative Y axis, for the elimination of outliers below of the ground. Points more than 40 m. away in the Z axis were deleted. After the point cloud size reduction, a down sampling process is applied using a 3D voxel grid. The voxel size used is 0.1 m.

The *min-cut* algorithm [3] implemented in the PCL libraries [25] was used for point cloud segmentation. It performs a binary segmentation of the input cloud. In this case, having radar points of the object detection as center and

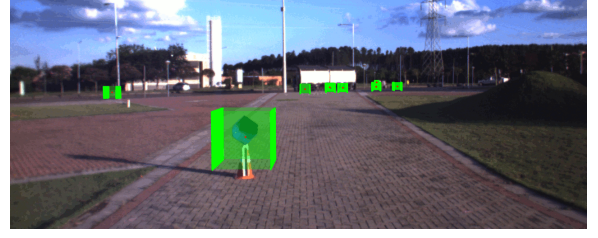


Fig. 10: Results of radar and camera calibration

setting a radius, the algorithm divides the cloud on two sets: foreground and background points. The *Min-cut-based* segmentation uses radar points as centers of the foreground. $\sigma = 0.1m$. is the spacing between points in the point cloud after down sampling. *Min-cut* was used only in areas where the radar detects obstacles. The *min-cut* PCL implementation is used and adapted for the avoidance of an extra processing of the original point cloud. The segmentation works as follows:

- 1) For each of the 64 points detected by radar we extract one square point cloud ($6m. \times 6m.$) taking as center every radar detection.
- 2) Bearing the 64 new point cloud, we build a graph for each one containing every single point of the extracted cloud and a set of vertices and two more vertices called source and sink. Each edge except source and sink are connected with their nearest neighbors.
- 3) The algorithm assigns three types of weights for each edge at each point cloud:

$$smoothCost = e^{-(dist/\sigma)^2} \quad (3)$$

$$backgroundPenalty = (distanceToCenter/radius) \quad (4)$$

$$distanceToCenter = \sqrt{(x - centerX)^2 + (y - centerY)^2} \quad (5)$$

- 4) According to the weights assigned in the previous step for the edges, the points connected to them are then differentiated.

V. RESULTS

Figs. 10 and 11 show results for radar-camera and 3D-LIDAR calibration respectively. For both the re-projection of the points generated by sensors (radar and 3D LIDAR) fit whit the objects in the image plane (corner reflector and Styrofoam rectangle).

Data coming from the radar, 3D LIDAR and stereo camera in real and controlled scenarios were captured by the ROS framework [26] for the evaluation of the obstacle detection method proposed. Data were processed off-line. The ground truth was created manually on the 3D LIDAR point cloud in order for comparing results of the detections obtained by the three sensors (radar, 3D LIDAR and stereo camera) separated and fusing two pairs of sensors.

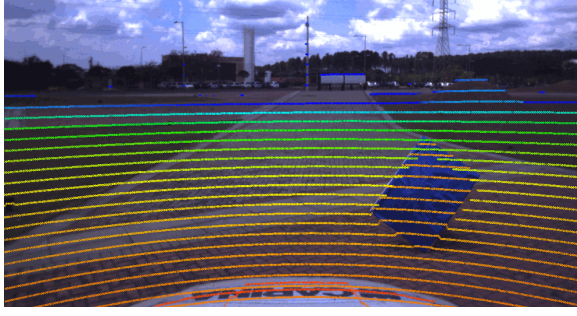


Fig. 11: Results of 3D LIDAR and camera calibration

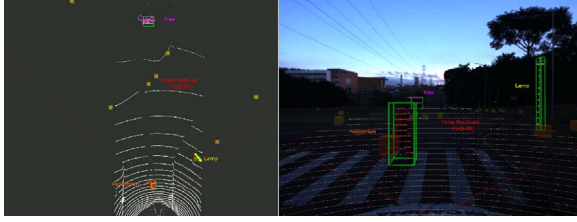


Fig. 12: Final results for on-road obstacle detection fusing radar and 3D point cloud data

For our dataset, 50 frames were saved with the information from the three sensors. Throughout the dataset, 325 objects were marked in an area of $30m. \times 5m$. (30m. is the distance limit to where we can have a good estimate of the disparity for obstacle detection using stereo camera, and 5 meters is the distance to each side of the road). Fig. 12 shows examples of obstacle detection using radar + 3D sensor fusion. The objects were automatically segmented and detected and the hit/error was verified against the ground truth.

TABLE I: Results (100 points)

Sensors	T.P.	F.P.	F.N.
radar	320	113	5
3D LIDAR	315	22	10
stereo camera	282	150	43
radar-3D LIDAR	308	10	17
radar-stereo camera	271	61	54
Stixel World	252	29	73

TABLE II: Results (Obstacle height = 50 cm.)

Sensors	T.P.	F.P.	F.N.
radar	320	113	5
3D LIDAR	315	22	10
stereo camera	282	150	43
radar-3D LIDAR	320	39	5
radar-stereo camera	320	68	5
Stixel World	252	29	73

Table I shows results where 100 points are the minimum number of points for a detection to be considered like obstacle. Therefore, very small obstacles (composed of 100 points or less) are disregarded. The approach generates a larger number of false negatives (objects not found) and causes some objects to be erroneously discarded for radar-

TABLE III: Results (20 points)

Sensors	T.P.	F.P.	F.N.
radar	320	113	5
3D LIDAR	315	22	10
stereo camera	282	150	43
radar-3D LIDAR	320	51	5
radar-stereo camera	320	78	5
Stixel World	252	29	73

stereo camera fusion. Moreover, a large number of false positives was estimated due to the detection of points near the ground and road edges and should not be considered.

According to Table II, a detection is considered an obstacle from 0 cm to 50 cm of height, this threshold allowed to increase the number of true positives (hits). In this case, the classification error is reduced. Finally, Table III shows obstacles with 20 points that form detections of smaller obstacles; however, the number of false positives was slightly increased in the radar-camera detection. The best result in terms of true positives was obtained by the radar only, and fusing with other sensors. The same also occurs as number of false negatives in the approaches based on the use of radar in the Tables II and III.

According to the results 98.46% of the obstacles present in the scene were detected by radar, but at a high false positive rate. Using only 3D LIDAR, 96.92% of the obstacles were detected and only stereo camera 86.76%.

The number of false positives detected for each sensor individually is high, especially in the case of the radar and the stereo camera. This is due to the presence of street edges (borders) sometimes detected by the sensors as obstacles. In radar data, false positives also appear in a large number in the middle of the street. Such false positives may have been caused by a wrong radar position, relatively low, and with an inclination that may also be unsuitable for a correct sensor operation. A better positioning of the sensor is being considered in order to obtain better radar data set.

The results for radar and 3D sensor fusion show reductions of 91.15% in the false positives for radar-3D LIDAR and 46.02% for radar-stereo camera fusion. Both percentages are calculated in relation to the individual use of the radar. This result is relevant and shows the importance of working with more than one sensor for improvement in the accuracy of obstacle detection. There are also a 54.54% reduction of false positives for the radar-3D LIDAR fusion compared with individual use of 3D LIDAR. For radar-stereo camera fusion we have a reduction of 59.33% of false positives compared to the use of stereo camera only.

To facilitate the visualization and comparison of the data of the tables above, two more tables are showed (Table IV and Table V). This tables aggregate only the results of the approaches using radar-stereo camera (Table IV)) and radar-3D LIDAR (Table V).

VI. CONCLUSIONS

This paper has presented a simple and effective obstacle detection approach that uses radar, 3D LIDAR and stereo

TABLE IV: Comparison between approaches for stereo camera obstacle detection

Sensors	T.P.	F.P.	F.N.
radar	320	113	5
stereo camera	282	150	43
radar-stereo camera (100 pts)	271	61	54
radar-stereo camera (50 cm)	320	68	5
radar-stereo camera (20 pts)	320	78	5

TABLE V: Comparison between approaches for 3D LIDAR obstacle detection

Sensors	T.P.	F.P.	F.N.
radar	320	113	5
3D LIDAR	315	22	10
radar-3D LIDAR (100 pts)	308	10	17
radar-3D LIDAR (50 cm)	320	39	5
radar-3D LIDAR (20 pts)	320	51	5

camera sensors to improve and complement the characteristics of the sensors for obstacle detection in the context of autonomous vehicles. The system takes advantages of radar precision to detect obstacles in all types of scenarios and environmental conditions and combines them with the 3D LIDAR and stereo camera detections, sensors that are less accurate but give the notion of the shape of the detected object.

ACKNOWLEDGMENT

The authors acknowledge the financial support provided by CAPES (Coordenadoria de Aperfeiçoamento de Pessoal de Nível Superior)

REFERENCES

- [1] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, pp. 155–166, 2009.
- [2] Y. Ioannou, B. Taati, R. Harapp, and M. Greenspan, "Difference of normals as a multi-scale operator in unorganized point clouds," *IEEE Computer Society*, 2012, pp. 501–508.
- [3] A. Golovinskiy and T. Funkhouser, "Min-cut based segmentation of point clouds," in *Computer Vision Workshops (ICCV Workshops)*, 2009 *IEEE 12th International Conference on*. IEEE, 2009, pp. 39–46.
- [4] L. C. Fernandes, J. R. Souza, G. Pessin, P. Y. Shinzato, D. Sales, C. Mendes, M. Prado, R. Klaser, A. C. Magalhães, A. Hata *et al.*, "Carina intelligent robotic car: Architectural design and applications," *Journal of Systems Architecture*, vol. 60, no. 4, pp. 372–392, 2014.
- [5] F. A. R. Alencar, L. A. Rosero, C. M. Filho, F. S. Osório, and D. F. Wolf, "Fast metric tracking by detection system: Radar blob and camera fusion," in *2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*, Oct 2015, pp. 120–125.
- [6] T. Kato, Y. Ninomiya, and I. Masaki, "An obstacle detection method by fusion of radar and motion stereo," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 182–188, 2002.
- [7] M. Bertozzi, L. Bombini, P. Cerri, P. Medici, P. Antonello, and M. Miglietta, "Obstacle detection and classification fusing radar and vision," in *Intelligent Vehicles Symposium*, 2008 *IEEE*, June 2008, pp. 608–613.
- [8] Y. Benezeth, B. Emile, H. Laurent, and C. Rosenberger, "Vision-based system for human detection and tracking in indoor environment," *International Journal of Social Robotics*, vol. 2, no. 1, pp. 41–52, 2010.
- [9] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 7, pp. 1239–1258, 2010.
- [10] X. Wang, L. Xu, H. Sun, J. Xin, and N. Zheng, "Bionic vision inspired on-road obstacle detection and tracking using radar and visual information," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, Oct 2014, pp. 39–44.
- [11] P. Fritsche, S. Kueppers, G. Briese, and B. Wagner, "Radar and lidar sensorfusion in low visibility environments," in *ICINCO - 13th International Conference on Informatics in Control, Automation and Robotics*, July 2016, pp. 1–8.
- [12] Itseez. (2016, November) Opencv. opencv.org/. [Online]. Available: <http://opencv.org/>
- [13] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [14] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
- [15] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: a hands-on survey," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 12, pp. 2633–2651, 2016.
- [16] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 930–943, Aug. 2003.
- [17] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *International journal of computer vision*, vol. 15, no. 1-2, pp. 123–141, 1995.
- [18] R. Horaud, F. Dornaika, and B. Lamiroy, "Object pose: The link between weak perspective, paraperspective, and full perspective," *International Journal of Computer Vision*, vol. 22, no. 2, pp. 173–189, 1997.
- [19] C.-P. Lu, G. D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 610–622, 2000.
- [20] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 578–589, 2003.
- [21] P. D. Fiore, "Efficient linear solution of exterior orientation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 140–148, 2001.
- [22] N. A. W. C. NAWC, "Electronic warfare and radar systems engineering handbook," Tech. rept. TP 8347. ewhdbks. mugu. navy. mil, Tech. Rep., 1997.
- [23] David. (2016, November) Computing oriented minimum bounding boxes in 2d. [Online]. Available: <https://geidav.wordpress.com/2014/01/23/computing-oriented-minimum-bounding-boxes-in-2d>
- [24] H. Freeman and R. Shapira, "Determining the minimum-area enclosing rectangle for an arbitrary closed curve," *Communications of the ACM*, vol. 18, no. 7, pp. 409–413, 1975.
- [25] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.
- [26] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.