Features in the Point Cloud: An Automatic Approach to High Density LiDAR to Camera Calibration

Tianyi Zhang[1], Eduardo Iscar[2], Matthew Johnson-Roberson[2]

***Abstract*— In this paper, we introduce a novel method to estimate relative pose calibration parameters between the coordinate frames of high-density LiDAR sensors and optical cameras. The main contribution of this paper is the introduction of rendered synthetic LiDAR images using the point cloud reflectivity information that enable the use of 2D feature detectors to match calibration tag corners between the camera image and a dense LiDAR point cloud. Experimental results are evaluated by synthetic tests as well as real data collected using a Livox Mid-40 LiDAR and MYNT EYE D camera. Synthetic tests quantitatively demonstrate low error in transformation under a certain range of noise level. Moreover, our algorithm doesn't require measurement of the physical size of calibration targets, which avoids the measurement error in real-world applications. Alignment on real data between the modalities shows qualitatively well aligned visual results.**

I. INTRODUCTION

With the evolution of perception systems on autonomous vehicles and mobile robots, the combination of cameras and Light Detection And Rangings (LiDARs) plays an increasing role in sensing the environment. Cameras and LiDARs provide information in complementary structures: cameras project the three-dimensional (3D) environment into two-dimensional (2D) space with color and texture; LiDARs measure the 3D environment and output a metric point cloud. The alignment of data from cameras and LiDARs is a fundamental requirement for higher level detection, navigation, mapping, and 3D reconstruction algorithms, but remains a challenging problem because the modalities observe the world in different ways.

The increase in popularity of LiDAR technology with the development of autonomous vehicles has boosted the development of higher resolution, lower cost devices utilizing novel technologies such as non-repetitive scanning LiDAR [1]. This in turn is driving an increase in adoption by researchers and industry. LiDAR technology is being deployed on an increasing number of robotic platforms and domains. As a consequence, robust and fully automated calibration algorithms have to be developed to integrate LiDAR devices with the rest of a robots sensing technology.

This work has been motivated by the realization that reflectivity, a value associated with each point in the laserscan that measures the strength of the returned signal can be cross-registered with the intensity captured by an optical camera in the visible light spectrum [2]. This characteristic can be exploited to design appropriate calibration targets whose features can be detected in both the LiDAR and camera data to establish correspondences between the sensors.

In this work we introduce a novel calibration method based on the rendering of synthetic images from a high-resolution LiDAR point cloud. Robust computer vision target detection [3] and pose estimation algorithms [4] developed for 2D imagery can be applied on the rendered images to obtain feature correspondences between the camera images and the LiDAR point cloud, enabling the estimation of the sensor relative pose. The introduced method is fully automated and does not require operator intervention. Results on real data are presented together with an analysis of the method's robustness.

The main contributions of this paper are 1) The introduction of synthetic LiDAR images to enable cross-modality feature identification and matching, 2) Its application for dense LiDAR to camera calibration problem without bringing measurement error of target size into the loop. Both quantitative and qualitative results are evaluated and validated on experimental data.

The rest of the paper is structured as follows: Section II reviews previous work in the field of LiDAR to camera calibration. Section III describes the developed calibration procedure, while Section IV introduces the experimental evaluation of the method. Section V outlines our conclusions and future work.

II. RELATED WORK

The calibration of LiDAR and camera systems consists in the estimation of the transformation that aligns the two measurement coordinate systems. Early work in the field was done by Zhang and Pless [5]. The authors used a checkerboard calibration target to estimate extrinsics between a pinhole camera and a 2D range finder by constraining the laser scan points to lie on the planar surface observed in the camera image. Their work was extended by Unnikrishnan et al. [6] into the first 3D LiDAR to camera calibration toolbox. Their method requires the user to manually selects the points in the LiDAR point cloud to locate the calibration checkerboard plane. Once identified, a plane to plane correspondence is imposed on the chosen points to build constraints. Such manual selections slow down the calibration procedure and introduce the possibility for operator error. Latter methods extend the concept of corresponding planes between the camera and LiDAR data. Pandey et al. [7] proposed a calibration routine for LiDAR and omnidirectional cameras. Zhou and Deng [8] introduced a virtual coordinate frame

[1]T. Zhang is with the Robotics Institute, University of Michigan, Ann Arbor, MI 48109, USA tyz@umich.edu

[2]E. Iscar, and M. Johnson-Roberson are with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI 48109 USA {eiscar,mattjr}@umich.edu

on the checkerboard target and used it to estimate rotation and translation decoupled from each other. Geiger et al. [9] developed an algorithm capable of identifying multiple checkerboards simultaneously with one shot. Simultaneous detection of multiple targets enables faster data collection, which is supported by the method introduced in this paper as well. In addition to plane to plane constraints, Zhou et al. [10] further proposed a method that incorporates plane to plane constraints and edge to line constraints. Their method requires a single checkerboard observation, greatly reducing calibration data acquisition time.

All the camera calibration methods introduced above employ a checkerboard as the calibration target. In such methods the cameras detect the calibration target corner points and can thus identify the surface, while the 3D spatial distribution of point clouds is used to identify the plane in the LiDARs frame of reference. Most frequently, at least three such corresponding observations are required in order to be able to recover the camera to LiDAR relative pose. In our proposed target-based algorithm, instead of establishing 2D plane to plane constraints between the camera and LiDAR we match point features between the image and point cloud, requiring less data to constraint the calibration problem. Such 3D-2D point correspondence is also employed by [11]–[13]. In comparison, our method uses standardised calibration target, and detects features without human intervention nor measurement on the physical size of target to avoid introducing errors into our system.

In addition to checkerboards, variations of target pattern and geometry are introduced in [14]–[17] to increase detection precision. However, the increase in complexity of the target geometry negatively affects the ease of use and robustness of the algorithms. In order to develop a robust method the proposed calibration procedure uses printed AprilTag targets. These targets have been extensively used for the calibration of optical cameras [18] and shown to outperform the accuracy of checkerboard based methods [3].

Other approaches have attempted to exploit intensity values in a LiDAR point cloud in addition to geometric information. Pandey et al. [2] proposed to correspond LiDAR reflectivity with intensity in the camera image, maximizing the mutual information. Irie et al. [19] expanded the method including point cloud norms and discontinuity. Instead of global alignment of LiDAR and camera intensity, our approach uses LiDAR reflectivity to extract and identify feature points from point clouds of the calibration target, avoiding the need for costly global optimization.

III. METHODOLOGY

This section details our proposed LiDAR-camera calibration method to estimate the rotation R_l^c and translation T_l^c between the LiDAR and camera coordinate frames, as shown in Fig. 1. Our method is based on the observation of known calibration targets by both the camera and LiDAR and requires matching features between the camera image and LiDAR point cloud. Feature extraction in 2D images [20]–[23] is a well studied topic in the robotics community. However much less work has been done on 3D point cloud feature detection due to the lack of texture and the discrete nature of LiDAR data. In order to leverage 2D feature extraction algorithms, in this work we propose to render a synthetic image from the LiDAR point cloud using the reflectivity values. This allows us to identify the calibration targets in the point cloud as well as in the camera image and compute the required 3D-2D matches. The relative pose between the two sensors is finally obtained using the Perspective-n-Point algorithm (PnP) [4]. The complete process is shown in Fig. 2.

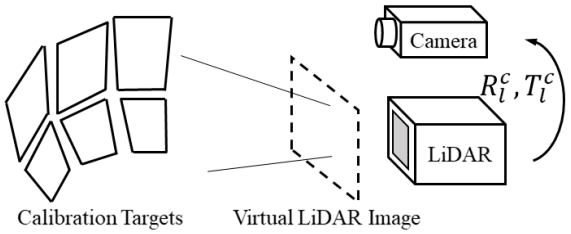


Fig. 1: Configuration of targets and sensors. The LiDAR and camera are rigidly mounted together. 3D LiDAR points can be projected onto a virtual image plane in LiDAR coordinate.

A. Targets and Sensors Setup

1) Calibration Targets: In order to obtain robust and salient features in both the camera and LiDAR data we employ AprilTag calibration targets. AprilTags can be uniquely identified by their pattern, enabling the use of multiple tags in the same scene for better coverage and quicker calibration.

2) Camera: The camera intrinsic parameters (focal length, center of projection, and distortion coefficients) need to be known a priori and can be estimated with a conventional camera calibration process [18], [24].

3) LiDARs: Our algorithm is developed for non-repetitive scanning LiDARs that can provide accumulated high resolution scans of the calibration targets. Our experience suggests that LiDARs with a scanning density of 100 pts/deg^2 are required for the proposed method to work reliably.

B. Feature Detection

1) LiDAR synthetic image: In order to render the LiDAR synthetic image, we construct a virtual camera projection model to match the LiDAR Field Of View (FOV). The aspect ratio of the virtual camera image is chosen to match the scanning footprint of the LiDAR, shown in Fig. 3. The center of the projection (c_x, c_y) is defined to be at the center of the image. Image resolution has to be chosen smaller than the point cloud resolution to ensure most pixels in the image correspond to at least a point in the point cloud. If the selected resolution is too low, artifacts and poorly defined tag edges will be generated. Specific values will thus depend on the LiDAR and distance to target used.

Given the LiDAR FOV angle α_{fov}, specified in the device data sheet, we can compute minimum focal length, f_x and f_y, required to ensure all 3D LiDAR points are projected into the image. From the camera projection model [25], we can

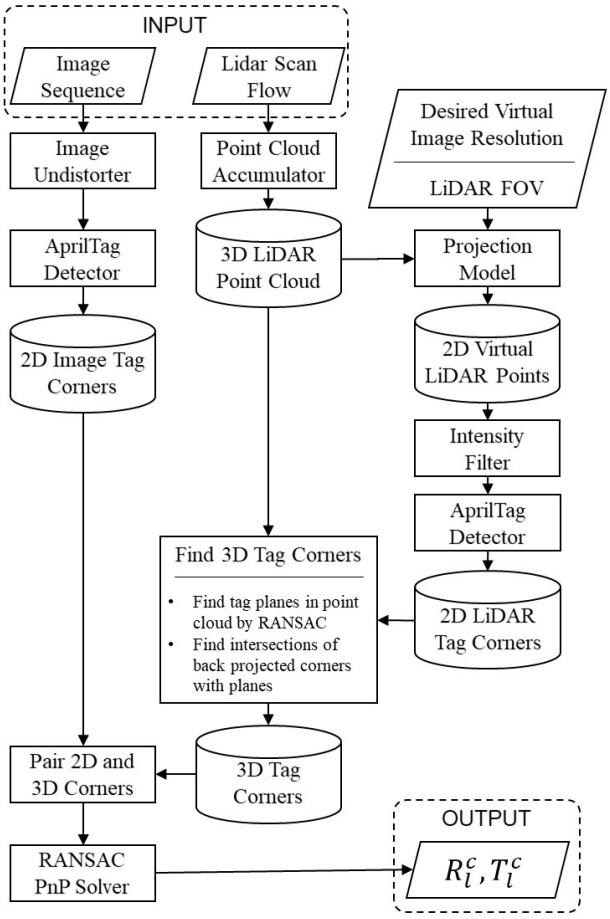


Fig. 2: Flow Chart of our algorithm: Left column describes how camera image stream is processed; center column is the flow of raw LiDAR data. With projection model (column in the right), features are extracted in the point cloud. 2D and 3D features are further paired and fed to the PnP solver.

derive the following relationship between the focal length, image size $l_w \times l_h$ and FOV angle:

$$\frac{\frac{1}{2}l_w}{f_x} = \frac{\frac{1}{2}l_h}{f_y} > tan(\frac{1}{2}\alpha_{fov}) \tag{1}$$

Once the virtual camera intrinsic parameters have been determined, the 3D points can be projected into the image as:

$$Z_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{2}$$

where (X_i, Y_i, Z_i) is the 3D coordinate of i^{th} LiDAR point, and (u_i, v_i) is the 2D coordinate of the corresponding point projected onto the virtual LiDAR image.

For each projected point (u_i, v_i), we set the brightness of the corresponding pixel, $B(u_i, v_i)$, according to the following rule:

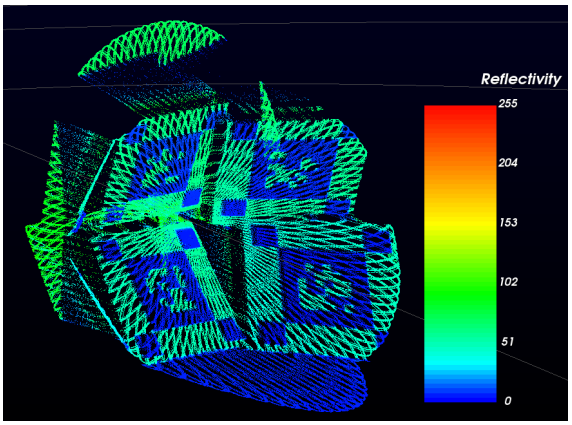


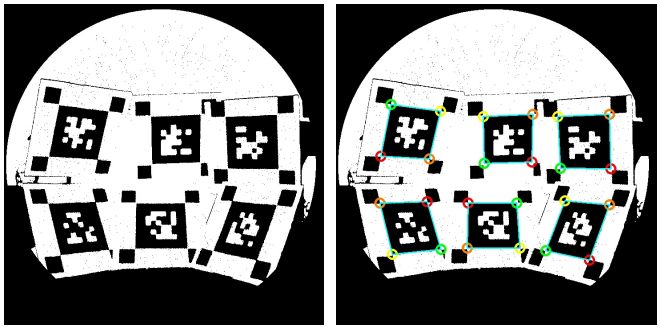
Fig. 3: Point cloud reflectivity can be used to identify the black and white areas on the AprilTag targets. The white parts on the targets have a reflectivity value around 100, while on the black parts, reflectivity values are lower than 10.

$$B(u_i, v_i) = \begin{cases} 0 & \text{if } \phi \leq \phi_{thr} \\ 1 & \text{if } \phi > \phi_{thr} \end{cases}$$

where ϕ is the reflectivity value associated with each P_i and ϕ_{thr} is a fixed threshold value. Median filtering with a 3×3 kernel or image down-sampling with nearest neighbor interpolation can be applied to help improve image quality in the case of high noise in the rendered image. Fig. 4a shows an example synthetic image.

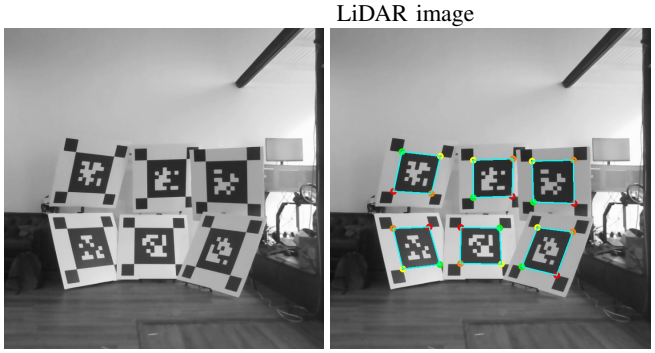
2) Point cloud corner detection: 2D AprilTag detection algorithms can be applied to the synthetic rendered image to find the corners of the tags. Fig. 4b shows the detected AprilTag corners on six calibration boards. Once the corner positions have been found it is required to map them back to 3D space. We first extract and parameterize 3D target planes from 3D LiDAR cloud with RANdom SAmple Consensus (RANSAC) algorithm [4]. Then we project 2D tag corners back into 3D space. The 3D corner points $\mathbf{C}$, where $C_j^k \in \mathbf{C}$ is the j^{th} corner of the k^{th} tag in one observation, are localized as the intersections of target planes and back projection rays. This method doesn't requires any knowledge on the size of the tags which avoids hand-measure errors. However, in the case that we have confidence in knowing the physical size of tags, 3D corners can also be estimated by PnP algorithm as an alternative.

3) Camera corner detection: The calibration targets are detected in the camera image using the same detector used on the virtual image. This will provide a set of 2D corner coordinates. We notate the set of camera observed tag corners as $\mathbf{c}$, where $c_j^k \in \mathbf{c}$ is the j^{th} corner of the k^{th} tag in one camera observation in 2D image coordinates. Fig. 4c shows a raw camera image of the AprilTags, while Fig. 4d shows the extracted corners in the same image.



(a) Synthetic LiDAR image

(b) Corners detected on synthetic LiDAR image



(c) Raw camera image

(d) Tag corners detected on raw camera image

Fig. 4: Detection of tag corner points on camera image and generated LiDAR images, observed at 4.5m from targets. Corners on one tag can be identified by tag orientation as shown by the matching corner color. Corners on different tags can be identified by AprilTag ID.

C. Extrinsic Parameters Estimation

The uniqueness of the AprilTag calibration targets allows us to pair tag corner detections $\mathbf{C}$ from the LiDAR and camera corner detections $\mathbf{c}$. This generates a set of corresponding 3D LiDAR corners and 2D camera corners (C_j^k, c_j^k).

The problem of estimating the rotation and translation to align 2D observations with 3D points has been extensively studied in the computer vision literature and is known as PnP. PnP and RANSAC algorithms [4] are combined to find the rotation R_l^c and translation T_l^c that aligns the camera observations with the LiDAR coordinate frame. Given camera matrix K the PnP problem can be formulated as the following equation for each 2D-3D point pair:

$$s \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = K_c \left[R_l^c | T_l^c \right] \begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} \tag{3}$$

where $(u', v') \in \mathbf{c}$, $(X', Y', Z') \in \mathbf{C}$ and s is the scale factor. The above system of equations can be solved using Levenberg-Marquardt optimization. RANSAC is integrated into the PnP solver to rejected outliers in the 3D to 2D point pairs. Implementations are provided by open source libraries such as OpenCV.

To visualize the calibration result, we can map the 3D point cloud to the camera image pixels, then generate colored point cloud with corresponding color from the image. A good calibration aligns colors and shape reasonably.

IV. EXPERIMENTS

In this section, we evaluate our proposed algorithm in comparison with Huang's method [13] and Zhou's method [10]. Since ground truth of sensor calibration is a chicken-and-egg problem, we did the quantitative comparison by synthetic tests. We qualitatively validate our real-world calibration result by mapping LiDAR cloud to camera image, and examine both the color-intensity alignment in 2D image and color-shape alignment in 3D cloud.

1) Synthetic Tests: Huang's method employs similar idea with ours, which first identifies the 3D pose of target from LiDAR then solve relative camera pose with PnP, but features a "Soft L1 Norm" cost function. Zhou's method fully exploits the widely studied plane-plane and line-line correspondence and significantly reduces the number poses needed to achieve state-of-the-art accuracy. We used AprilTags from the 36h11 family as our calibration target. The target IDs are encoded in 36 bits with a minimum Hamming distance of 11 bits between family members. The LiDAR virtual image is initialized with a size of $l_w \times l_h = 1000 \times 1000$ and focal length $f_x = f_y = 1400$. We tested the algorithms with configurations of number of tags ranges 1 to 20 and zero mean Gaussian noise in LiDAR measurement with deviations of 0.1%, 0.2% and 0.3% of the range measurement value. We collected mean and median of rotation and translation error from 100 runs of each configuration with randomly generated camera-to-LiDAR poses. Specifically, camera's rotation angles and translation distances are uniformly distributed within $20\,\mathrm{deg}$ and $1m$ respectively, both in randomly distributed directions w.r.t. LiDAR coordinate frame. Given the ground truth transformation $(\mathbf{R}_{GT}, \mathbf{t}_{GT})$ and estimation $(\mathbf{R}, \mathbf{t})$, the rotation error can be represented by the following formula [26]:

$$\alpha = cos^{-1}((trace(\mathbf{R}_{GT}\mathbf{R}^T) - 1)/2) \tag{4}$$

We calculated the translation error by its magnitude $||\mathbf{t}_{GT} - \mathbf{t}||_2$.

As shown in Fig. 5, our algorithm achieves its best performance with no less than 10 tags and $\sigma \leq 0.2\%$. Specifically, our algorithm outperforms the rivals with noise level of $\sigma = 0.1\%$, while provide similar accuracy with $\sigma = 0.2\%$. Drawbacks of our algorithm are also revealed: 1) our algorithm gives underfitting results with only one tag observation, since only 4 corners are involved in solving PnP problem, 2) the translation error grows significantly with increase of noise level σ, the reason of which is to be further studied but one possibility is that both our rivals brings the knowledge of target's physical size into the loop which can eventually tighten the constraints in translation. Nevertheless, our rival's accuracy will possibly be downgraded in real world applications due to the error in measurement of the target size while ours will not.

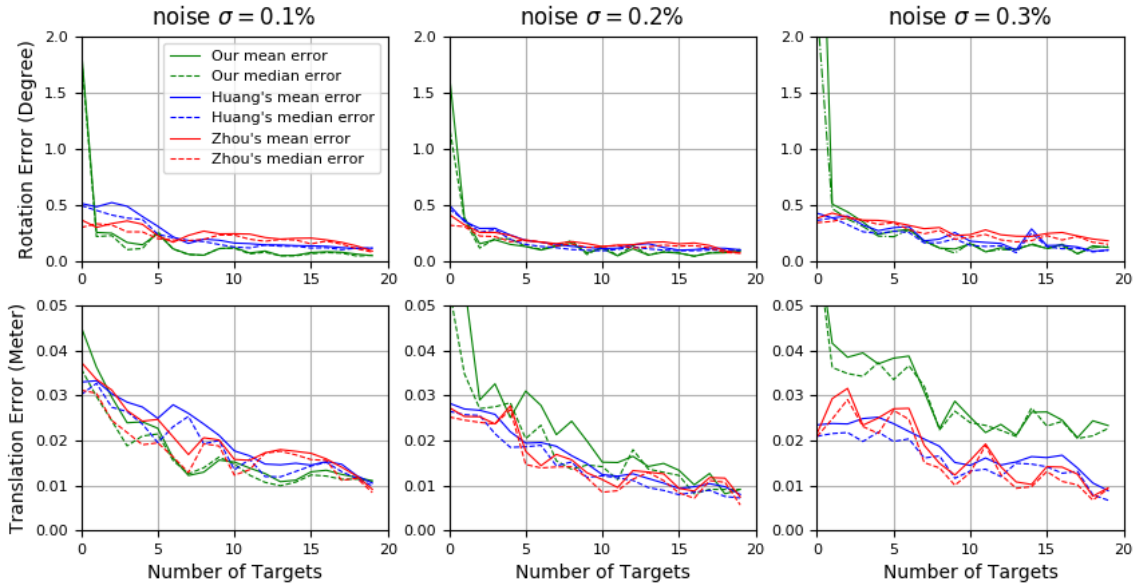
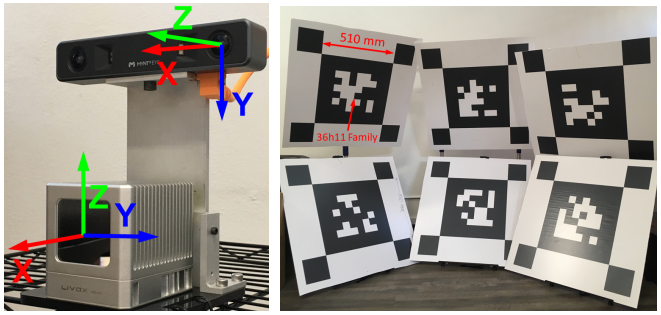


Fig. 5: Comparison of our algorithm with Huang's algorithm and Zhou's algorithm by synthetic tests. Rotation and translation errors are investigated with Guassian noise with deviation of 0.1%, 0.2% and 0.3% w.r.t. LiDAR measurement value.



(a) (b)

Fig. 6: Experiment set-up: (a) shows the coordinate configuration of Livox Mid-40 LiDAR and MYNT EYE D camera. (b) shows the AprilTags calibration targets.

2) Qualitative Results: In this section we present experimental results on real data validating the calibration method presented in Section III. Our experimental setup is composed of a Livox Mid-40 LiDAR and a MYNT EYE D stereo camera, of which only one camera was used for the presented results. The LiDAR and MYNT EYE D camera were rigidly mounted together as seen in Fig. 6a. The camera had been previously calibrated using the Kalibr toolbox [18] and the image stream undistorted before further processing.

The AprilTags from the 36h11 family were printed in poster size ($0.51m \times 0.51m$ for each block and $0.15m$ spacing on each edge) and mounted on foam boards, with a single tag per board. Six such AprilTag boards were assembled and hung in different orientations, as can be seen in Fig 6b.

The Livox Mid-40 LiDAR provides 10^5 unique points per second. A high density cloud with 10^6 points can be accumulated in 10 seconds for our experiment. An example raw LiDAR scan is shown in Fig. 3: the black areas on the AprilTag generate point clouds with very low reflectivity values (lower than 20) that are colored in deep blue. Laser points originating from the white areas off the targets generally have values higher than 40. This information was used to set ϕ_{thr} to 30. All the targets are successfully detected with 4 corners.

To visualize the alignment between image and point cloud, we apply the estimated transformation to the LiDAR point cloud, and project it into the camera image plane. This generates a camera image (Fig. 7a) and a LiDAR virtual image (Fig. 7b) on the same image plane, as they share the coordinate frame after the transformation. The LiDAR image is colored by reflectivity, where higher reflectivity values are assigned a lighter color. To compare the two images Fig. 7c shows sub-blocks of the two images interleaved, highlighting edge and texture alignment.

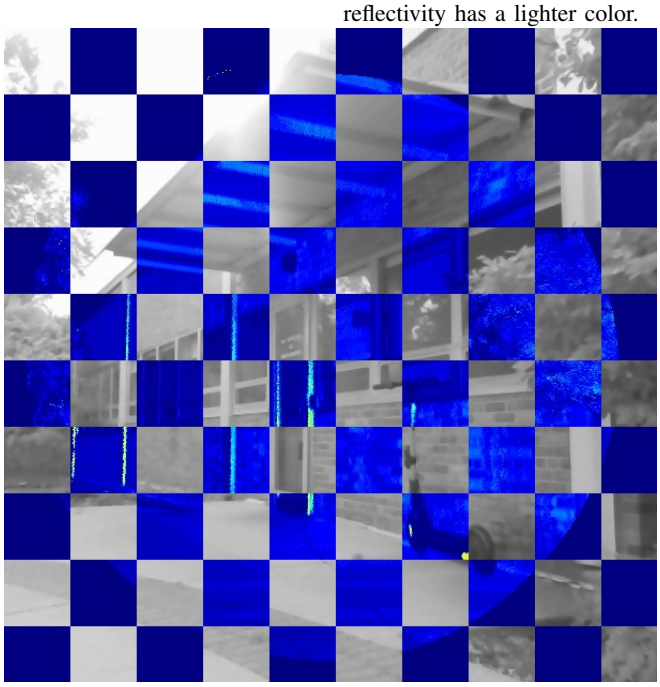
3D alignment are visualized by coloring the point could with the corresponding pixel color, as shown in Fig. 8. Alignment quality can be observed through the proper coloring of the walls, window frames, trees and scooter.

V. CONCLUSIONS

In this paper we have presented a novel approach to automatically calibrate the extrinsics of a LiDAR and camera system using fiducial targets. We rendered a virtual LiDAR binary image using the laserscan reflectivity information. Feature detectors are then applied to detect and identify feature points in the LiDAR image and are then mapped back to 3D space. With 3D feature points in the LiDAR coordinate

(a) Camera image

(b) LiDAR cloud projected onto camera image plane and colored with reflectivity value. Higher reflectivity has a lighter color.



(c) Alternatively tiled camera image and LiDAR image. Adjacent sub-blocks show alignment of texture on building.

Fig. 7: Alignment check of camera image and LiDAR point cloud: point cloud is projected to camera image plane with estimated transformation, and shown in blue-green spectrum

frame and the corresponding 2D points in the camera image paired by tag ID, the calibration problem can be solved using PnP solutions. Synthetic evaluation reveals that our algorithm provides comparable accuracy with other latest approaches. Real world experiments highlight the robustness of the method and its ability to correctly align the LiDAR and camera data without operator intervention.

As our algorithm only extracts 4 corners from each tag, it wastes the space on the calibration target. Our future work include developing a more space-efficient feature which can be shared by our LiDAR with optical cameras which can eventually boost up the efficiency of the whole calibration process.

REFERENCES

[1] *Livox mid series user manual v1.2*. Aug. 2019.

Fig. 8: Point cloud colored with corresponding RGB values to show qualitative results of calibration.

[2] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic extrinsic calibration of vision and lidar by maximizing mutual information," *Journal of Field Robotics*, vol. 32, no. 5, pp. 696–722, 2015.

[3] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 3400–3407.

[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[5] Qilong Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, Sep. 2004, 2301–2306 vol.3.

[6] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-09*, 2005.

[7] G. Pandey, J. McBride, S. Savarese, and R. Eustice, "Extrinsic calibration of a 3d laser scanner and an omnidirectional camera," *IFAC Proceedings Volumes*, vol. 43, no. 16, pp. 336 –341, 2010, 7th IFAC Symposium on Intelligent Autonomous Vehicles.

[8] L. Zhou and Z. Deng, "Extrinsic calibration of a camera and a lidar based on decoupling the rotation from the translation," in *2012 IEEE Intelligent Vehicles Symposium*, IEEE, 2012, pp. 642–648.

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

[10] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS*, Oct. 2018.

[11] D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3d laser range finder from

natural scenes," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 4164–4169.

[12] Z. Pusztai and L. Hajder, "Accurate calibration of lidar-camera systems using ordinary boxes," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 394–402.

[13] J.-K. Huang and J. W. Grizzle, "Improvements to target-based 3d lidar to camera calibration," *ArXiv preprint arXiv:1910.03126*, 2019.

[14] 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*, Oct. 2012, pp. 286–292.

[15] Y. Park, S. Yun, C. Won, K. Cho, K. Um, and S. Sim, "Calibration between color camera and 3d lidar instruments with a polygonal planar board," *Sensors*, vol. 14, no. 3, pp. 5333–5353, 2014.

[16] M. Vel'as, M. Španěl, Z. Materna, and A. Herout, "Calibration of rgb camera with velodyne lidar," 2014.

[17] V. Fremont, S. A. Rodriguez F, and P. Bonnifait, "Circular targets for 3d alignment of video and lidar sensors," *Advanced Robotics*, vol. 26, no. 18, pp. 2087–2113, 2012.

[18] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2013, pp. 473–480.

[19] K. Irie, M. Sugiyama, and M. Tomono, "Target-less camera-lidar extrinsic calibration using a bagged dependence estimator," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, IEEE, 2016, pp. 1340–1347.

[20] C. G. Harris, M. Stephens, *et al.*, "A combined corner and edge detector.," in *Alvey vision conference*, Citeseer, vol. 15, 1988, pp. 10–5244.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[22] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, Springer, 2006, pp. 404–417.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf.," in *ICCV*, Citeseer, vol. 11, 2011, p. 2.

[24] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[25] R. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, Aug. 1987.

[26] D. Eberly, "Rotation representations and performance issues," 2016.