$$K_C^W = \begin{bmatrix} R_C^W & T_C^W \\ 0\ \ 0\ \ 0 & 1 \end{bmatrix} \quad (6)$$

After applying the cross-calibration, we obtain the transformation between laser scanner and camera, transferring the points from the LIDAR coordinate system in the coordinate system of the camera.

$$K_L^C = \begin{bmatrix} R_L^C & T_L^C \\ 0\ \ 0\ \ 0 & 1 \end{bmatrix} \quad (7)$$

We can obtain the extrinsic calibration matrix of the LIDAR relative to the world coordinate system:

$$K_L^W = K_C^W \cdot K_L^C \quad (8)$$

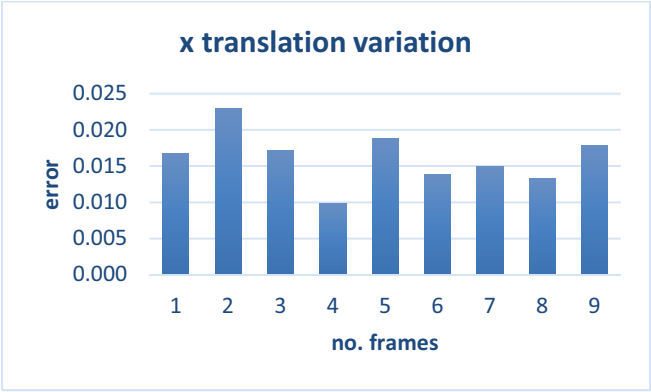
Therefore, if we know the extrinsic matrix of the camera relative to the world, and the extrinsic cross-calibration between the laser scanner and the camera, we can find the extrinsic calibration between the LIDAR and the world.

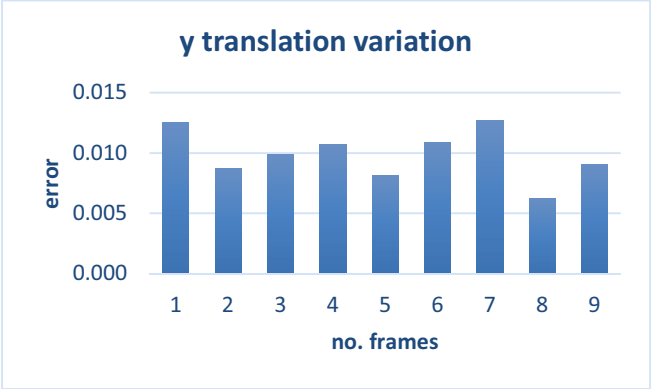
VII. Experimental Results

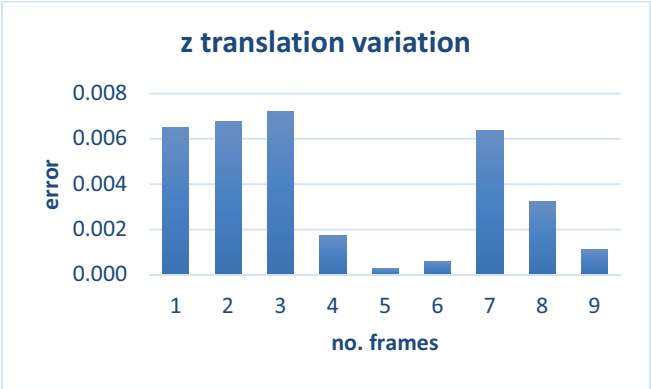
Our system is developed in C++, using the following libraries: PCL (Point Cloud Library), OpenCV, Boost, Qt, and Eigen. The hardware on which we ran the tests is Intel x64 processor, with the frequency of 4 GHz, and 64 GB RAM memory. The operating system is Windows 10.

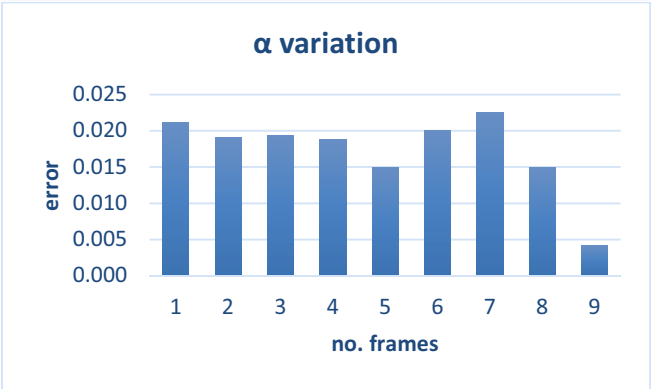
As sensors, we use a Velodyne VLP-16 sensor with 16 layers of data oriented between -15 and +15 degrees vertically, covering a vertical field of view of 30 degrees. The sensor covers a 360-degree horizontal FOV, while spinning at 10 Hz, and provides 300.000 points per second. It can cover a distance up to 100m, with an error of -/+3cm. The camera is JAI BM 141GE, that gives monochrome images. The coordinate systems of the world and the two sensors have the z-axis pointing forward, the x-axis – to the right and the y-axis – to the ground.

We performed experiments to determine the speed and accuracy of our algorithm. The algorithm running time is between 16ms for one frame and 90ms for 9 frames, the most time-consuming part being the grid search step. To track the miss-calibration correction, we have randomly altered the 6 extrinsic calibration parameters with noise, and compared the obtained values with the ground truth cross-calibration, which is known from the offline algorithm. Each translation parameter was modified with values between 1 and 2cm, and each rotation value was altered with up to 2 degrees, which are quite noticeable. Higher offsets would lead to having a low probability that the calibration is correct, which is computed in the miss-calibration step, therefore they would not be updated and the car should be stopped. The experiments were performed on a 1.900 frame log, in which we have variation in the speed of the car, pedestrians moving, and cars coming from the opposite direction. As metrics, we use the Mean Squared Error (MSE) between the values of the cross-calibration obtained from the offline method, which is considered the ground truth, and the values obtained from the online cross-calibration method.



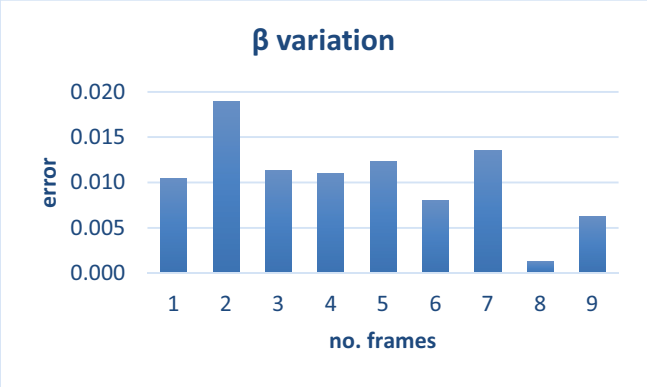


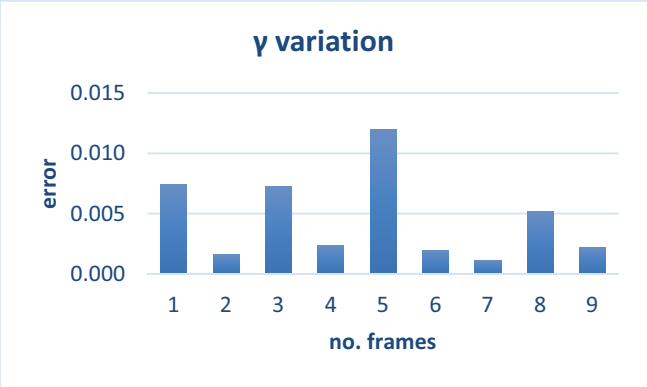




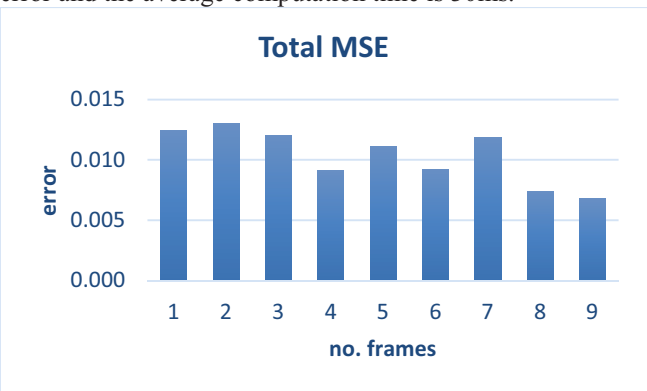
From the plots with the variations of the 3 translation parameters and 3 rotation parameters, we can observe the errors we obtain for each of the 6 cross-calibration parameters when we vary the number of frames. The biggest variations can be observed for the translation on the z-axis and for the γ angle, from which we can notice where we obtain high or low

errors, depending on the window size. This was used to decide on the optimal number of frames to be applied to the online cross-calibration algorithm, in such a way that the error between the ground truth and the refined result is minimized.





The error decreases when using more frames, the lowest error of 0.007 being obtained when having a window of 9 frames, as seen in the plot of MSE. The only downside is the computation time, which is quite high for such an application. Therefore, we choose to use 4 as the number of frames for the online cross-calibration algorithm, because it gives a low error and the average computation time is 30ms.



In Figure 7, the 3D points obtained from the laser scanner are projected onto the 2D image, with noise added to the cross-calibration between the camera and the LIDAR. After applying our online cross-calibration algorithm, we can successfully refine the calibration parameters and correct them; the result can be seen in Figure 8.

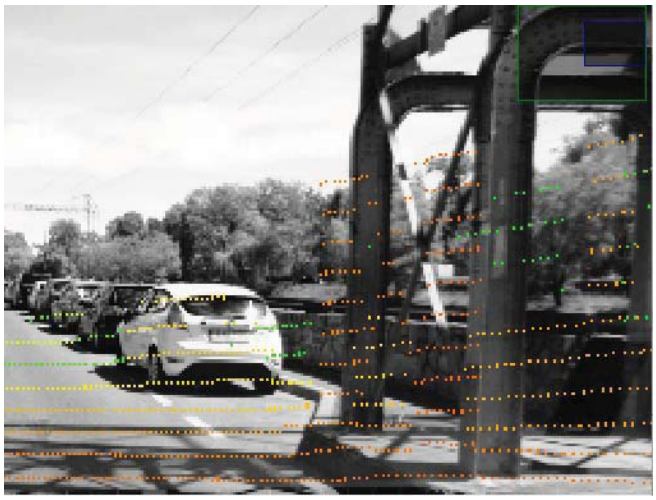


Figure 7. Miss-calibrated projection of LIDAR data on image

Figure 8. Corrected cross-calibration and projection of LIDAR data on image

In Table 1, from the results obtained in the miss-calibration tracking, it can be seen that our implementation has a higher degree of accuracy than other existing methods. We don't obtain an improvement when refining the translation on the y-axis, which is vertical because the laser scanner we use has only 16 layers; this gives a low density of points on the vertical edges, compared to the information obtained from edges in an image, which is dense.

Table 1. Online cross-calibration results

	Translation (m)			Rotation (degrees)		
Paper	x	y	z	α	β	γ
[23]	0.0045	0.0052	0.0046	0.38	0.39	0.44
[16]	0.02	0.014	0.006	0.672	0.628	0.476
[6]	0.305	-0.005	-0.426	-0.15	0	0.27
Our system	0.002	0.015	-0.005	-0.016	0.002	0.01

The cross-calibration algorithm can also be used to refine the results of an offline calibration algorithm, even when the sensors' position doesn't change, being able to correct small mistakes. In Figure 9, we have plotted the errors between the online and offline cross-calibration methods, when no noise has been added to the sensors' readings. The offline method has high errors for two rotation parameters, which are corrected with the online method.

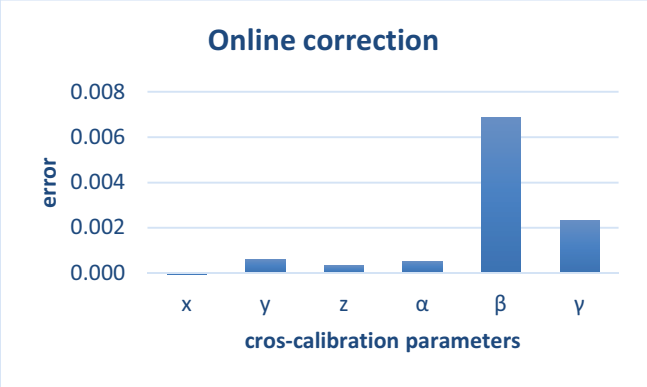


Figure 9. Offline cross-calibration correction

VIII. Conclusions

This work presents a new online solution for the cross-calibration of camera and LIDAR. Our system can detect miss-calibrations if sensor drift occurs during vehicle functioning. We are able to correct the cross-calibration between the camera and LIDAR, and also extract the extrinsic calibration parameters of LIDAR. We have obtained better results than previous existing cross-calibration methods, as seen in section VII because the most important improvement we bring is the usage of a motion correction algorithm, which eliminates distortions from the laser scanner that come from vehicle motion or objects moving in the scene. We also create a range image that is an organized representation of the 3D point cloud. By fusing the two sensors, camera, and LIDAR, we obtain a 2D image containing range information. The experimental results demonstrate that our solution is able to run in real time, which makes our online cross-calibration algorithm a convenient component of a driving assistance solution.

Acknowledgment

This work was supported by the MULTISPECT grant (Multispectral environment perception by fusion of 2D and 3D sensorial data from the visible and infrared spectrum) of the Romanian National Authority for Scientific Research and Innovation / UEFISCDI, project code PN-III-P4-ID-PCE-2016-0727, contract number 60/2017.

References

[1] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis, "3D LIDAR-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization," *Int. J. Rob. Res.,* vol. 31, pp. 452-467, 2012.

[2] C. Glennie and D. D. Lichti, "Static Calibration and Analysis of the Velodyne HDL-64E S2 for High Accuracy Mobile Scanning," *Remote Sensing,* vol. 2, p. 1610, 2010.

[3] L. Huang, "Lidar, camera and inertial sensors based navigation techniques for advanced intelligent transportation systems," University of California, Riverside, 2010.

[4] L. Huang and M. Barth, "A novel multi-planar LIDAR and computer vision calibration procedure using 2D patterns for automated navigation," in *2009 IEEE Intelligent Vehicles Symposium*, 2009, pp. 117-122.

[5] H. Alismail, L. D. Baker, and B. Browning, "Automatic Calibration of a Range Sensor and Camera System," in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, 2012, pp. 286-292.

[6] G. Pandey, J. McBride, S. Savarese, and R. Eustice, "Extrinsic Calibration of a 3D Laser Scanner and an Omnidirectional Camera," presented at the 7th IFAC Symposium on Intelligent Autonomous Vehicles, IAV 2010 - Proceedings, 2010.

[7] S. Debattisti, L. Mazzei, and M. Panciroli, "Automated extrinsic laser and camera inter-calibration using triangular targets," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 696-701.

[8] S. A. R. F, V. Fremont, and P. Bonnifait, "Extrinsic calibration between a multi-layer lidar and a camera," in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2008, pp. 214-219.

[9] M. Velas, M. Spanel, Z. Materna, and A. Herout, "Calibration of RGB camera With Velodyne LiDAR," *WSCG 2014 Communication Papers Proceedings,* pp. 135-144, 2014.

[10] J. Castorena, U. S. Kamilov, and P. T. Boufounos, "Autocalibration of lidar and optical cameras via edge alignment," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 2862-2866.

[11] S. Sim, J. Sock, and K. Kwak, "Indirect Correspondence-Based Robust Extrinsic Calibration of LiDAR and Camera," *Sensors,* vol. 16, p. 933, 2016.

[12] C. Weber, S. Hahmann, and H. Hagen, "Sharp Feature Detection in Point Clouds," presented at the Proceedings of the 2010 Shape Modeling International Conference, 2010.

[13] C. H. Rodríguez-Garavito, A. Ponz, F. García, D. Martín, A. d. l. Escalera, and J. M. Armingol, "Automatic laser and camera extrinsic calibration for data fusion using road plane," in *17th International Conference on Information Fusion (FUSION)*, 2014, pp. 1-6.

[14] G. Carrera, A. Angeli, and A. J. Davison, "SLAM-based automatic extrinsic calibration of a multi-camera rig," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2652-2659.

[15] M. Miksch, B. Yang, and K. Zimmermann, "Automatic extrinsic camera self-calibration based on homography and epipolar geometry," in *2010 IEEE Intelligent Vehicles Symposium*, 2010, pp. 832-839.

[16] J. Levinson and S. Thrun, "Automatic Online Calibration of Cameras and Lasers," *International Symposium on Experimental Robotics (ISER),* 2013.

[17] W. Maddern, A. Harrison, and P. Newman, "Lost in translation (and rotation): Rapid extrinsic calibration for 2D and 3D LIDARs," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 3096-3102.

[18] G. Atanacio-Jiménez, J.-J. González-Barbosa, J. B. Hurtado-Ramos, F. J. Ornelas-Rodríguez, H. Jiménez-Hernández, T. García-Ramirez, *et al.*, "LIDAR Velodyne HDL-64E Calibration Using Pattern Planes," *International Journal of Advanced Robotic Systems,* vol. 8, p. 59, 2011.

[19] J.-Y. Bouguet, "Camera Calibration Toolbox for Matlab," 2003.

[20] S. Hong, H. Ko, and J. Kim, "VICP: Velocity updating iterative closest point algorithm," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 1893-1898.

[21] S. Nedevschi. *Distance Transform (DT). Pattern Matching using DT.* Available: http://users.utcluj.ro/~nedevschi/PR/labs/prs_lab_04e.pdf

[22] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke, "Registration with the Point Cloud Library: A Modular Framework for Aligning in 3-D," *IEEE Robotics & Automation Magazine,* vol. 22, pp. 110-124, 2015.

[23] A. Napier, P. Corke, and P. Newman, "Cross-calibration of push-broom 2D LIDARs and cameras in natural scenes," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3679-3684.