An Interactive LiDAR to Camera Calibration

Yecheng Lyu, Lin Bai, Mahdi Elhousni and Xinming Huang
Department of Electrical and Computer Engineering
Worcester Polytechnic Institute
Worcester, MA 01609, USA
{ylyu,lbai2,melhousni,xhuang}@wpi.edu

***Abstract*—Recent progress in the automated driving system and advanced driver assistant system has shown that the combined use of 3D light detection and ranging (LiDAR) and the camera is essential for an intelligent vehicle to perceive the driving scenarios. LiDAR-camera fusion systems require precise intrinsic and extrinsic transformation between sensors. However, due to the limitation of the calibration equipment and susceptibility to noise, algorithms in existing methods tend to fail in finding LiDAR-camera correspondences in long-range terms. In this paper, we introduced an interactive LiDAR to camera calibration toolbox that estimates the intrinsic and extrinsic transform parameters. This toolbox automatically detects the corner of a planer board from a sequence of LiDAR frames and provides a convenient user interface to annotate the corresponding pixels on camera frames. Since the toolbox only detects the top corner of the board, there is no need to prepare a precise polygon planar board or a checkerboard with different reflectivity areas as in the existing methods. Furthermore, the toolbox uses genetic algorithms to estimate the transforms and supports multiple camera models such as the pinhole camera model and the fisheye camera model. Experiments using Velodyne VLP-16 LiDAR and Point Grey Chameleon 3 camera show robust results.**

***Index Terms*—LiDAR, calibration, autonomous driving system**

I. Introduction

Automated driving systems (ADS) and advanced driver assistant systems (ADAS) equipped on intelligent vehicles rely on multiple sensors to perceive their surroundings. In recent research works, LiDAR-based algorithms have shown their advantage on drivable region segmentation [8] [9], object detection [18], and simultaneous localization and mapping [19] [15]. LIDARs are also fused with cameras to improve the accuracy of 3D object detection [2]. However, perceive transformation from LiDAR and camera devices is required to assign the detections to a shared coordinate frame so that we are able fuse the sensor data. Owing to the development of multiple view geometry and computer vision, the models of 3D to 2D projection has been well established. Nevertheless, limitations are observed in existing calibration algorithms when applied to LiDAR-camera systems on intelligent vehicles.

Figure 1 shows a typical autonomous vehicle prototype. The LiDAR-camera system is designed to detect up to 100 meters in front of the vehicle. But most of the existing LiDAR-to-camera calibration algorithms are proposed for indoor use and are validated in meters rage [3] [12] [11].

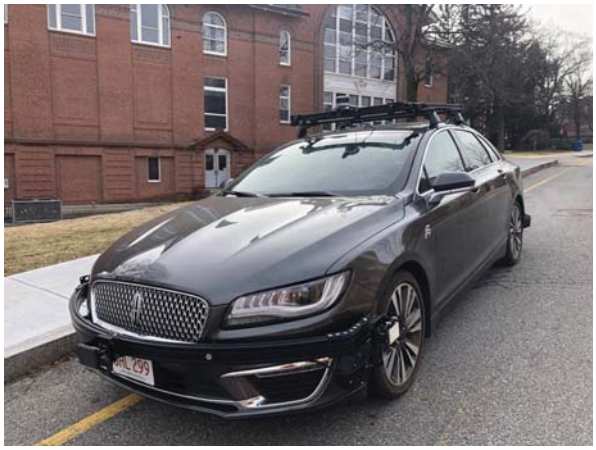


Figure 1: A Lincoln MKZ equipped with 1 LiDAR, 1 front camera and 10 side cameras

As shown in Figure 2, the increase of sensing range leads to lower resolution of the camera lens system and larger offset on LiDAR points when projected to the image plane. Thus a calibration method for the LiDAR-to-camera system is urgently required.

In addition, LiDAR sensors have limited resolution along their vertical axis, which makes it difficult to generate the point-to-point correspondences between a LiDAR frame and its associated camera frame. Existing works have proposed several indirect methods to estimate the correspondences. Rangesh et al [14] and Geiger et al [3] assumed that the checkerboards have a flat surface and that all the LiDAR points on a checkerboard should be co-planar. However, traditional checkerboards made from cardboard, wood or aluminum are not flat enough, which introduce significant noise to the calibration. To overcome this issue, [14] used a glass calibration board to ensure near-perfect flatness and rigidity. On the other hand, Park et al [11] and Pereira et al [12] employed objects in special shapes to estimate the point-to-point correspondences. However, their works were also sensible to the lack of rigidity. Due to the same reasons, target-less approaches [6] also experienced large calibration error. Moreover, due to the low vertical resolution, labeling correspondences for long-range calibration is difficult because the LiDAR sensor may not get enough rows of scans on the checkerboards or target objects. Thus few of the existing

978-1-7281-5020-8/19/$31.00 ©2019 IEEE

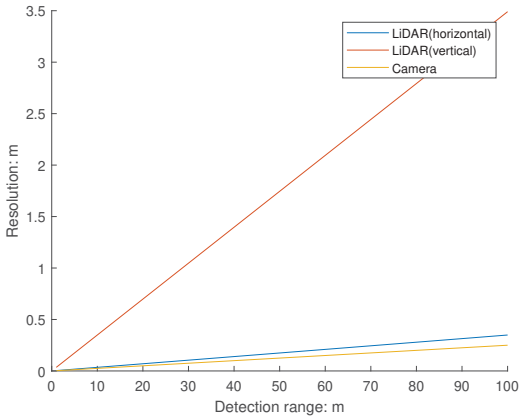


Figure 2: Resolution of LiDAR and camera. The LiDAR has resolution of $0.2°$ in vertical and $2°$ in horizontal. The scale/focus ratio of the camera is 400.

works can calibrate and validate their results for a distance longer than 5 meters.

In this paper, we introduce a new LiDAR to camera system calibration method by using a polygonal board. Through the automatic detection of the top corner point in LiDAR frames and manual labeling the correspondence pixel in camera frames, our solution collects direct point-to-point correspondences between LiDAR and camera coordinates. The correspondence are then used to estimate the intrinsic and extrinsic transforms via a genetic algorithm based approach. The rest of this paper is organized as follows: Section II discusses the related works. Section III describes the calibration models in the proposed method. In Section IV, we evaluate the proposed method on our LiDAR and Camera recordings. Section V gives the conclusions.

II. Related Work

A. LiDAR-camera correspondence collection

Acquiring the LiDAR-camera coordinate correspondence is the first step of calibration. Since point-to-point correspondences are difficult to capture, two categories of indirect approaches were proposed in the literature.

One method is to estimate the point-to-point correspondences via knowledge of specific objects. Park et al [11] estimated the vertices of the rigid checkerboards in LiDAR frames by detecting the checkerboard edges. The approach of [12] detected the surface of the target ball and estimated the position of its center. This method relies on the clarity and precision of the target objects. They also assume that an intrinsic calibration of the camera is valid.

The other method does not rely on point-to-point correspondences. Geiger et al [3] and Rangesh et al [14] assumed the LiDAR points on a checkerboard are co-planar and calibrated through planar-to-planar correspondences. Ishikawa et al [7] estimated the motion of LiDAR and camera separately and determined the intrinsic and extrinsic transform parameters through motion-to-motion correspondences. Banerjee et al [1] detected the edges of objects in camera frames and calibrated through edge-to-edge correspondences. These approaches are not only subject to the quality of calibration equipment but also rely on the segmentation of LiDAR and camera frames.

In this paper, we propose a point-to-point correspondence based approach. To overcome the limitations, we detect the vertices of target objects in a sequence of frames rather than a single frame. We also involve manual annotation of vertices in camera frames to minimize the error of correspondences.

B. Solver of calibration

Different optimizers have been used to estimate the parameters in the calibration models. Hulik et al [5] used 3D Hough transform to search for a continuous plane in point clouds. Velas et al [17] and Vasconcels et al [16] applied the RANSAC method to solve the extrinsic calibration. Paynot et al [13] and Zhang [20] introduced a probabilistic model to estimate the likelihood of a transformation and then iterated using the Levenberg–Marquardt algorithm. Heikkila et al [4] also used the Levenberg–Marquardt algorithm but applied direct linear transform (DLT) to solve the initial estimation of camera intrinsic parameters. Pandey et al [10] estimated the extrinsic calibration based upon the maximization of mutual information. In our work, a genetic algorithm is applied to estimate the parameters in extrinsic and intrinsic transforms since it works for non-linear models and avoids local optimal.

III. Methodology

This section gives an overview of our work on LiDAR to camera calibration. In this section, we first demonstrate the models applied in the proposed calibration, describe the data collection and processing, and then present the use of the genetic algorithm to solve the calibration task.

A. Calibration models

Models of LiDAR to camera projection have been well investigated. Suppose we have a scanned point (x, y, z) in LiDAR coordinates, its corresponding point (u, v, w) in camera coordinates, and its corresponding point (i, j) in the image plane. The transformation from LiDAR to the image includes two parts, as described in [4] [3].

The first one is the extrinsic transformation that is the projection model from LiDAR to camera coordinate. This 6-DOF matrix can be expressed as (1).

978-1-7281-5020-8/19/$31.00 ©2019 IEEE

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & u_0 \\ R_{21} & R_{22} & R_{23} & v_0 \\ R_{31} & R_{32} & R_{33} & w_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{1}$$

where$R \in R^{3\times3}$ is the rotation matrix and $t \in R^{3\times1}$ is the translation vector. Since it is a linear projection and both coordinates share the same unit (meter), the rotation matrix R equals a multiplication of three sub-rotation matrix R_{roll}, R_{pitch} and R_{yaw}, as shown in (2).

$$\begin{aligned} R &= R_{roll}R_{pitch}R_{yaw} \\ R_{roll} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix} \\ R_{pitch} &= \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix} \\ R_{yaw} &= \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2}$$

Where α, β, and γ are the rotation angle along the x, y and z axis. The goal of the extrinsic matrix calibration is to estimate the 6 parameters $(\alpha, \beta, \gamma, u_0, v_0, w_0)$.

The second part is the intrinsic transformation that projects the 3D points in camera coordinate to the 2D image plane. Pinhole camera model and fisheye camera model are two commonly used intrinsic models. The pinhole model is described in (3).

$$\begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} f_x/w & 0 & i_0 \\ 0 & f_y/w & j_0 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{3}$$

Where f_x and f_y are the focus lengths of the lens system along x and y axis, and i_0, j_0 are the offsets on the target image plane. For the fisheye model, lens distortion and tangential distortion and skew are also considered as in (4).

$$\begin{aligned} x_d &= (1 + k_1r^2 + k_2r^4 + k_5r^6) \begin{bmatrix} u/w \\ v/w \end{bmatrix} \\ dx &= \begin{bmatrix} 2k_3uv + k_4(r^2 + 2u^2) \\ k_3(r^2 + 2v^2) + 2k_4uv \end{bmatrix} \\ \begin{bmatrix} i \\ j \end{bmatrix} &= \begin{bmatrix} f_x & \alpha_c \cdot f_x & i_0 \\ 0 & f_y & j_0 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \end{aligned} \tag{4}$$

Where $r = \sqrt{u^2 + v^2}$.

Thus, an intrinsic transformation using pinhole model has 4 parameters (f_x, f_y, i_0, j_0) and the one using fisheye has 10 parameters $(f_x, f_y, i_0, j_0, \alpha_c, k_1, k_2, k_3, k_4, k_5)$.

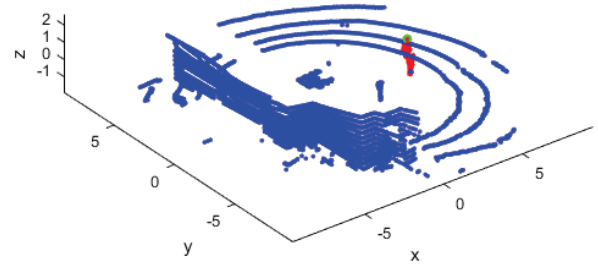
B. Data collection and processing

The key issue of point-to-point based approaches in LiDAR to camera calibration is to collect the point-to-point pairs between LiDAR and camera coordinates. Since LiDAR point cloud is sparse on the vertical axis, it is difficult to get the target correspondence on a checkerboard in a single frame. In our approach, we detect the vertices of the checkerboard in a sequence of LiDAR frames.

The proposed approach has the following assumptions: (1) only the checkerboard and the tester holding the checkerboard exist in the region of interest (ROI), and (2) the target corner of the checkerboard is the highest point in the ROI. Comparing to other related works, our approach does not require a perfect checkerboard nor lots of scan points on the checkerboards. The proposed approach not only reduces the difficulty of calibration task but also makes it capable of calibrating for low-resolution LiDAR devices such as 4-line LiDAR and 16-line LiDAR. The procedures to detect the vertex in a LiDAR frame sequence are described in Algorithm 1.

When the toolbox starts, it shows a LiDAR frame and the user needs to input the ROI boundaries. For the subsequent LiDAR frames, the toolbox segments the points inside the ROI, validates the size of segmented object and searches for the top line inside the ROI. If the top line has only one point, it is picked up as a corner points, otherwise the LiDAR frame is dropped. For each frame in which a corner point is detected, the toolbox asks the user's confirmation and

978-1-7281-5020-8/19/$31.00 ©2019 IEEE



(a)

(b)

Figure 3: Example LiDAR frame (a) and example camera frame (b)

Algorithm 1 vertex detection in LiDAR frame sequence

Input: LiDAR Frame Sequence $\{P\}$, LiDAR Frame Sequence $\{P\}$, Region of interest ROIn of interest ROI

for LiDAR frame P_i in $\{P\}$ **do**
  Get point cloud in the ROI: $P_i \leftarrow \{p|p \epsilon P_i \cap ROI\}$
  if max_line(P_i) - min_line(P_i) $\geqslant$3 **then**
    $\hat{P}_i \leftarrow p|line(p) = max\_line(P_i)$
    if Count($\hat{P}_i$)=1 **then**
      return$\hat{P}_i$
    else then
      drop frame P_i
    end if
  end if
end for

presents the corresponding camera frame for annotation if a valid vertex is detected. In practice, the LiDAR and camera sensors might not be perfectly synchronized, so the toolbox select the camera frame that is captured closest to time of the LiDAR frame. After each annotation, the toolbox generates a new point-to-point correspondence between the LiDAR and camera coordinate.

Algorithm 2 Genetic Algorithm based transformation parameter optimization

Input: Initial parameter set $Param$, bounding shift $Bound$, bounding scale $Scale$

for iteration $Iter$ **do**
  Upper Bound $ub \leftarrow Param + Bound$
  Lower Bound $lb \leftarrow Param - Bound$
  $\hat{P}_i \leftarrow p|line(p) = max\_line(P_i)$
  for slot $Slot$ **do**
    $Param_i, error_i \leftarrow GA(Param, ub, lb)$
    $Param \leftarrow \{Param_i|error_i = min(error)\}$
    $Bound \leftarrow Scale \times Bound$
  end for
end for
return $Param$

C. Genetic Algorithm

The proposed toolbox calibrates the extrinsic and intrinsic transformations of LiDAR to camera through the genetic algorithm (GA). GA is a widely used algorithm in parameter estimation since it is model-based, data-driven and robust to non-linear optimization. In theory, a GA instance rolls out thousands of alternative parameter sets and selects certain candidates with least offset as seeds of the next generation. The algorithm terminates when determined generations are reached or the convergence goal is reached. Comparing to gradient-based algorithms, GA is capable to avoid local optima.

In this toolbox, we employe the MATLAB genetic algorithm toolbox as the GA solver. According to the calibration model, we set 10 parameters for pinhole-model based optimization and 16 parameters for fisheye-model based optimization. The mean offset of labeled points projected from LiDAR to the image plane is selected as the target loss of GA. The GA process iterates multiple generations to minimize the loss. The GA optimizer keeps an 800 population and runs up to 30 generations. In the first iteration, the upper bound and lower bound of GA slots are manually determined. During the rest of the iterations, the toolbox narrows the upper bound and lower bound according to the optima of last iteration and the number of iterations it has executed. The iteration algorithm is described in Algorithm 2.

IV. Experiment

In the experiment, we use the proposed toolbox to calibrate the LiDAR-camera system of our autonomous vehicle prototype. The system as shown in Figure 4 contains a Velodyne VLP-16 LiDAR and a Point Grey Chameleon 3 camera. Both devices are installed on the roof rack facing the front. The LiDAR has 16 row scanners covering a $[-15°, 15°]$ vertical field of view and $360°$ horizontal one resulting in a $0.2°$ horizontal resolution and a $2°$ vertical resolution.

To calibrate this LiDAR-camera system, we use a 2×3 feet rigid board rotated $45°$ so that one corner is up to the sky. A tester holds the board and moves around in a $10m \times 30m$ field in front of the car. We recorded a 219 seconds data

978-1-7281-5020-8/19/$31.00 ©2019 IEEE

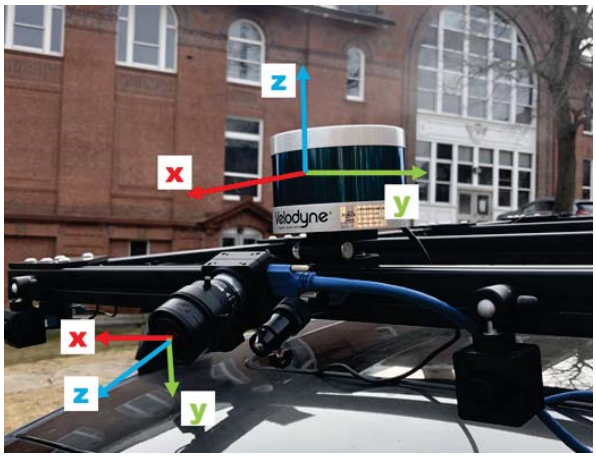


Figure 4: LiDAR-camera system on an Lincoln MKZ

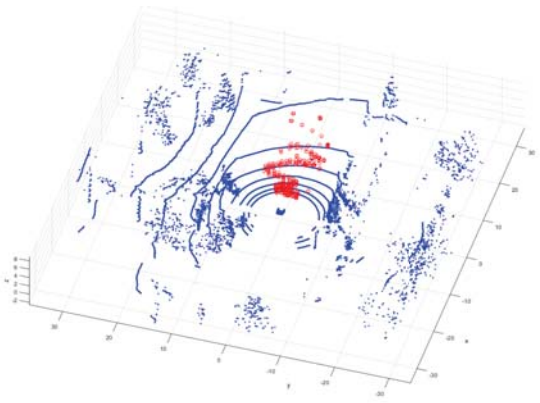


Figure 5: Labeled data in LiDAR coordinate frame. The blue dots are the point cloud from one LiDAR frame, and the stems present the position of labeled correspondences in the LiDAR coordinate.

sequence containing 11582 camera frames and 7639 LiDAR frames. After processing, 222 point-to-point correspondences between LiDAR and camera coordinate frames are generated as shown in Figure 5. The lower bound and upper bound setting is manually determined as shown in Table I based on the device setting and installation. For validation, we record another data sequence with 3293 camera frames and 2177 LiDAR frames, and annotated 358 correspondences. For the training data, this calibration method is able to achieve 6.35 pixel offset in average for pinhole camera model and 5.02 pixel for fisheye camera model. On the test data, it results 6.75 pixel error in average using pinhole camera model and 5.1 pixel using fisheye camera model. An example projection result is shown in Figure 6. The result shows that the proposed approach generates precise transformation from LiDAR to camera coordinate. Figure 7 and 8 shows the validation of the LiDAR-camera transformation along the LiDAR coordinate. It is obvious that the offset around the camera normal is small, but as the angle to the camera normal grows, the offset grows to as much as 16 pixels. In practical,

parameter	α	β	γ	u_0	v_0
lower boundary	0.2π	-0.8π	-0.3π	-1	-1
upper boundary	0.8π	-0.2π	0.3π	$+1$	$+1$
parameter	w_0	f_x	f_y	i_0	j_0
lower boundary	-1	300	300	300	300
upper boundary	$+1$	900	900	900	900
parameter	k_1	k_2	k_3	k_4	k_5
lower boundary	-1	-1	-1	-1	-1
upper boundary	$+1$	$+1$	$+1$	$+1$	$+1$

Table I: Boundaries of parameters for GA solver

Figure 6: Example calibration result

the calibrated LiDAR-to-camera transform is precise enough for ADS applications on the test car.

For further testing, we drive the test vehicle on the road and record a 1043-frame LiDAR and camera sequence. From the data, we detect the lane markers by thresholding the intensity of LiDAR points in the lowest ring. We then project the detected lane markers to the corresponding camera frames using the calibrated transformation function and visualize them on the camera frames with red asterisks. As shown in Figure 9, the offset of the lane markers is acceptable for automated driving.

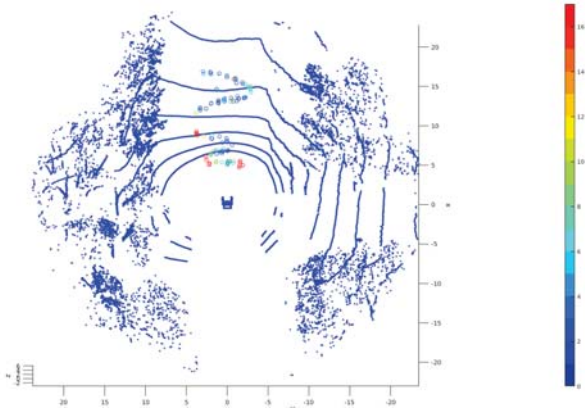


Figure 7: Validation of LiDAR-camera calibration using pinhole camera model

978-1-7281-5020-8/19/$31.00 ©2019 IEEE

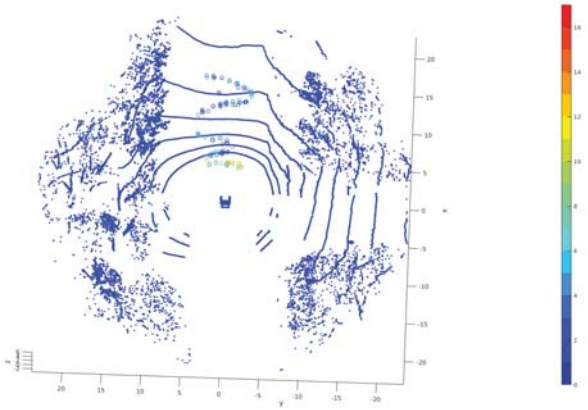


Figure 8: Validation of LiDAR-camera calibration using fisheye camera model

Figure 9: Lane markers detected in LiDAR frames and projected to Camera frames

V. Conclusions

In this paper, we propose a novel and accurate approach for LiDAR-to-camera system calibration. This approach is not sensitive to the checkerboard quality and works for long-range calibration. We also develop a MATLAB based toolbox to calibrate the LiDAR and camera. The toolbox automatically detects the vertex in a LiDAR frame sequence and provide a convenient user interface to annotate the correspondence in camera frames. A genetic algorithm based approach is applied to estimate the extrinsic and intrinsic parameters. Experiment on test vehicle shows that the toolbox works well on long range calibration and achieves good results.

Future work will investigate automatic corner detection on camera frames, extend the calibration to LiDAR-thermal camera systems, and investigate the correspondences of 2D to 3D projection.

VI. Acknowledgement

This work is partially supported by U.S. NSF Grant CNS-1626236.

References

[1] Koyel Banerjee, Dominik Notz, Johannes Windelen, Sumanth Gavarraju, and Mingkang He. Online Camera LiDAR Fusion and Object Detection on Hybrid Data for Autonomous Driving. *2018 IEEE Intelligent Vehicles Symposium (IV)*, (Iv):1632–1638, 2018.

[2] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *IEEE CVPR*, volume 1, page 3, 2017.

[3] Andreas Geiger, Frank Moosmann, Ömer Car, and Bernhard Schuster. Automatic camera and range sensor calibration using a single shot. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3936–3943, 2012.

[4] Janne Heikkila, Olli Silvcn, and Infotech Oulu. A Four-step Camera Calibration Procedure with Implicit Image Correction. pages 1106–1112, 1997.

[5] Rostislav Hulik, Michal Spanel, Pavel Smrz, and Zdenek Materna. Continuous plane detection in point-cloud data based on 3D Hough Transform. 2014.

[6] Kiyoshi Irie, Masashi Sugiyama, and Masahiro Tomono. Target-less camera-LiDAR extrinsic calibration using a bagged dependence estimator. *IEEE International Conference on Automation Science and Engineering*, 2016-Novem:1340–1347, 2016.

[7] Ryoichi Ishikawa, Takeshi Oishi, and Katsushi Ikeuchi. LiDAR and Camera Calibration using Motion Estimated by Sensor Fusion Odometry. apr 2018.

[8] Y. Lyu, L. Bai, and X. Huang. Real-time road segmentation using lidar data processing on an fpga. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, May 2018.

[9] Yecheng Lyu, Lin Bai, and Xinming Huang. Chipnet: Real-time lidar processing for drivable region segmentation on an fpga. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2018.

[10] Gaurav Pandey, James R. McBride, Silvio Savarese, and Ryan M. Eustice. Automatic Extrinsic Calibration of Vision and Lidar by Maximizing Mutual Information. *Journal of Field Robotics*, 2015.

[11] Yoonsu Park, Seokmin Yun, Chee Sun Won, Kyungeun Cho, Kyhyun Um, and Sungdae Sim. Calibration between color camera and 3D LIDAR instruments with a polygonal planar board. *Sensors (Switzerland)*, 2014.

[12] Marcelo Pereira, David Silva, Vitor Santos, and Paulo Dias. Self calibration of multiple LIDARs and cameras on autonomous vehicles. *Robotics and Autonomous Systems*, 2016.

[13] Thierry Peynot and Abdallah Kassir. *Laser-Camera Data Discrepancies and Reliable Perception in Outdoor Robotics*. 2010.

[14] Akshay Rangesh, Kevan Yuen, Ravi Kumar Satzoda, Rakesh Nattoji Rajaram, Pujitha Gunaratne, and Mohan M. Trivedi. A Multimodal, Full-Surround Vehicular Testbed for Naturalistic Studies and Benchmarking: Design, Calibration and Deployment. sep 2017.

[15] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. 2018.

[16] Francisco Vasconcelos, Joao P Barreto, and Urbano Nunes. A Minimal Solution for the Extrinsic Calibration of a Camera and a Laser-Rangefinder. Technical report.

[17] Martin Velas, Michal Spanel, Zdenek Materna, and Adam Herout. *Calibration of RGB Camera With Velodyne LiDAR*.

[18] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[19] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, page 9, 2014.

[20] Zhengyou Zhang. A Flexible New Technique for Camera Calibration; a typo in Appendix B) (last updated on Aug A Flexible New Technique for Camera Calibration. Technical report, 2008.

978-1-7281-5020-8/19/$31.00 ©2019 IEEE