

# Real-time Ground Filtration Method for a Loader Crane Environment Monitoring System Using Sparse LIDAR Data

Karol Miądlicki, Mirosław Pajor, Mateusz Saków

Faculty of Mechanical Engineering and Mechatronics

West Pomeranian University of Technology

Szczecin, Poland

karol.miadlicki@zut.edu.pl, miroslaw.pajor@zut.edu.pl, mateusz.sakow@zut.edu.pl

**Abstract**— The segmentation and analysis of the environment using three-dimensional (3D) data (point clouds) is a dynamically developing area. This article presents a real-time ground filtration method for the loader crane environment monitoring system. The falling prices of depth sensors based on light detection and ranging (LIDAR), time of flight (ToF), radio detection and ranging (RADAR) technologies, and growth-computing power led us to use the Velodyne VLP-16 sensor in the developed system. In the presented filtering solution, we use characteristic scan pattern properties, the dot product of vectors, and interpolation using the RANSAC method. Algorithm performance was evaluated based on real data acquired under different conditions, and the results were compared to known filtration methods. The described algorithm is developed for real-time operation; therefore, the computation time is critical. Furthermore, in this article, we discuss other methods used to extract ground points from the entire point cloud in real time, describe Velodyne VLP-16 scanner and data acquisition methods.

**Keywords**—ground segmentation; ground filtration; loader crane; LIDAR; Velodyne; point-cloud; real-time processing

## I. INTRODUCTION

Due to the increasing availability, decreasing prices, and development of virtual techniques, three-dimensional (3D) scanners are used in an increasing number of applications. Scanners based on laser or light beam return time can be divided into two groups: dense data scanners (high resolution) and sparse data scanners (low resolution). Accurate scanners from, e.g., Riegl and Teledyne Optech generate a highly dense grid of dots and need much time to perform a full scan. This long scan time makes them unsuitable for real-time applications. Accurate scanners are used for mapping and site inspection [1], creating 3D maps of cities [2, 3], or in forestry [4, 5]. In these applications, a 3D scan is first performed and then the point cloud is analyzed. The second group is scanners performing 20–30 scans per second. This group includes Microsoft Kinect v2, Leap Motion, SICK LMS, Ibeo LIDAR, and Velodyne LIDAR, among others. Despite the lower resolution of the generated point cloud, the scanning time allows them to be used in real-time applications. Therefore, the scanners of the second group are commonly used in the design of control systems: autonomous cars [6], robots [7], drones [8], submarines [9], and wheelchairs [10]. The control systems of these devices are

suitable to analyze the environment [11, 12]; recognize pedestrians [13, 14], objects, and obstacles [15]; or even find the right paths [16]. In all these applications, proper segmentation and classification based on sensor data are crucial. In the segmentation process, typically, the first step is the filtration of ground points. The result of this process has a direct impact on classification results.

In this paper, an innovative use for the real-time ground points filtration algorithm, developed for low-resolution 3D data obtained from a light detecting and ranging (LIDAR) sensor, has been presented. The presented algorithm is part of a sensor-fusion system for locating the operator and recognizing his or her gestures that are applied to control the loader crane. It is an innovative concept, because research on intelligent loader cranes using extended and virtual reality in their control systems has been conducted by few research centers. Currently, the loader cranes are commonly controlled by hydraulic levers or control panels. Joysticks, electronic levers, buttons, switches, light-emitting diodes (LEDs), and light crystal displays (LCDs) have also become standard. However, following the trends of dynamically developing industries, together with the Cargotec company, we decided to develop an innovative control and monitoring system for the loader crane.

The sensor-fusion environmental monitoring system for the loader crane will consist of an operator tracker with two red, green, and blue (RGB) cameras and near-infrared (NIR) cameras; a thermal imager; active LED markers; attitude and heading reference system (AHRS) and Global Positioning System (GPS) sensors; and a Velodyne LIDAR VLP-16 scanner. The algorithm proposed in this work is part of a subsystem that uses the LIDAR scanner to monitor the working area of the loader crane. The purpose of this subsystem is to segment the loader crane working area, monitor movements in the area, track the operator's position, and allow for mapping of the crane environment in virtual reality. The applied algorithm analyzes 16 points from each VLP-16 sensor scan. Points are projected onto the YZ plane. Then, after receiving a full 360 scan, the ground plane is fitted by the random sampling consensus (RANSAC) algorithm.

The rest of paper is organized as follows. In the next paragraph, the previous work on the ground points filtration problem will be discussed. Then, the Velodyne VLP-16 scanner

and the methods of data reading and interpretation are described. In sections IV and V, the proposed ground filtration algorithm and conducted experiments are described. Finally, a conclusion of this paper and an outline of future work are provided in section VI. To facilitate reading the paper, an abbreviation table (Table 1) is provided.

TABLE I. ABBREVIATIONS USED IN PAPER

Abbreviation	Expand Abbreviation
LIDAR	Light Detection and Ranging (depth sensor type)
ToF	Time of Flight (depth sensor type)
RADAR	Radio Detection and Ranging (depth sensor type)
LED	Light-emitting Diode (lighting diode)
LCD	Liquid Crystal Display (display type)
RGB	Red, Green, Blue (color model)
NIR	Near Infrared (wavelengths between 700-1200 nm)
AHRS	Attitude and Heading Reference System (MEMS sensor)
GPS	Global Positioning System (positioning sensor)
RANSAC	Random Sample Consensus (algorithm)
KNN	K-Nearest Neighbors (algorithm)
FH	Felzenszwalb and Huttenlocher (segmentation algorithm)
IMU	Inertial Measurement Unit (MEMS sensor)
VRU	Vertical Reference Unit (MEMS sensor)
GP-INSAC	Gaussian Process - Increment Sample Consensus (algorithm)
PROSAC	Progressive Sample Consensus (algorithm)
WPF	Windows Presentation Foundation (rendering user interface system)
UNITY	3D rendering engine

## II. RELATED WORK

### A. Intelligent Systems for Loader Cranes

A detailed overview of the control systems used in loader cranes is presented in [17]. Only a few academic centers are working on the development and testing of control interfaces and support systems for loader cranes. In [18, 19], the researchers presented a novel approach to controlling the crane. The authors outlined the main assumptions of the intelligent system for the loader crane and focused on the development of the voice interface. Their system allows the operator to use voice commands and natural language to control the crane. On the other hand, in [20], the authors proposed a method of controlling the loading crane using gestures and a Microsoft Kinect sensor. Another approach was presented by Westerberg et al. in [21, 22]. In the presented solutions, they used virtual reality and teleoperation. In addition, crane manufacturers are constantly improving their control systems. In 2016, Hiab presented the HiVision control system for forestry cranes. It allows the operator to monitor the environment around the crane without leaving the cab. Cameras with a 270° angle of view mounted on the crane arm and Oculus Rift (virtual reality goggles) were used to implement this system. A review of the available literature reveals a substantial lack of research on the crane environment monitoring system or tracking the crane operator's position using LIDAR or other vision systems.

### B. Ground Filtration Methods

Ground points filtration is the first step in the segmentation of the entire point cloud. The result of filtration has a direct effect on segmentation and results classification. Several methods for separating ground points with dense and sparse point clouds were developed and applied. The algorithms developed for dense point clouds are most often based on dividing them into a voxel grid. Subsequently, K-nearest neighbors (KNN), graphs, Felzenszwalb and Huttenlocher (FH), and other algorithms described in [23-26] are used for segmentation and classification. These methods are highly accurate, but due to computational complexity, they are not suitable for real-time applications. The second group of algorithms developed for sparse point clouds is ideal for solutions operating in real time. These methods can be divided into several groups depending on the used feature: points coordinate and sensor position, dividing scan into segments and points projection, spherical coordinates, or graph theory.

Height-based methods are used when the position and orientation of the sensor are known relative to the ground or wheels. These methods are used in autonomous vehicles in which the sensor is mounted on the roof of the car [27, 28], on the car's bumper [29], or in mobile robots [7]. Information about the sensor position allows for an estimate of the ground plane to be obtained by analyzing the points coordinates (Z) at the scanning stage [30]. The effectiveness of this method increases the use of inertial measurement unit (IMU), vertical reference unit (VRU), or AHRS. Sensors make the algorithm immune to all LIDAR sensor movements. Unexpected changes to the position or orientation of the sensor could cause errors in the estimation of the ground plan. Improvements to the height-based method by adding the angle threshold were proposed by Asvadi [31].

An approach based on a segmentation LIDAR scan is used in many algorithms. These algorithms work particularly well with Velodyne devices, because the scans performed by these devices after projecting to the horizontal plane have a characteristic scan pattern (Fig. 1). This pattern facilitates the allocation of a scan on a radial grid. The number of circles shown in Fig. 1 depends on the number of laser sensors in the scanner.

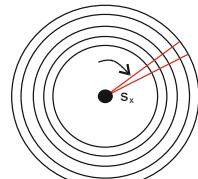


Fig. 1 A typical Velodyne scan pattern shown in a horizontal plane. Only five laser lines are shown

Himmelsbach et al. [32] proposed to divide the point cloud into voxels, as in dense cloud methods. However, the processing time of his method was about 100 ms. Consequently, a year later, he and his team presented an algorithm [31] in which they used the characteristic Velodyne sensor pattern. In [33], they partition the data in a way that allows ground plane points to be estimated by a simple comparison to local line fits. Efficient partitioning is achieved by representing the XY-plane by a circle of infinite radius and dividing it into a discrete number of

segments. Then, each segment is split into bins. To each bin is assigned the Z coordinate value depending on the points assigned that bin. In the end, on the bins, a two-dimensional (2D) line extraction is performed to indicate the approximate ground level. The operation is repeated for all segments. The processing time for this algorithm is only about 75 ms. A similar approach is proposed in Tongtong et al.'s work [34]. However, after receiving 2D bin points, the Gaussian process-increment sample consensus (GP-INSAC) was used for interpolation. Another method based on a single read analysis was proposed in [6]. In this paper, the authors analyze the angle between two vectors, AB and BC, created from three consecutive points. The value of the angle should be about zero if points A, B, and C lie on the ground. The scalar product of the vectors is used to determine the angle. For the points lying on the ground, the value tends to be one. In the end, the threshold based on the scalar product value allows for the filtration of ground points. On the other hand, Hu [35] proposed an approach using a moving window classifying points. Another approach was proposed by Hu [35]. His algorithm classifies ground points on a sliding window basis. An iterative method to estimate ground points with local constraints was proposed by Hwang et al. [36]. They divide all 3D points into several groups according to the distance from the sensor and estimate local planes from close plane to far plane sequentially. Postica with his team [37] used Markov Random Fields to remove ground points from the scan, while the team from the Institute of Automation in Beijing used RANSAC [38] or progressive sample consensus (PROSAC) [39] for ground plane fitting.

The method developed by Yin et al. [40] is also based on an analysis of the characteristic pattern for rotary LIDAR sensors. However, the point filtration algorithm is based on the breakpoints and turning points of the radial distance curve in the spherical coordinates without assuming the road is located in the lowest part of scan. In addition, in the algorithm, they use three consecutive scans: the previous, the current, and the next. If the radial distance of the adjacent points is less than the specified threshold, they are classified as ground plane. The segmentation time for this algorithm is approximately 2.8 ms.

The graph theory in the segmentation algorithm was used by Moosmann et al. [41]. The assumption of the authors was to use all the information provided by the 3D scan for fast segmentation. The algorithm proposed by them consists of a few steps: performing a scan, performing a neighborhood graph, calculating coefficients, and segmenting. The execution time of the algorithm is 352 ms, but the authors are planning to optimize or parallelize the calculations.

### III. VLP-16 LIDAR SCANNER

Laser distance or LIDAR sensors measure distance by illuminating a target with a laser and analyzing the reflected light. Due to the decreasing price and improving measurement parameters, they have gained more and more attention. LIDAR sensors are used in many applications: urban planning, urban mapping, intelligent autonomous transport, and scanning forests and agricultural fields. Used in the described system, the VLP-16 scanner is the third generation of laser scanners produced by Velodyne. A scanner is composed of 16 individual laser-detector pairs that are individually aimed in two-degree increments on

the laser head. That sensor configuration provides a 30° field of view (-15° to 15°), which results in 300,000 depth points per second. Based on research conducted at the University of Houston [42], where researchers have evaluated the sensor for long-term stability, geometric calibration, and the effect of temperature variations, we can conclude that the information about sensor accuracy for the factory calibration is correct. Moreover, after the local calibration, it is even possible to improve the sensor accuracy by +/- 10 mm. The VLP-16 sensor only provides information about azimuth (readings from an encoder) and distance (depth). To read the coordinates of the points (x, y, z) in space, it is necessary to convert the received data while taking into account the position of the laser sensor pairs and the current head position (encoder measurement). Coordinates are calculated based on the simplified equation described in [42]. The simplification is allowed because the first six parameters of the equation are internal sensor calibration constants. These constants are defined by the manufacturer during the production calibration process of the device at the production stage. Only the azimuth (encoder reading) and the point distance from the sensor are obtained from the sensor. The simplified equations for each coordinate and its geometric representation are shown in Fig. 3.

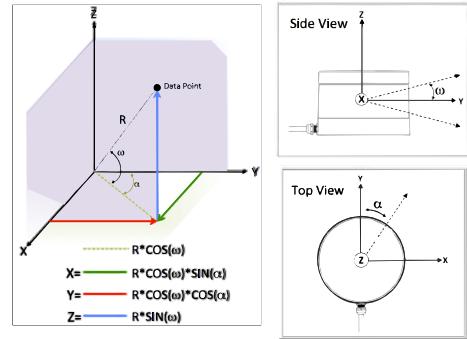


Fig. 3 Raw sensor data to XYZ conversion

Fig. 4 shows the point cloud obtained after converting raw data from the sensor. Sparse lines visible on the image are caused by only 16 laser sensors in the VLP-16. Despite the low resolution in Fig. 4, three human bodies, a loader crane, room walls, and other environmental elements (hydraulic pump, test stand) are visible.

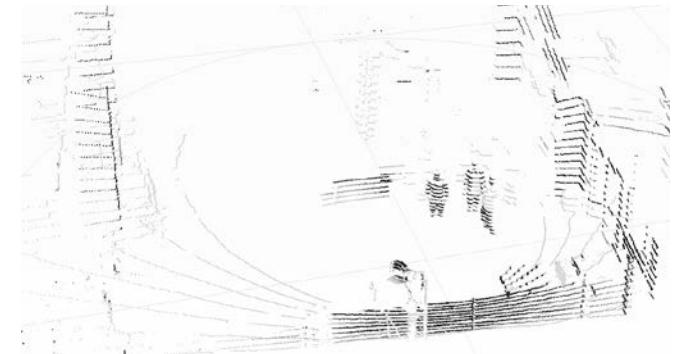


Fig. 4 LIDAR scan of the crane test stand

#### IV. GROUND FILTRATION ALGORITHM

The algorithm we propose is similar to the solutions presented in the works [6, 32, 34]. However, in our solution we use scan segmentation, dot product of vectors, and interpolation using the RANSAC method. In the presented algorithm, we analyze a single scan from 16 laser sensors (Fig. 5B). Then, the points are projected onto the YZ plane (Fig. 5C). Starting with the point having the smallest Z coordinate value, we create two vectors from three successive points. If the angle between the vectors DE and EF is zero, the points are probably lying on the ground plane. To determine this condition, we use the dot product of the DE and EF vectors, whose value should be close to 1. In the next step, we select points D, F, and G. The operation is repeated until all points have been analyzed. Because each point is involved in computing the scalar product of the vectors, we stored all three values in the array three times. This avoids misclassifying a point if it is near an obstacle or it there is a slightly uneven terrain, because the threshold is based on two maximum values from the dot product array. In addition, the use of this matrix allows for the extraction of the image edges, i.e., the place where obstacles or walls begin. Additionally, to avoid the detection of walls as a ground plane, the thresholds for differences in the X coordinate are observed. The last step in the algorithm is to determine the ground plane. For this purpose, we used the RANSAC method. The linear function is fitted to the filtered ground points projected on the YZ plane (Fig. 5D). The segmentation algorithm is summarized in Algorithm 1.

---

**Algorithm 1:** Ground points filtration

---

```

Input      : IN_points = [x,y,z;....;x16,y16,z16]
Output    : [ground_points, ground_model]
Parameters : threshold_val, X_tresh, num_iter,
               threshDist, inlierRatio
1:   for i = 0:14
2:     vecDE = vector([Y1,Z1],[Yi,Zi])
3:     vecEF = vector([Yi,Zi],[Yi+1,Zi+1])
4:     cos_th = dot(vecDE vecEF)
5:     cos_mem = [cos_thi-2,cos_thi-1,cos_th]
6:     if (sum(cos_mem) > threshold_val) &&
        Xi-Xi+1 > X_tresh)
7:       ground_points = [ground_points,[Xi, Yi, Zi]]
8:     Else
9:       break
10:    end
11:   end
12:   if full_360_scan == true
13:     ground_model = RANSAC(ground_points,
           num_iter, threshDist, inlierRatio)
14:   End
```

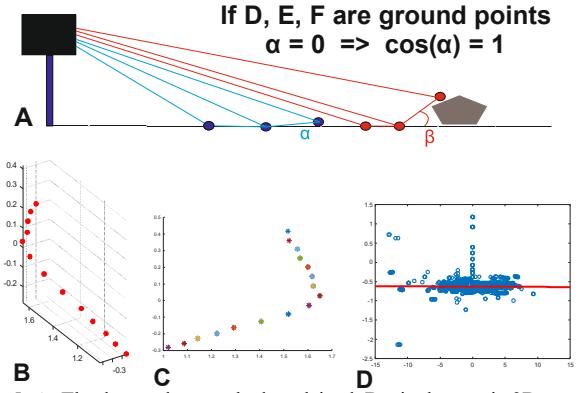


Fig. 5. A: The dot product method explained, B: single scan in 3D, C: single scan in 2D, D: RANSAC interpolation

#### V. EXPERIMENT

During the experiment, three ground point filtration algorithms were tested. The first algorithm filtered the ground points only based on the Z coordinate, the second algorithm was based on the work presented in [7], and the third algorithm is described in this paper. The principle of operation of the evaluated algorithms is shown in Fig. 6A–C. The tests were carried out at the test stand of the loader crane using three different positions of the VLP-16 scanner. The position of the scanner during the tests is shown in Fig. 6D. The segmentation thresholds for methods 2 and 3 were selected based on experiments. A comparison of the ground plane filtration results using different methods is shown in Fig. 7.

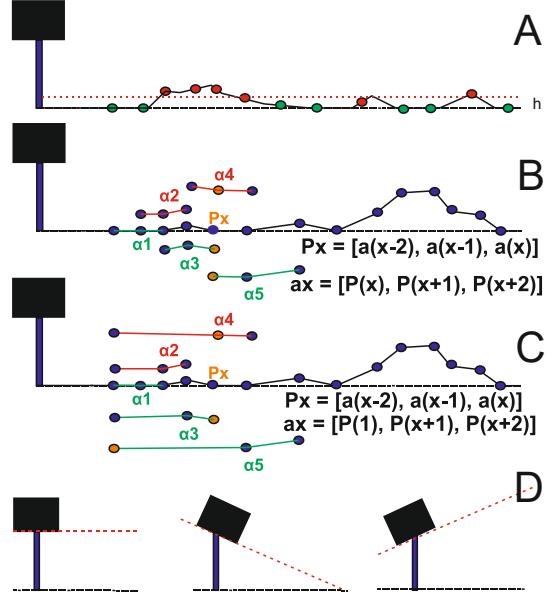


Fig. 6. A–C: Evaluated filtration algorithms, D: LDAR scanner test positions

The left column (Fig. 7A–C) shows a straight segmentation based on the height (Z coordinate). The ground filtration threshold was set at -0.44 because the scanner was placed on a tripod at a height of 44 cm. As shown in Fig. 7, the filtration for this method functions correctly only if the sensor is set parallel to the ground.

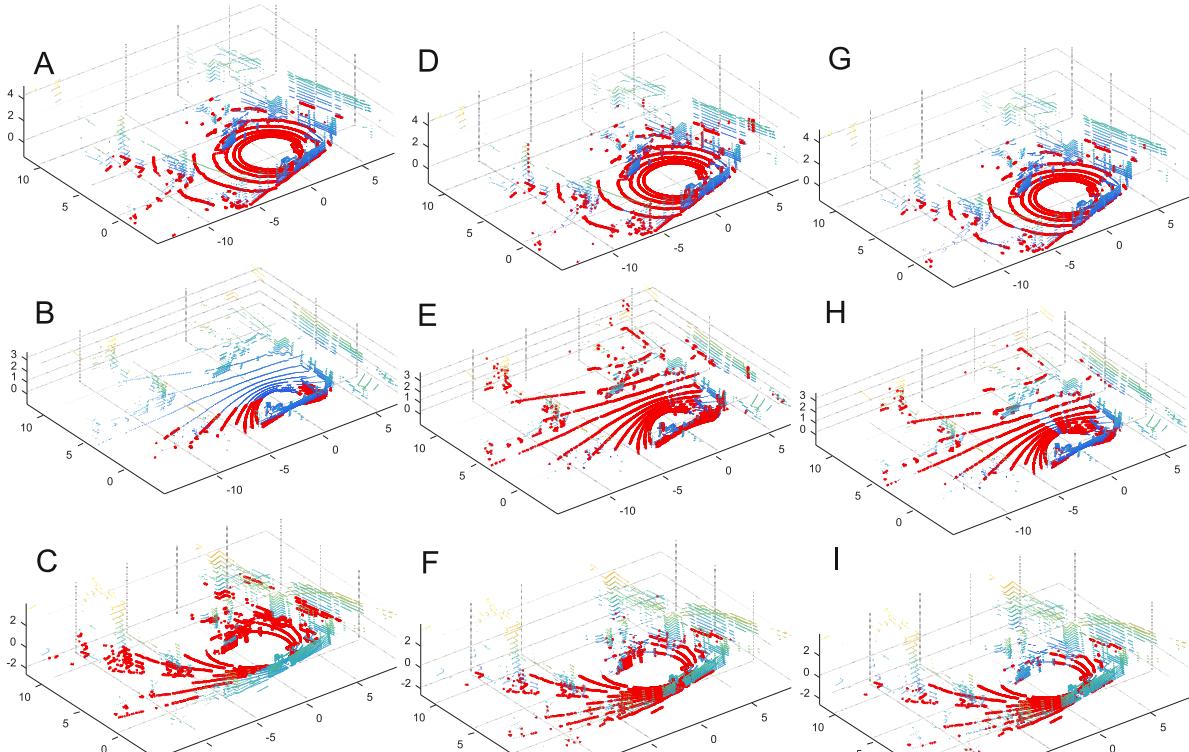


Fig. 7. Filtration results of the proposed method (right column) in three different VLP-16 scanner positions, compared to the results of modified methods in [7] (center column) and simple Z coordinate value threshold (left column)

If the scanner is tilted, any points of a given height are marked as land points. In the middle column (Fig. 7D–E), a memory array extended method, proposed in [6], is presented. In this case, filtration is much better regardless of the sensor position. However, the method has the tendency to mark too many points as the ground plane. Especially, the walls are often marked as the ground. This is because the points on the walls are arranged in the same way as on the ground. Wall points lie on the line but on a vertical plane, as shown in Fig. 5C. The right column (Fig. 7G–H) shows the results obtained using the proposed filtration algorithm. In this case, the segmentation results regardless of the position of the sensor are acceptable. The false classification of the points forming the walls was also limited by thresholding by X coordinate. A few erroneously filtered points do not corrupt the model's RANSAC matching. This is confirmed by fitting, as shown in Fig. 5D. However, during the tests, we noticed a few minor drawbacks. Points will be misclassified if the first point of the single scan does not lie on the ground or the scan will contain noise points. As well, the threshold value in the X direction depends on the range of space being scanned. Too small a threshold for a large space causes no points to be classified as ground; on the other hand, too high a value in a small range does not affect the wall points.

## VI. CONCLUSIONS AND FUTURE WORK

In this work, we presented a ground points filtering algorithm for a sensor fusion system designed for the localization and gesture recognition of a loader crane operator. The presented algorithm allows the ground points to be filtered accurately from the point cloud in real time and allows

for the generation of a ground plane model. Experiments show that a combination of the dot product of vectors and interpolation using the RANSAC algorithm provides better results than using a single method. The results of filtration using our algorithm and computation time are promising. The calculation time for the entire scan is about 20 ms. However, time and accuracy depend on the size of the memory matrix and the parameters of the RANSAC algorithm. Currently, the optimization of the code in C# and the Windows Presentation Foundation (WPF) library for charts visualization is being performed. It is expected to increase the efficiency of the algorithm by about 30 times. As part of further work, the implementation of the wall separation algorithm, the complete segmentation of the load crane environment, and an operator-tracking algorithm are planned. In addition, there is a plan to use the UNITY engine to transfer the load crane work zone to virtual reality. The algorithm will be implemented in the loader crane control system.

## ACKNOWLEDGMENTS

The work was carried out as part of the PBS3/A6/28/2015 project, "The use of augmented reality, interactive voice systems and operator interface to control a crane," financed by the NCBiR.

## REFERENCES

- [1] M. Vlaminck, L. Hiep Quang, W. Goeman, P. Veelaert, and W. Philips, "Towards online mobile mapping using inhomogeneous lidar data," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 845–850.
- [2] F. Yu, J. Xiao, and T. Funkhouser, "Semantic alignment of LiDAR data at city scale," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1722–1731.
- [3] L. Li, X. Zhao, L. Yi, and Y. Li, "Extracting urban ground object information from images and LiDAR data," in *2015 8th International Congress on Image and Signal Processing (CISP)*, 2015, pp. 754–759.

- [4] J. Tian, L. Wang, X. Li, C. Shi, and H. Gong, "Differentiating Tree and Shrub LAI in a Mixed Forest With ICESat/GLAS Spaceborne LiDAR," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 1, pp. 87-94, 2017.
- [5] D. Marinelli, C. Paris, and L. Bruzzone, "Fusion of high and very high density LiDAR data for 3D forest change detection," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016, pp. 3595-3598.
- [6] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Auton. Robots*, vol. 26, no. 2-3, pp. 123-139, 2009.
- [7] T. A. Lima, M. D. d. N. Forte, F. G. Nogueira, B. C. Torrico, and A. R. d. Paula, "Trajectory tracking control of a mobile robot using lidar sensor for position and orientation estimation," in *2016 12th IEEE International Conference on Industry Applications (INDUSCON)*, 2016, pp. 1-6.
- [8] M. U. d. Haag, C. G. Bartone, and M. S. Braasch, "Flight-test evaluation of small form-factor LiDAR and radar sensors for sUAS detect-and-avoid applications," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, pp. 1-11.
- [9] D. McLeod, J. Jacobson, M. Hardy, and C. Embry, "Autonomous inspection using an underwater 3D LiDAR," in *2013 OCEANS - San Diego*, 2013, pp. 1-8.
- [10] H. Grewal, A. Matthews, R. Tea, and K. George, "LiDAR-based autonomous wheelchair," in *2017 IEEE Sensors Applications Symposium (SAS)*, 2017, pp. 1-6.
- [11] R. Oprimolla, G. Fasano, G. Rufino, M. Grassi, and A. Savvaris, "LiDAR-inertial integration for UAV localization and mapping in complex environments," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016, pp. 649-656.
- [12] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard, "Motion-based detection and tracking in 3D LiDAR scans," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4508-4513.
- [13] C. Benedek, B. Nagy, B. Gálai, and Z. Jankó, "Lidar-based gait analysis in people tracking and 4D visualization," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 1138-1142.
- [14] J. Wang, W. Tao, and Z. Zhongyang, "LiDAR and vision based pedestrian detection and tracking system," in *2015 IEEE International Conference on Progress in Informatics and Computing (PIC)*, 2015, pp. 118-122.
- [15] E. Shang, X. An, J. Li, and H. He, "A novel setup method of 3D LiDAR for negative obstacle detection in field environment," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 1436-1441.
- [16] J. K. Kim *et al.*, "Experimental studies of autonomous driving of a vehicle on the road using LiDAR and DGPS," in *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, 2015, pp. 1366-1369.
- [17] K. Miądlicki and M. Pajor, "Overview of user interfaces used in load lifting devices," *International Journal of Scientific & Engineering Research*, vol. 6, no. 9, pp. 1215-1220, 2016.
- [18] M. Majewski and W. Kacalak, "Human-Machine Speech-Based Interfaces with Augmented Reality and Interactive Systems for Controlling Mobile Cranes," in *Interactive Collaborative Robotics: First International Conference, ICR 2016, Budapest, Hungary, August 24-26, 2016, Proceedings*, A. Ronzhin, G. Rigoll, and R. Meshcheryakov, Eds. Cham: Springer International Publishing, 2016, pp. 89-98.
- [19] M. Majewski and W. Kacalak, "Conceptual Design of Innovative Speech Interfaces with Augmented Reality and Interactive Systems for Controlling Loader Cranes," in *Artificial Intelligence Perspectives in Intelligent Systems, Vol 1*, vol. 464, R. Silhavy, R. Senkerik, Z. K. Oplatkova, P. Silhavy, and Z. Prokopova, Eds. (Advances in Intelligent Systems and Computing, Berlin: Springer-Verlag Berlin, 2016, pp. 237-247.
- [20] K. Pietrusiewicz and K. Miądlicki, "Gestures can control cranes," *Control Engineering*, vol. 61, no. 2, p. 14, 2014.
- [21] S. Westerberg, I. R. Manchester, U. Mettin, P. L. Hera, and A. Shiriaev, "Virtual environment teleoperation of a hydraulic forestry crane," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 4049-4054.
- [22] S. Westerberg and A. Shiriaev, "Virtual environment-based teleoperation of forestry machines: Designing future interaction methods," *Journal of Human-Robot Interaction*, vol. 2, no. 3, pp. 84-110, 2013.
- [23] B. Douillard *et al.*, "On the segmentation of 3D LiDAR point clouds," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 2798-2805.
- [24] A. Nguyen and B. Le, "3D point cloud segmentation: A survey," in *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 2013, pp. 225-230.
- [25] G. Vosselman, B. G. Gorte, G. Sithole, and T. Rabbani, "Recognising structure in laser scanner point clouds," *International archives of photogrammetry, remote sensing and spatial information sciences*, vol. 46, no. 8, pp. 33-38, 2004.
- [26] A. Ajazi, P. Checchin, and L. Trassoudaine, "Segmentation Based Classification of 3D Urban Point Clouds: A Super-Voxel Based Approach with Evaluation," *Remote Sensing*, vol. 5, no. 4, p. 1624, 2013.
- [27] S. Cho *et al.*, "Sloped Terrain Segmentation for Autonomous Drive Using Sparse 3D Point Cloud," *The Scientific World Journal*, vol. 2014, p. 9, 2014, Art. no. 582753.
- [28] Y. Zhang, J. Wang, X. Wang, C. Li, and L. Wang, "A real-time curb detection and tracking method for UGVs by using a 3D-LIDAR sensor," in *2015 IEEE Conference on Control Applications (CCA)*, 2015, pp. 1020-1025.
- [29] K. Stiens, G. Tanzmeister, and D. Wollherr, "Local elevation mapping based on low mounted lidar sensors with narrow vertical field of view," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 616-621.
- [30] I. Bogoslavskyi and C. Stachniss, "Fast range image-based segmentation of sparse 3D laser scans for online operation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 163-169.
- [31] A. Asvadi, P. Girão, P. Peixoto, and U. Nunes, "3D object tracking using RGB and LiDAR data," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1255-1260.
- [32] M. Himmelsbach, T. Luettel, and H. J. Wuensche, "Real-time object classification in 3D point clouds using point feature histograms," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 994-1000.
- [33] M. Himmelsbach, F. v. Hundelshausen, and H. J. Wuensche, "Fast segmentation of 3D point clouds for ground vehicles," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, 2010, pp. 560-565.
- [34] T. Chen, B. Dai, D. Liu, B. Zhang, and Q. Liu, "3D LiDAR-based ground segmentation," in *The First Asian Conference on Pattern Recognition*, 2011, pp. 446-450.
- [35] X. Hu, X. Li, and Y. Zhang, "Fast Filtering of LiDAR Point Cloud in Urban Areas Based on Scan Line Segmentation and GPU Acceleration," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 2, pp. 308-312, 2013.
- [36] S. Hwang, N. Kim, Y. Choi, S. Lee, and I. S. Kweon, "Fast multiple objects detection and tracking fusing color camera and 3D LiDAR for intelligent vehicles," in *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2016, pp. 234-239.
- [37] G. Postica, A. Romanoni, and M. Matteucci, "Robust moving objects detection in lidar data exploiting visual cues," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1093-1098.
- [38] X. Meng, Z. Cao, L. Zhang, S. Wang, and C. Zhou, "A slope detection method based on 3D LiDAR suitable for quadruped robots," in *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, 2016, pp. 1398-1402.
- [39] X. Meng, C. Zhou, Z. Cao, L. Zhang, X. Liu, and S. Wang, "A slope location and orientation estimation method based on 3D LiDAR suitable for quadruped robots," in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2016, pp. 197-201.
- [40] H. Yin, X. Yang, and C. He, "Spherical Coordinates Based Methods of Ground Extraction and Objects Segmentation Using 3-D LiDAR Sensor," *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 1, pp. 61-68, 2016.
- [41] 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*, 2009, pp. 215-220.
- [42] C. L. Glennie, A. Kusari, and A. Facchini, "Calibration and stability analysis of the vlp-16 laser scanner," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XL-3/W4, pp. 55-60, 2016.