SFM AND SEMANTIC INFORMATION BASED ONLINE TARGETLESS CAMERA-LIDAR SELF-CALIBRATION

Balázs Nagy[1,2], *Levente Kovács*[1], *Csaba Benedek*[1,2*]

[1]Institute for Computer Science and Control (MTA SZTAKI), Kende u. 13-17 Budapest, Hungary
[2]Pázmany Péter Catholic University, Faculty of Inf. Tech. and Bionics, Práter u. 50/A, Budapest

ABSTRACT

In this paper we propose an end-to-end, automatic, online camera-LIDAR calibration approach, for application in self driving vehicle navigation. The main idea is to connect the image domain and the 3D space by generating point clouds from camera data while driving, using a structure from motion (SfM) pipeline, and use it as the basis for registration. As a core step of the algorithm we introduce an object level alignment to transform the generated and captured point clouds into a common coordinate system. Finally, we calculate the correspondences between the 2D image domain and the 3D LIDAR point clouds, to produce the registration. We evaluated the method in various different real life traffic scenarios.

Index Terms— LIDAR, camera, calibration

1. INTRODUCTION

Autonomous driving systems [1], equipped with 3D LIDAR sensors and electro-optical cameras can achieve accurate and comprehensive environment perception. Accurate LIDAR and camera calibration is essential for robust data fusion, issues that are extensively studied in the literature. Existing calibration techniques can be grouped based on various aspects: the necessity of user interaction, specific environmental conditions, operational requirements, semi- [2] of fully automatic [3], target-based [2, 4, 5, 6] or targetless [3, 7], offline [2] or online [7]. In self driving applications, however, even a well calibrated system needs some re-calibration due to vibration on the roads and some sensor artifacts, calling for robust online registration techniques, which are able to precisely calibrate LIDAR and camera sensors on the fly.

In this paper we propose a novel targetless fully automatic extrinsic calibration method between a camera and a Rotating Multi-Beam (RMB) LIDAR mounted on a moving car. We only have to fix the sensors on the vehicle and start driving in a typical urban environment, and the method will calculate all necessary registration parameters in situ, online. State-of-the-art competing approaches extract features for correspondence calculation from the observed *natural* environment without calibration objects. [3] transforms the range sensor's 3D measurement into a so called Bearing Angle (BA) image, and identifies point correspondences between the BA and the camera image. Alternatively, mutual Information was used in [8] to calibrate different range sensors with cameras. However, experiments show that the above techniques require a critical point density of the point cloud for reliable operation, which is not ensured at the single RMB LIDAR frames provided by a car during self-driving operation [8]. The correspondences in [7] are detected based on automatically extracted sets of lines both in the 2D images and in the 3D point clouds. According to [7] the method is preferably used indoors, where the required number of line correspondences can be often observed. However, such conditions cannot be guaranteed in RMB LIDAR point cloud frames recorded in outdoor urban environments, which are notably sparse and their density rapidly decreases as a function of the distance from the sensor. In summary, finding meaningful feature correspondences between the 3D point cloud and the 2D image domain is the main challenge in online, targetless calibration, which we aim to overcome here in a novel way (Fig. 1).

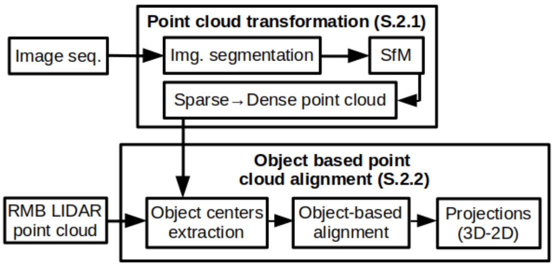


Fig. 1. Main steps of the proposed approach.

2. THE PROPOSED APPROACH

To avoid feature (2/3D interest points, line and planar segments) detection we turn to a structure from motion (SfM) based technique [9] to generate point clouds from the image

[*] This work was supported by the National Research, Development and Innovation Fund (grants K-120233 and KH-125681), and by the Széchenyi 2020 Program (grants EFOP-3.6.2-16-2017-00013 and 3.6.3 VEKOP-16-2017-00002). Balázs Nagy was partially supported by the ÚNKP-18-3 New National Excellence Program of the Ministry of Human Capacities.

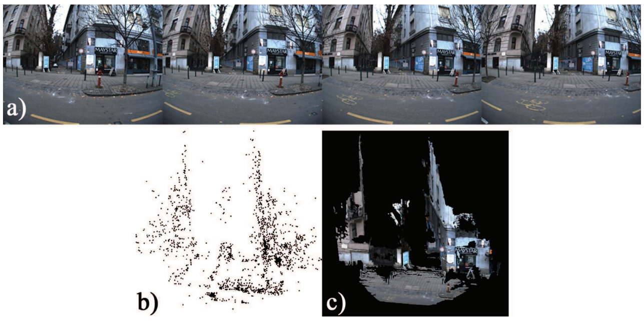


Fig. 2. (a) 4 from a set of 8 images to process. (b) Generated sparse point cloud (2041 points). (c) Densified point cloud (257796 points).

sequences recorded by the moving vehicle (Fig. 2-3), and we perform an alignment between the LIDAR and the generated point clouds. In this way, our main task can be interpreted as a point cloud registration problem (Fig. 4). Most of the conventional point level iterative registration techniques, such as variants of ICP or NDT [10], may fail when the density characteristic is quite different between the point clouds, and in our case, they can also be misled by false correspondences on the ground caused by the typical ring patterns of RMB LIDAR data. To avoid such artefacts we proposed a robust object level alignment approach between sparse RMB LIDAR point clouds and a dense reference pont map in [11, 12]. This technique extracts object blob centers in both point cloud frames, which are matched in the Hough domain, based on the idea of a fingerprint minutiae matching algorithm [13]. Although that approach is able to find a robust transformation between two point sets even if the number of points are different, it becomes sensitive to several false or inaccurate hits of the object detector, which are present in our case since both the RMB LIDAR and the SfM point clouds are quite sparse and noisy. In particulary, we observed that vehicles in the SfM clouds often fall into several pieces due to their homogeneous surfaces (Fig. 4(b)(c)), causing false matches to the Hough-based estimator [12]. The next key step is to use *semantic information* for eliminating many of the false object candidates. While object segment classification in sparse point clouds is notably challenging and often unreliable do to occlusion, we can robustly detect vehicle instances in the original camera images with deep neural networks such as *Mask R-CNN* [14] (Fig. 4(d)). By projecting the 2D class labels into 3D, vehicle points can be efficiently identified even in deficient SfM clouds (Fig. 4(e)) helping registration enhancement.

Our final aim is to find correspondences between the RMB LIDAR points and the pixels of the individual camera images. Therefore, we calculate three matrices: T_1 which projects the points of the SfM cloud onto the image domain, T_2 which transforms the LIDAR frame to the coordinate system of the SfM cloud, and T_3 to project the LIDAR point cloud directly onto the 2D image domain. The steps of the new algorithm (Fig. 1) are presented in the following subsections in details.

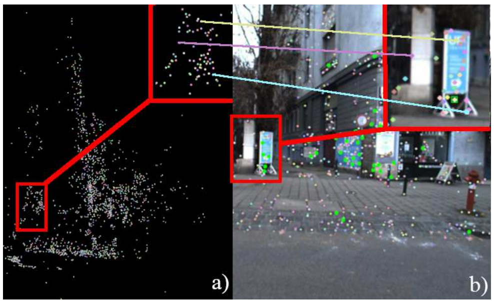


Fig. 3. (a) Sparse cloud with each point assigned a unique color. (b) One frame showing color coded 2D points that contribute to the 3D point with the same color in (a) - also showing 3 example correspondences.

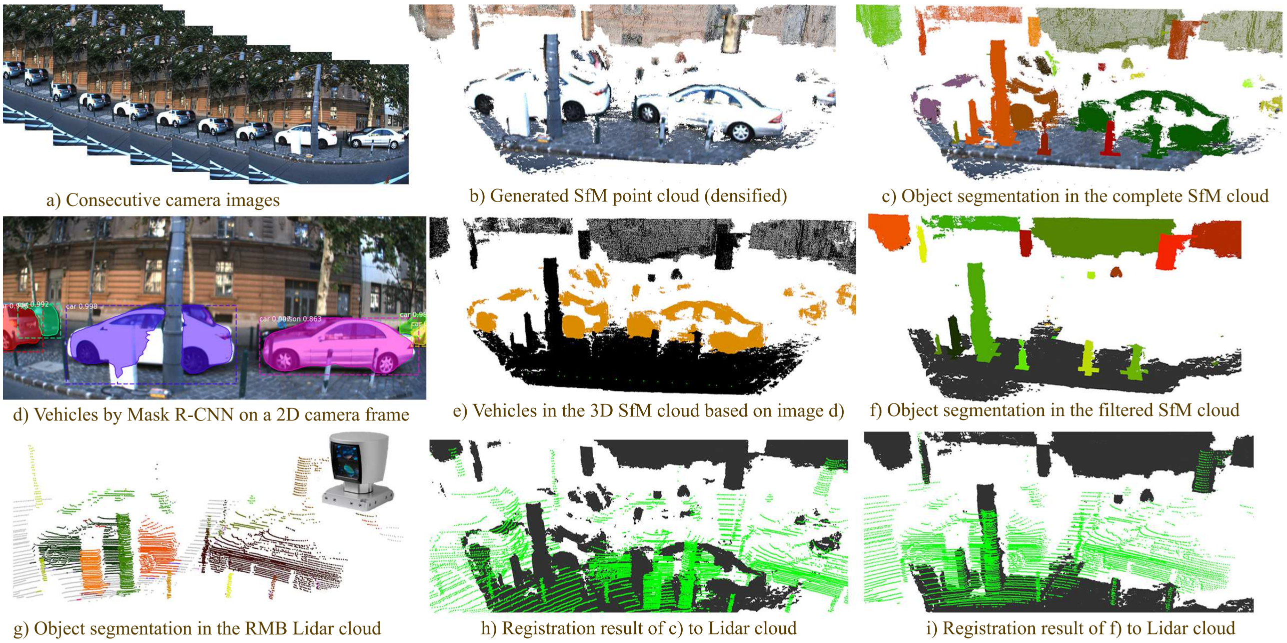
2.1. Point cloud and transformation calculation

As the first step, we generate a sparse point cloud from a continuous series of camera images, using a modified OpenMVG library [9][15], as described in the following.

We select $N \geq 3$ consecutive non-static camera frames ($N = 8$ constant in this paper, resolution is 1288×964 pixels), and feed the images into our structure from motion pipeline:

1) *Rectification*: we rectify and store the selected frames.
2) *Semantic segmentation* of the rectified frames using Mask R-CNN [14] to obtain pixel level class labels.
3) *Extraction and matching* (L_2 fast cascade matching) using SIFT feature points for the selected images.
4) *Sparse point cloud calculation*: Perform structure from motion to generate a sparse point cloud (Fig. 2(b)), then i). store the class labels - obtained in step 2 - of the feature points, and ii). assign unique IDs and store the feature points that contribute to the point cloud calculation. For each 3D point we store the 2D image points (IDs and class) from all images that contributed to the estimation of the 3D point. We also assign unique IDs to all 3D points and save their associated image points from the selected frames.
5) Using the stored 3D-2D point associations (Fig. 3(a-b)) we select M points from each frame based on point density ($M = 45$ constant), and from these 2D-3D associations we calculate the transformation T_1 using [16].
6) *Densification* of the sparse point cloud (Fig. 2(c), Fig. 4(a)) using OpenMVS [17]. This cloud and the obtained transformation will be used for alignment and registration.

The above steps can be performed on the fly, either in a loop by selecting the next N frames in a moving time window and updating the obtained transformations, or periodically, since the vehicle's movements can cause sensor displacements requiring regular updates.



a) Consecutive camera images

b) Generated SfM point cloud (densified)

c) Object segmentation in the complete SfM cloud

d) Vehicles by Mask R-CNN on a 2D camera frame

e) Vehicles in the 3D SfM cloud based on image d)

f) Object segmentation in the filtered SfM cloud

g) Object segmentation in the RMB Lidar cloud

h) Registration result of c) to Lidar cloud

i) Registration result of f) to Lidar cloud

Fig. 4. Results of the proposed object based alignment method. RMB LIDAR data is displayed with green, while the generated SfM point cloud is shown with dark grey.

2.2. Object based point cloud alignment

According to [12] we extract connected components (objects) after ground removal, i.e., we extract two sets of object centers O_1 and O_2 from the SfM-generated and the LIDAR-captured point clouds. Using an iterative voting process [13] we estimate an optimal matching between the two object sets. In the LIDAR cloud, large objects such as *facade segments* and *large vehicles* may be only partially visible providing invalid object centers. These large targets may mislead the transformation estimation, so first we eliminate them based on geometric constraints, and we only rely on compact blobs containing mainly street furniture elements such as *poles*, *traffic sings*, *trash bins* or *billboards*. Dynamic objects from the SfM point cloud are eliminated using the semantic information by Mask R-CNN as mentioned earlier.

During the transformation estimation we take into account the translation and the rotation component around the *upwards* vector. Thus, we define the problem as a 3D rigid body transformation which can be formulated as a rotation around the *upwards* vector with the proper α value and a 3D translation among the three coordinate axes.

The transformation estimation is a discrete and finite problem, so we divide the transformation space into equal bins. We address a 4D voting array $V[\alpha, d_x, d_y, d_z]$ by the α rotation value and with the calculated translation components. Iterating through all O_1 and O_2 object center pairs and rotating O_2 with all α values we can calculate the Euclidean distance between the rotated and the other center point:

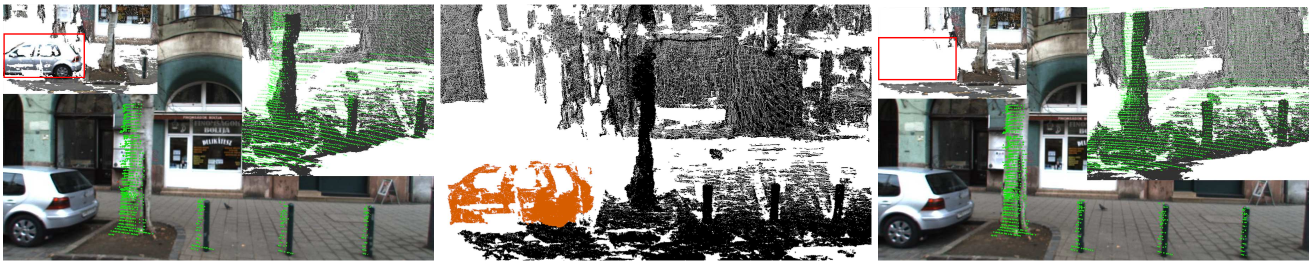
$$\begin{bmatrix} dx^* \\ dy^* \\ dz^* \end{bmatrix} = o_1 - \begin{bmatrix} \cos\alpha^* & \sin\alpha^* & 0 \\ -\sin\alpha^* & \cos\alpha^* & 0 \\ 0 & 0 & 1 \end{bmatrix} o_2$$

During the iteration we increase the evidence of each candidate, we find the maximum value in the voting array which determines the best transformation by the corresponding rotation and translation components, and we transform the LIDAR point cloud into the coordinate system of the SfM-generated cloud (Fig. 4).

At the last step we project the points of the LIDAR point cloud onto the image domain using transformation T_1 and using the inverse of T_2 we transform the LIDAR point cloud to the original position. To calculate transformation T_3 which is able to project the 3D points directly to the image domain without the intermediate transformation T_2 we use [16] (knowing the internal parameters of the camera).

3. EVALUATION

We evaluated the proposed method on a new manually annotated dataset (with ground truth) and we compared it with a state-of-the-art target based offline calibration [2] method. To demonstrate the significance of the Mask R-CNN-based semantic filtering of the SfM point cloud, we also compared two variants of the proposed method: first we matched the LIDAR frame to the raw SfM point cloud; second - as described



(a) Proposed approach based on raw SfM

(b) SfM cloud with 3D semantic information

(c) Prop. approach using Mask-R CNN filter

Fig. 5. Qualitative results of the proposed online LIDAR-camera self-calibration approach. Projections of the LIDAR points are displayed with green over the camera image. The improvement due to the Mask-R CNN filter is clearly observable by comparing (a) and (c).

Set	Method	x-error		y-error	
		Avg.	Dev.	Avg.	Dev.
Slow	Target-based reference [2]	2.87	0.47	3.57	0.86
	Prop. based on raw SfM	6.62	1.35	7.69	1.01
	Prop. using Mask-R CNN	5.35	0.98	5.97	0.65
Fast	Target-based reference [2]	4.78	1.04	6.21	1.03
	Prop. based on raw SfM	6.75	1.28	7.43	0.97
	Prop. using Mask-R CNN	5.49	1.17	5.78	0.87

Table 1. Performance analysis of the proposed automatic target-less self-calibration approach. Error values are measured in pixels. Test set names *Slow* and *Fast* refer to the speed of the data acquisition platform.

in Sec. 2 - we eliminated dynamic parts from the generated SfM point cloud, through propagating the semantic labeling information of the Mask R-CNN through the SfM pipeline.

Pixel level projection errors and standard deviations are shown in Table 1, and qualitative results are in Fig. 5. Although results show that the offline target-based calibration method can have higher accuracy, calibrating the camera and the LIDAR with [2] method is a lengthy process, taking more than 1 hour. When parameters change during measurements (e.g., sensor displacement) one needs to stop driving and repeat the offline calibration process.

Another artifact of conventional offline calibration [2] comes from platform motion: because of the nature of the RMB scanning, as the speed of the sensor increases the shape of the point cloud gets distorted. Since offline calibration can only be performed with a static vehicle, its accuracy may decrease as the car moves with higher speed. The effect of this phenomenon is also demonstrated in Table 1.

The proposed method calculates the correspondences between camera and LIDAR online during the operation of the vehicle and calculations can be repeated online periodically, thus, the average $5-6$ pixel error can be considered acceptable considering we process camera images with relatively large resolution (1288×964). At this resolution with $5-6$ pixel error we are able to robustly assign the 3D objects to the corresponding image regions using the calculated projection matrix, and this data fusion enables the autonomous vehicles to extract more visual features from the surroundings.

There can be situations when we cannot produce a robust cloud, which might increase registration errors. However, the intended use case of the proposed approach is to periodically repeat the online alignment, and only update the calibration when the current estimation improves the previously used one. Currently we perform such updates at fixed time intervals, however, in the future we also plan to include an automatic step to intelligently find suitable locations based on the current camera images.

Since the proposed approach is based on an object level alignment method, the quality of the registration is greatly depend on the amount and the type of the detected objects. Our experiments show that the proposed method performs better if the scenes contain vertical objects such as *traffic signs*, *tree trunks* and *poles*, so after the object detection we count such objects based on simple geometric constraints and we only calculate the calibration if the given scene seems appropriate. Typically in the case of main roads and larger crossroads containing several vertical landmark objects the proposed algorithm works more robustly.

4. CONCLUSION

This paper proposed a targetless camera-LIDAR sensor self-calibration approach using 2D-3D data fusion, that can be performed on the fly, and updated periodically during the data capturing process, thus eliminating the need of lengthy offline sensor calibrations. The method uses a series of camera frames, along with their semantic segmentations, from a continuous time-window and the captured LIDAR sensor data to perform automatic 2D-3D registration and alignment.

5. REFERENCES

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI Dataset,” *International Journal of Robotics Research (IJRR)*, 2013.

[2] Z. Pusztai, I. Eichhardt, and L. Hajder, “Accurate calibration of multi-lidar-multi-camera systems,” in *Sensors*, 2018, vol. 18, pp. 119–152.

[3] D. Scaramuzza, A. Harati, and R. Siegwart, “Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4164–4169, 2007.

[4] M. Velas, M. Spanel, Z. Materna, and A. Herout, “Calibration of RGB camera with velodyne LiDAR,” 2014.

[5] A. Geiger, F. Moosmann, O. Car, and B. Schuster, “Automatic camera and range sensor calibration using a single shot,” *IEEE International Conference on Robotics and Automation*, pp. 3936–3943, 2012.

[6] Yoonsu Park, Seok Min Yun, Chee Sun Won, Kyungeun Cho, Kyhyun Um, and Sungdae Sim, “Calibration between color camera and 3D LIDAR instruments with a polygonal planar board,” in *Sensors*, 2014.

[7] P. Moghadam, M. Bosse, and R. Zlot, “Line-based extrinsic calibration of range and image sensors,” *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3685–3691, 2013.

[8] R. Wang, F. P. Ferrie, and J. Macfarlane, “Automatic registration of mobile lidar and spherical panoramas,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 2012, pp. 33–40.

[9] P. Moulon, P. Monasse, and R. Marlet, “Global fusion of relative motions for robust, accurate and scalable structure from motion,” in *IEEE International Conference on Computer Vision*, Dec 2013, pp. 3248–3255.

[10] M. Magnusson, A. Nuchter, C. Lorken, A. J. Lilienthal, and J. Hertzberg, “Evaluation of 3D registration reliability and speed - a comparison of ICP and NDT,” in *IEEE International Conference on Robotics and Automation*, May 2009, pp. 3907–3912.

[11] B. Gálai, B. Nagy, and C. Benedek, “Crossmodal point cloud registration in the Hough space for mobile laser scanning data,” in *International Conference on Pattern Recognition (ICPR)*, Cancun, Mexico, 2016, pp. 3374–3379, IEEE.

[12] B. Nagy and C. Benedek, “Real-time point cloud alignment for vehicle localization in a high resolution 3D map,” in *Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving (ECCV)*, Munchen, Germany, 2018.

[13] N. K. Ratha, K. Karu, S. Chen, and A. K. Jain, “A real-time matching system for large fingerprint databases,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 799–813, Aug 1996.

[14] K. He, G. Gkioxari, P. Dollr, and R. Girshick, “Mask R-CNN,” in *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2980–2988.

[15] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, “OpenMVG: Open Multiple View Geometry,” in *Workshop on Reproducible Research in Pattern Recognition*, 2016, pp. 60–74.

[16] 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. 155166, 2009.

[17] “OpenMVS: Open Multi-View Stereo reconstruction library,” `https://github.com/cdcseacave/openMVS`.