LiDAR data augmentation by interpolation on spherical range image

1st Edison Velasco-Sánchez
AUtomatics, RObotics, and Artificial Vision Lab
University of Alicante, Spain.
edison.velasco@ua.es
0000-0003-2837-2001

2nd Ignacio de Loyola Páez-Ubieta
AUROVA Lab
University Miguel Hernández and University of Alicante
Elche and Alicante, Spain
ipaez@umh.es and ignacio.paez@ua.es
0000-0001-9901-7264

3rd Francisco A. Candelas
AUROVA Lab
University of Alicante
Alicante, Spain
francisco.candelas@ua.es
0000-0002-7126-0374

4th Santiago T. Puente
AUROVA Lab
University of Alicante
Alicante, Spain
santiago.puente@ua.es
0000-0002-6175-600X

2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA) | 979-8-3503-3991-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ETFA54631.2023.10275512

***Abstract*—LiDAR sensors are used for mapping tasks, LiDAR odometry or 3D environment reconstruction. Several of them count with a high number of vertical layers, which increase their price and prevents research groups from carrying out experiments and scientific advances. In this paper, we propose a method for augmenting point cloud data by bilinear interpolation in a Spherical Range Image. Our method improves others on the state-of-the-art by means of standard deviation filtering of the newly generated layers. The system operates at a frequency greater than 10 Hz for data interpolation up to 20 times. In addition, we present two applications for our approach such as LiDAR odometry and LiDAR-Camera fusion, obtaining better results than others that do not apply data augmentation. Finally we make available to the scientific community a package development on ROS (Robot Operating System). The code is available at https://github.com/EPVelasco/lidar-camera-fusion**

***Index Terms*—LiDAR, sensor fusion, LiDAR odometry, bilinear interpolation.**

I. Introduction and related works

Light Detection And Ranging (LiDAR) technology has been used in recent years in the field of robotics for autonomous driving. Its characteristics of generating point clouds using optical waves have the potential to achieve better accuracy in detecting at indoor or outdoor environments [1], [2]. One of the mostly used sensors of this kind in the autonomous robotics are those having a 360-degree Field Of View (FOV) [3]–[5]. They are marketed depending on the resolution and number of layers on the vertical axis. Generally, a higher number of layers represent a higher price. That is why several works that include them are performed with less-layer sensors [6]–[9].

Research work was funded by the Valencian Regional Government through the PROMETEO/2021/075 project and the Spanish Government through the Formación del Personal Investigador [Research Staff Formation (FPI)] under Grant PRE2019-088069.

Shan et al. have proposed a method that increases the number of layers of a LiDAR sensor [10]. On it, the authors increase the resolution of the LiDAR by projecting a point cloud onto a range image and increasing the resolution of that image using a deep neural network. Then, this image is converted back to a point cloud, which is denser than the original one. In this way, the virtual augmentation generated data can convert the one with a few layers into one with more. A method that does not use neural networks is [11], where the authors use a 4-channel Spherical Range Image (SRI) (x, y, z and intensity) of the point cloud. They interpolate the data in the empty space by calculating both the distance between pixels and the range values among the six nearest neighbours points to preserve the original object shapes during the reconstruction process.

In this work, we use a Velodyne VLP-16 rotating LiDAR sensor that has 16 layers to augment its number of layers using bilinear interpolation. By converting the point cloud to an SRI, we perform data augmentation with image plane interpolation. In addition, to reduce the noise generated during the process, we apply a point cloud filtering on each layer. Paying attention to the standard deviation between layers generated from the image depth data, we determine if the points generated virtually could be part of the real scenario. The experiments performed on point cloud interpolation in the image plane show satisfactorily that the data augmentation can help to improve other robotic areas that use this type of sensors such as LiDAR-Camera fusion, LiDAR odometry or mapping.

This work in progress is organized as follows: section II explains how we implemented our interpolation algorithm, converting the point cloud to image and augmenting it with a bilinear interpolation in order to finish with its filtering in the image plane, sec. III shows different experiments run with the

Authorized licensed use limited to: UNIVERSIDAD DE ALICANTE . Downloaded on October 03,2024 at 10:52:01 UTC from IEEE Xplore. Restrictions apply.

images obtained by applying the proposed algorithm and sec. IV summarises the paper and comments some future works.

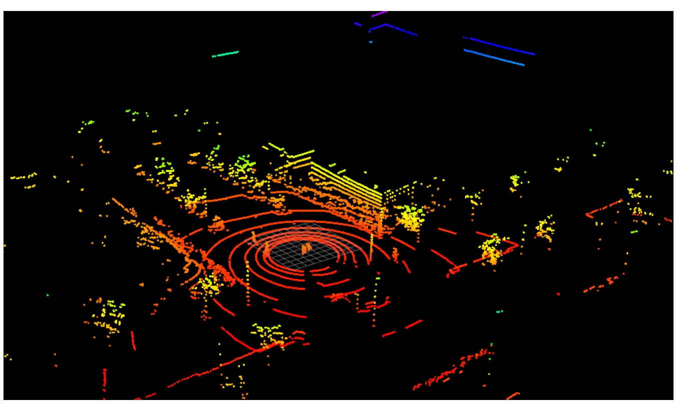
II. Methodology

The proposed method consists of the following steps. First, we convert the input point cloud into a SRI and we apply bilinear interpolation on the plane image. Then, we filter the generated image by using the standard deviation between each generated layer. Finally, we convert that interpolated and filtered SRI back to a point cloud that will be denser than the original input. Finally, in the experiments sec. III, we propose 3 experiments with our method for augmenting data from the aforementioned LiDAR sensor. The first one is the point cloud interpolated on an RGB image. The next one is LiDAR-Camera fusion with interpolated point clouds, where each point in the point cloud has a color channel. The last one is on LiDAR odometry with a point cloud interpolated with our method.

A. LiDAR to spherical range imaging

The interpolation algorithm requires a projection from the input point cloud onto an SRI. To convert this point cloud into a SRI, each layer of the LiDAR is transformed into a row of the SRI, and each column is the horizontal viewing angle of the LiDAR. In our case, the FOV is 360 degrees. Once the row and column coordinates of each image element are obtained, the value of each pixel is the depth of each point obtained from the LiDAR. In this way, we convert an $\mathbb{R}^3$ point cloud into an $\mathbb{R}^2$ range image as shown in the Fig. 1.

(a) Point cloud to range image



(b) Original Point cloud

Fig. 1: Conversion from point cloud (a) to SRI (b) with a resolution of 16x720. The number of rows corresponds to the number of laser layers and the number of columns to the 0.5 degrees resolution for the FOV of the used LiDAR sensor.

B. Data augmentation with bilinear interpolation

Using the range image obtained from the transformation of the point cloud, a bilinear interpolation is performed to virtually increase the number of LiDAR layers. To do this, the original image data is saved in an array and we use the bilinear interpolation method ($interp2$) of the armadillo library [12]. Fig. 2 shows the linearly interpolated image 5, 10 and 20 times. In this way, we can linearly increase the number of sensor layers by interpolating them. In order to easily reconstruct the new interpolated point cloud, we generate an array with the z-components, which is interpolated with the same method as the SRI.

(a) Without Interpolation

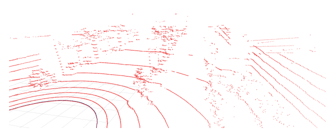
(b) With 5-layer interpolation

(c) With 10-layer interpolation

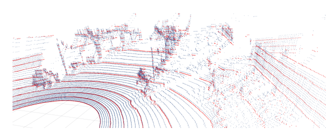
(d) With 20-layer interpolation

Fig. 2: SRI linearly interpolated 5, 10 and 20 times.

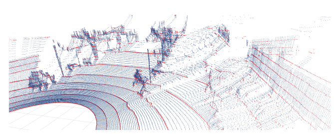
Subsequently, having interpolated and filtered the SRI $\boldsymbol{I}$ and its $\boldsymbol{z}$ components, we calculate the new points of the generated virtual layers. For this, the x-components are calculated as $\boldsymbol{x} = \sqrt{\boldsymbol{I} - \boldsymbol{z}} \cdot \cos(\boldsymbol{\omega})$ and the y-components as $\boldsymbol{y} = \sqrt{\boldsymbol{I} - \boldsymbol{z}} \cdot \sin(\boldsymbol{\omega})$. In both components, $\boldsymbol{\omega}$ is determined by the column value transformed into a value between π and $-\pi$. Fig. 3 shows the reconstruction of the interpolated images in a new point cloud.



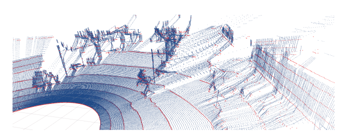
(a) Without Interpolation



(b) With 5-layer interpolation



(c) With 10-layer interpolation



(d) With 20-layer interpolation

Fig. 3: Point cloud reconstruction with SRI on different interpolation values. The red points represent the original point cloud and the blue ones the new generated points.

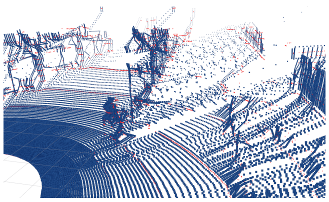
C. Filtering of the interpolated point cloud

When data interpolation is performed on the image plane to virtually create LiDAR layers, it generates points that could not be part of the real environment. As shown in Fig. 3c and Fig. 3d, this series of points are given when performing linear

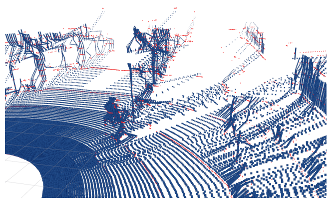
Authorized licensed use limited to: UNIVERSIDAD DE ALICANTE . Downloaded on October 03,2024 at 10:52:01 UTC from IEEE Xplore. Restrictions apply.

interpolation. Some of them are far away in the reality but together on the image plane. Our interpolation method works well for vertical objects such as walls, poles or trees, but has the problem of generating a trail of points when interpolating layers between objects that are not distant from each other. Therefore, we have implemented a filter based on the standard deviation σ between the interpolated points, determined as $\sigma^2 = \frac{1}{N}\sum_0^n (x_n - u)^2$. Where u and N are the average and number of elements analyzed, respectively.

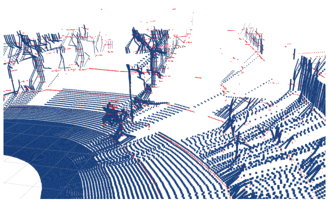
In this way, when examining the interpolated data between each layer, the data with a higher standard deviation value than the calculated one is filtered out. Fig. 4 shows the filtering of data with standard deviation limit values of 1, 10, and 100 for a point cloud interpolated 20 times.



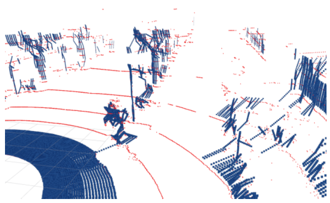
(a) Point cloud without layer filtering.



(b) Layer filtering with. $\sigma = 100$



(c) Layer filtering with. $\sigma = 10$



(d) Layer filtering with. $\sigma = 1$

Fig. 4: Standard deviation filtering analysis of interpolated layers.

III. Experiments

In this section, we detail the experiments performed with a non-interpolated and an interpolated to different values point cloud.

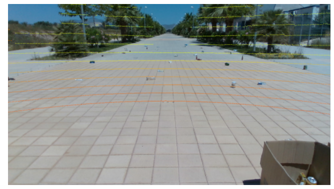
A. Point cloud on Image

For this set of experiments, we have calibrated the camera and the LiDAR sensor with the method shown in [13]. This calibration method returns the homogeneous transformation matrix between the LiDAR and an RGB camera. We used a Intel® RealSense™ D435i and a Velodyne VLP-16 sensor.

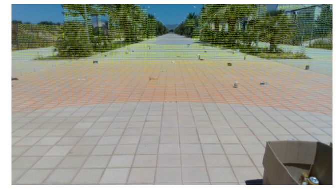
In Fig. 5, we show a point cloud projected onto a RGB image. It can be seen that the interpolated point cloud allows to generate virtually a large number of points to the environment. This experiment has been tested in paper [14], where the depth of domestic waste in outdoor environments is estimated.

B. LiDAR-Camera Fusion

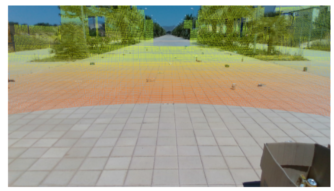
Having the homogeneous transformation matrix between the LiDAR-Camera, the interpolated point cloud can be reconstructed with a color channel at each point. Thus, Fig. 6 shows the LiDAR-Camera fusion on which the interpolated point cloud is colored. This method can be used for getting a denser environment mapping.



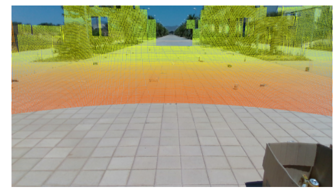
(a) Original point cloud projected on the image plane.



(b) Point cloud 5 times interpolated.

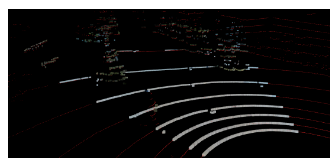


(c) Point cloud 10 times interpolated.

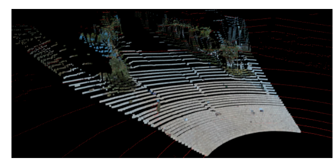


(d) Point cloud 20 times interpolated.

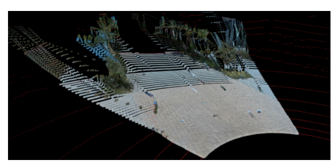
Fig. 5: Point cloud interpolated at different values and projected on the image plane.



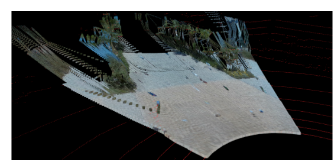
(a) Point cloud without interpolation.



(b) Point cloud 5 times interpolated.



(c) Point cloud 10 times interpolated.



(d) Point cloud 20 times interpolated.

Fig. 6: LiDAR-Camera fusion with interpolated point clouds at different values.

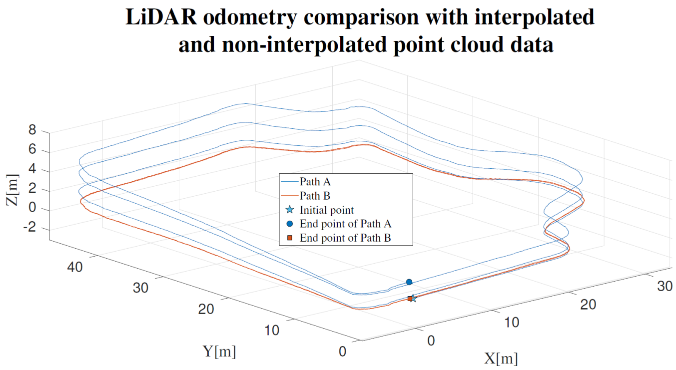
C. LiDAR Odometry with an interpolated point cloud

LiDAR odometry is a process used in robotics and autonomous navigation to estimate the position and orientation of a moving vehicle using LiDAR sensors [15]. In these experiments, we use the LiDAR odometry F-LOAM method [9] with a non-interpolated and an up to 10 times interpolated point cloud set. This set is from a 151 meters closed loop, which has been traversed 4 times.

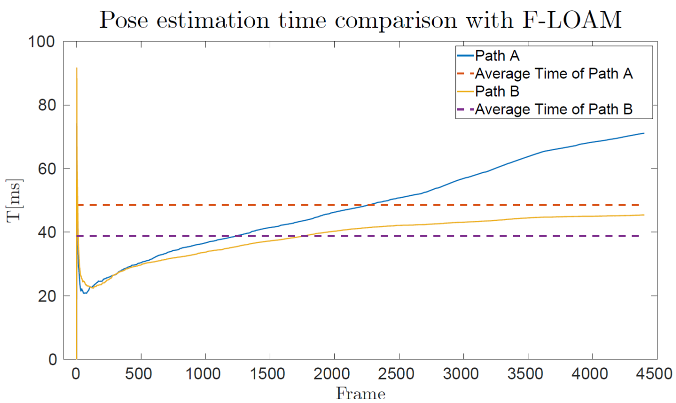
As shown in Fig. 7a, by interpolating the point cloud by 10 times, the pose and orientation estimation of the F-LOAM method generates almost a zero deviation in all 3 axes. In contrast, in the experiments without interpolating the point cloud, the error is accumulated in the z-axis as the robot moves forward in the circuit.

Authorized licensed use limited to: UNIVERSIDAD DE ALICANTE . Downloaded on October 03,2024 at 10:52:01 UTC from IEEE Xplore. Restrictions apply.

In addition, as shown in Fig. 7b, using the interpolated point cloud decreases the pose estimation time of the F-LOAM method. Despite having a larger number of points to process, the optimizer that estimates the position of the robot decreases its computation time.



(a) Comparison of LiDAR odometry with interpolated and non-interpolated point clouds.



(b) LiDAR odometry with a 10-layer interpolation

Fig. 7: Trajectory and pose estimation times generated by F-LOAM after 4 loops of the robot around the track. The results of path A are the experiments performed with the raw point cloud. The results of path B, on the other hand, are the experiments with the point cloud interpolated up to 10 times.

IV. Conclusion and Future works

The proposed method achieves the interpolation of a 16-layer LiDAR by converting the input point cloud onto a SRI by using bilinear interpolation. In addition, a standard deviation filter is applied so as to remove the noisy data generated during the interpolation. The generated point cloud significantly improved the results on LiDAR odometry and LiDAR-Camera fusion experiments. As future work, we are working on improving the data interpolation method using machine learning techniques and improving the filtering of virtual data with nearest neighbor groupings.

References

[1] T. Raj, F. Hanim Hashim, A. Baseri Huddin, M. F. Ibrahim, and A. Hussain, "A survey on lidar scanning mechanisms," *Electronics*, vol. 9, no. 5, p. 741, 2020. doi:https://doi.org/10.3390/electronics9050741

[2] N. Li, C. P. Ho, J. Xue, L. W. Lim, G. Chen, Y. H. Fu, and L. Y. T. Lee, "A progress review on solid-state lidar and nanophotonics-based lidar sensors," *Laser & Photonics Reviews*, vol. 16, no. 11, p. 2100511, 2022. doi:https://doi.org/10.1002/lpor.202100511

[3] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR)*, ser. IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012, pp. 3354–3361, iEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, JUN 16-21, 2012. doi:10.1109/CVPR.2012.6248074

[4] I. Maksymova, C. Steger, and N. Druml, "Review of lidar sensor data acquisition and compression for automotive applications," in *Proceedings*, vol. 2, no. 13. MDPI, 2018, p. 852. doi:https://doi.org/10.3390/proceedings2130852

[5] I. del Pino, M. A. Munoz-Banon, S. Cova-Rocamora, M. A. Contreras, F. A. Candelas, and F. Torres, "Deeper in blue," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 1, pp. 207–225, 2020. doi:https://doi.org/10.1007/s10846-019-00983-6

[6] G.-M. Lee, J.-H. Lee, and S.-Y. Park, "Calibration of vlp-16 lidar and multi-view cameras using a ball for 360 degree 3d color map acquisition," in *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2017, pp. 64–69. doi:10.1109/MFI.2017.8170408

[7] P. Zhou, X. Guo, X. Pei, and C. Chen, "T-loam: Truncated least squares lidar-only odometry and mapping in real time," *IEEE Transactions on Geoscience and Remote Sensing*, 2021. doi:10.1109/TGRS.2021.3083606

[8] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765. doi:10.1109/IROS.2018.8594299

[9] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-loam : Fast lidar odometry and mapping," in *2021 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS)*, ser. IEEE International Conference on Intelligent Robots and Systems. IEEE; RSJ, 2021, pp. 4390–4396, iEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), ELECTR NETWORK, SEP 27-OCT 01, 2021. doi:10.1109/IROS51168.2021.9636655

[10] T. Shan, J. Wang, F. Chen, P. Szenher, and B. Englot, "Simulation-based lidar super-resolution for ground vehicles," *Robotics and Autonomous Systems*, vol. 134, p. 103647, 2020. doi:https://doi.org/10.1016/j.robot.2020.103647

[11] J. You and Y.-K. Kim, "Up-sampling method for low-resolution lidar point cloud to enhance 3d object detection in an autonomous driving environment," *Sensors*, vol. 23, no. 1, p. 322, 2023. doi:10.1016/j.eswa.2022.118624

[12] C. Sanderson and R. Curtin, "Armadillo: a template-based c++ library for linear algebra," *Journal of Open Source Software*, vol. 1, no. 2, p. 26, 2016. doi:10.21105/joss.00026 1

[13] A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna, "LiDAR-Camera Calibration using 3D-3D Point correspondences," *ArXiv e-prints*, May 2017. doi:https://doi.org/10.48550/arXiv.1705.09785

[14] I. d. L. Páez-Ubieta, E. Velasco-Sánchez, S. T. Puente, and F. A. Candelas, "Detection and depth estimation for domestic waste in outdoor environments by sensors fusion," *arXiv preprint arXiv:2211.04085*, 2022. doi:https://doi.org/10.48550/arXiv.2211.04085

[15] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

Authorized licensed use limited to: UNIVERSIDAD DE ALICANTE . Downloaded on October 03,2024 at 10:52:01 UTC from IEEE Xplore. Restrictions apply.