# Calibrating LiDAR and Camera using Semantic Mutual information

Peng Jiang[1], Philip Osteen[2], and Srikanth Saripalli[1]

*Abstract*— We propose an algorithm for automatic, targetless, extrinsic calibration of a LiDAR and camera system using semantic information. We achieve this goal by maximizing mutual information (MI) of semantic information between sensors, leveraging a neural network to estimate semantic mutual information, and matrix exponential for calibration computation. Using kernel-based sampling to sample data from camera measurement based on LiDAR projected points, we formulate the problem as a novel differentiable objective function which supports the use of gradient-based optimization methods. We also introduce an initial calibration method using 2D MI-based image registration. Finally, we demonstrate the robustness of our method and quantitatively analyze the accuracy on a synthetic dataset and also evaluate our algorithm qualitatively on KITTI360 and RELLIS-3D benchmark datasets, showing improvement over recent comparable approaches.

## I. INTRODUCTION

Camera and Light Detection and Ranging (LiDAR) sensors are essential components of autonomous vehicles with complementary properties. A camera can provide high-resolution color information but is sensitive to illumination and lack of spatial data. A LiDAR can provide accurate spatial information at longer ranges and is robust to illumination changes, but its resolution is much lower than the camera and doesn't measure color. Fusing the measurements of these two sensors allows autonomous vehicles to have an improved understanding of the environment. In order to combine the data from different sensor modalities, it's essential to have an accurate transformation between the coordinate systems of the sensors. Therefore, calibration is a crucial first step for multi-modal sensor fusion.

In recent years, many LiDAR-camera calibration methods have been proposed. These methods can be categorized based on whether they are online [1][2][3][4][5][6] or offline [7],[8] [9] as well as whether they require a calibration target [7],[8] or not [4][5][6][10]. Target-based methods require carefully designed targets [11], [12], [9] while targetless methods use data from the natural environment to perform calibration. In this paper, we develop an effective targetless method.

Calibration algorithms can also be categorized into whether they are optimization-based methods or learning-based methods. Most traditional methods are optimization-based methods that try to minimize or maximize a metric [7],[8],[13],[14],[12],][10]. Meanwhile, learning-based methods construct and train neural network models to directly predict the calibration parameters [4],[15],[10],[5]. Both target-based and targetless optimization-based methods must define features for calibration. Common features include edges[7],[8],[13],[16], gradient[14], and semantic information [1], [3], [2]. Among these, semantic information is a higher-level feature that is available from human annotations or learning-based semantic segmentation model predictions [17], [18]. Several methods have been proposed to utilize semantic information to calibrate LiDAR and camera. Nagy et al. [1] use a structure from motion (SfM) pipeline to generate

points from image and register with LiDAR to create basis calibration and refine based on semantic information. Zhu et al. [3] use semantic masks of image and construct height map to encourage laser points to fall on the pixels labeled as obstacles. The methods mentioned above only use semantic information from images and also define other features to help calibration. Wang et al. [2] proposed a new metric to calibrate LiDAR and camera by reducing the distance between the misaligned points and pixels based on semantic information from both image and point cloud.

In this paper, we also use semantic information from LiDAR and camera measurements. We propose to use mutual information as the metric to optimize. Mutual information is widely used in medical image registration [19], [20]. Pandey et al.[21] first proposed to apply it to LiDAR camera calibration. Their approach considers the sensor-measured surface intensities (reflectivity for LiDAR and grayscale intensity for camera) as two random variables and maximizes the mutual information between them. Inspired by [20], we use mutual information neural estimate (MINE) [22] to estimate the mutual information. We also use matrix exponential to compute transformation matrix and use the kernel-based sampler to sample points from images based on projected LiDAR. These treatments allow us to propose a fully differentiable LiDAR camera calibration framework based on semantic information, and implement the algorithm using popular Deep learning libraries. The major contributions of this dataset can be summarized as follows:

- We propose an algorithm for automatic, targetless, extrinsic calibration of a LiDAR and camera system using semantic information.
- We show how to make the objective function fully differentiable and optimize it using the gradient-descent method.
- We introduce an initial calibration method using 2D MI-based image registration.
- We evaluate our method on a synthetic dataset from Carla simulator [23], as well as the real KITTI360 [24] and RELLIS-3D [25] datasets.

## II. METHOLOGY

### A. Algorithm

The following formula describes the transformation relationship of a point $p_i^L$ from the LiDAR local coordinate system to the camera projection plane:

$$\begin{bmatrix} p_i^{uv} \\ 1 \end{bmatrix} = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_i^L \\ 1 \end{bmatrix} \quad (1)$$

- $p_i^L = \begin{bmatrix} x_i^L & y_i^L & z_i^L \end{bmatrix}^T$ represents the coordinate of point $p_i$ in Lidar local frame;
- $t$ represents the $3 \times 1$ translation vector;
- $R$ represents the $3 \times 3$ rotation matrix;
- $K$ represents the $3 \times 3$ intrinsic matrix of camera;
- $p_i^{uv} = \begin{bmatrix} u_i & v_i \end{bmatrix}^T$ represents the coordinate of point $p_i$ in Camera projection plane;

This paper focuses on the extrinsic calibration between LiDAR and camera; therefore, we assume that the camera

[1]J. Mike Walker '66 Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, USA {maskjp, ssaripalli}@tamu.edu
[2]DEVCOM Army Research Laboratory (ARL), Adelphi, MD 20783, USA philip.r.osteen.civ@mail.mil
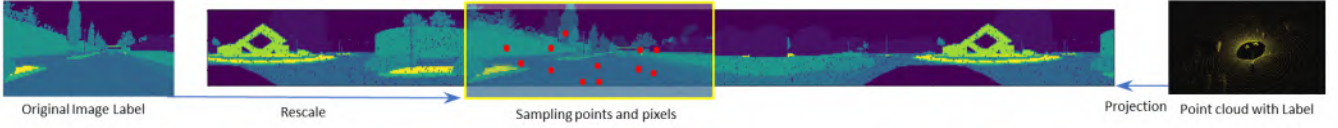
Fig. 1: Initial Calibration Procedure: (a) Project LiDAR label into 2D cylinder plane and zoom the camera image label. (b) Register 2D semantic Images. (c) Sample points and pixels from the overlapped area of the two semantic labels.

---

**Algorithm 1:** 3D Calibration

**input :** Lidar Point Cloud $P$, Point Cloud Labels $L^P$, Image Labels $L^C$, Camera Intrinsic Matrix $K$, Initial Transformation Matrix $T_{init}$

**output:** Transformation Matrix $T$

Use $T_{init}$ to initialize $v_i$;

Use random initialization for MINEnet parameters $\theta$;

Initialize learning rate $\alpha$, $\beta$, optimizer and learning rates scheduler;

**while** *not converge* **do**

$\quad T = \sum_{i=0}^{5} v_i B_i$;

$\quad$ Sample $b$ minibatch points $P_b$ and labels $L_b^P$ from $P$ and $L^P$;

$\quad P_b^{u,v}, = \text{Proj}(P_b, T, K)$;

$\quad \widetilde{L}_b^C = \text{Sample}(L^C, P_b^{u,v})$;

$\quad MI = \text{MINE}(L_b^P, \widetilde{L}_b^C)$;

$\quad$ Update MIME parameter: $\theta + = \alpha \bigtriangledown_\theta MI$;

$\quad$ Update matrix exponential parameters:

$\quad v + = \beta \bigtriangledown_v MI$;

**end**

Return $T = \sum_{i=0}^{5} v_i B_i$;

---

and LiDAR have been intrinsically calibrated. The extrinsic calibration parameters are given by $R$ and $t$ in Eq.1. In addition to point cloud coordinates and image, we also assume the semantic labels of point cloud $L^L$ and image $L^C$ are available.

We consider the semantic label value of each point cloud and its corresponding image pixel as two random variables $X$ and $Y$. The mutual information of the two variable have the maximum value when we have the correct calibration parameters. In order to perform calibration, we need to perform the following three operations:

1) **Transformation Matrix Computation**: $P^{uv} = \text{Proj}(P^L, R, t)$ projects point cloud from lidar coordinate to camera coordinate;

2) **Image Samping**: $\widetilde{L}^C = \text{Sample}(L^C, P^{uv})$ samples semantic label values from image labels based on projected LiDAR coordinates.

3) **Mutual Information Estimation**: $I(X, Y) = \text{MI}(\widetilde{L}^C, \widetilde{L}^L)$ estimate mutual information base the samples from the semantic labels of LiDAR points and its corresponding pixels on image.

Therefore, our full optimization objective function can be written as Eq.2

$$R, t = \arg \max_{R,t} \text{MI}(\text{Sample}(L^C, \text{Proj}(P^L, R, t))) \quad (2)$$

*B. Optimization*

The cost function Eq.2 is maximized at the correct value of the rigid-body transformation parameters between LiDAR and camera. Therefore, any optimization technique that iteratively converges to the global optimum can be used,

although we prefer gradient based optimization methods for their fast convergence properties. Here, we present how to make the cost function fully differentiable, which allows us to optimize Eq.2 with a gradient-based optimization method. In the remaining part of this section, we describe how to make transformation matrix computation, image sampling and mutual information estimation differentiable, with the full algorithm given in Algorithm.1.

*1) Transformation Matrix Computation:* $P^{uv} = \text{Proj}(P^L, R, t)$ involves rigid 3D transformation $T$ which can be represented as the matrix exponential $(T = \exp(H))$. And the matrix $H$ can be parameterized as a sum of weighted basis matrix of the Lie algebra $se(3)$ $(H = \sum_{i=0}^{5} v_i B_i)$ as described in [26]. Therefore, we can represent $T$ with 6 parameters (see Eq.3) and apply standard partial derivative to compute the gradient while optimizing.

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} = \exp\left(\sum_{i=0}^{5} v_i B_i\right) = \sum_{j=0}^{\infty} \frac{(\sum_{i=0}^{5} v_i B_i)^n}{n!} \quad (3)$$

*2) Image Sampling:* Differentiable image sampling is widely used in deep learning for computer vision [27]. The sampling operation can be written as

$$\widetilde{l}_i^C = \sum_h^H \sum_w^W l_{hw}^C k(u_i - h; \Phi_x) k(v_i - w; \Phi_y) \quad (4)$$

where $\Phi_x$ and $\Phi_y$ are the parameters of a generic sampling kernel $k()$ which defines the image interpolation (e.g. bilinear), $l_{hw}^C$ is the label value at location $(h, w)$ of the image semantic label. $\widetilde{l}_i^C$ is the corresponding semantic label of point $p_i$ after projecting on image plane.

*3) Mutual Information Estimation:* Mutual information is a fundamental quantity for measuring the relationship between random variables. Traditional approaches are non-parametric (e.g., binning, likelihood-ratio estimators based on support vector machines, non-parametric kernel-density estimators), which are not differentiable. Several mutual information neural estimators have been proposed [22], [28], [29]. These estimators are consist of neural networks that are fully differentiable. In our implementation, we use MINE[22] to estimate mutual information. This method uses the Donsker-Varadhan (DV) duality to represent MI as

$$\widehat{I(X; Y)}_n = \sup_{\theta \in \Theta} \mathbb{E}_{P_{(X,Y)}}[F_\theta] - \log\left(\mathbb{E}_{P_{(X)}P_{(Y)}}\left[e^{F_\theta}\right]\right) \quad (5)$$

, where $P(X, Y)$ is the joint density for random variables $X$ and $Y$ and $P(X)$ and $P(Z)$ are marginal densities for $X$ and $Y$. $F_\theta$ is a function parameterized by a neural network, and $\theta$ is the parameters of the neural network.

*C. Initial Calibration*

Most targetless calibration approaches require good initialization because good initialization helps the optimization converge faster. Besides, good initialization can help gradient-based optimization methods to avoid some local
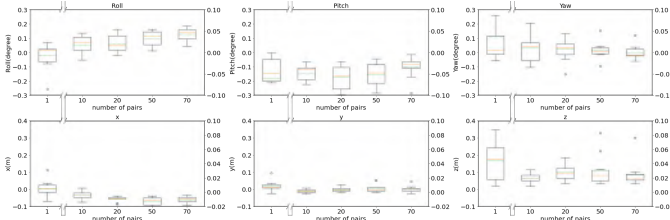
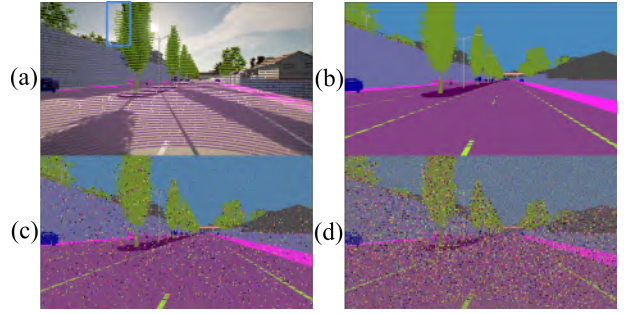Fig. 2: Calibration error using different number of pairs.



Fig. 3: (a) Simulated LiDAR point cloud projected on image; (b) Ground true image label with point cloud; (c) image label and point cloud label with 20% error (d) image label and point cloud label with 50% error.

minimum. By utilizing the semantic information, we convert the initial calibration method into a combination of 2D image registration and Perspective-n-Point (PnP) problem[30]. We project LiDAR points into a spherical 2D plane [31] and get a 2D semantic range image from the LiDAR. We consider the LiDAR to be a low-resolution 360-degree camera with a partially overlapping field of view with the ordinary camera as shown in Fig.1. Therefore, we zoom the semantic label of the camera image into the same resolution as LiDAR. Then, we perform 2D MI-based image registration on the two semantic images. After registration, we can get the raw correspondence between points of LiDAR and pixels of the camera image. Then, we sample several pairs of points and pixels and solve the PnP problem to get an initial calibration between the LiDAR and camera. The full algorithm is described in Algorithm.2

---

**Algorithm 2:** Initial Calibration

**input** : Lidar Point Cloud $P$, Point Cloud Labels $L^P$, Lidar Field of View $FoV_H^L, FoV_V^L$, Lidar Channel Number $H^L$ and Ring Point Number $W^L$ Image Labels $L^I$, Camera Field of View $FoV_H^C, FoV_V^C$, Image Height $H^I$ and Width $W^I$

**output:** Initial Transformation Matrix $T_{init}$

$L_{cy}^P = \text{SphericalProj}(P, L^P, H^L, W^L)$;

$W_z^I, H_z^I = \frac{W^L}{Fov_V^L} FoV_V^C, \frac{H^L}{Fov_H^L} Fov_H^C$;

$L_z^I = \text{Zoom}(L^I, W_z^I, H_z^I)$;

Register $L_z^I$ and $L_{cy}^P$ using 2D MI-based method;

Sample pixels $I_{cy}^P$ and $I_z^I$ from the overlapping between $L_{cy}^P$ and $L_z^I$;

Recover image pixels $I_s^I = \text{DeZoom}(I_z^I, W_z^I, H_z^I, W^I, H^I)$;

Recover points $P_s = \text{DeSphericalProj}(I_{cy}^P, P)$;

$T_{init} = \text{PnPsolver}(P_s, I_s^I)$;

---

## III. EXPERIMENT AND RESULTS

This section describes the experiments to evaluate the accuracy and robustness of the proposed automatic calibration technique. We first evaluate our methods on a synthetic dataset. Then we tested our methods on the real-world KITTI360 [24] and RELLIS-3D [25] datasets.

### A. Synthetic dataset

To test the accuracy and robustness of our methods, we created a dataset including paired LiDAR and camera data with semantic labels using the Carla simulator [23]. The simulator can generate 21 classes. During, experiments we only used 20 classes. The image has a size of $1280 \times 720$. The Lidar has 360 degrees and 64 channels. Each channel includes roughly 800 points per frame. Due to the limitation of simulation, the point cloud and image are completely over-lapped with each other (see Fig.3 (a)). However, the synthetic
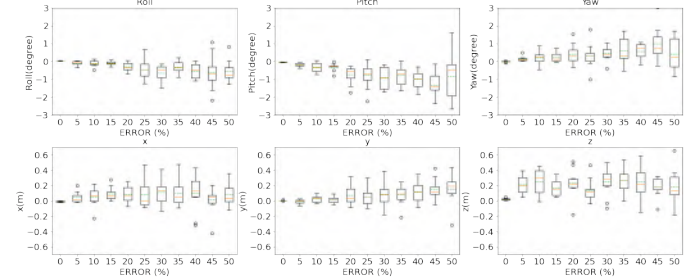


Fig. 4: Calibration error of different noisy labels

data can still provide the most accurate transformation and semantic labels. Another disadvantage of synthetic data is that some real-world sensor and environmental characteristics are not perfectly modeled.

We first tested the performance and robustness of our methods and the effects of the number of data frames on overall performance. In each test, we used the ground truth labels and provided the initial transformation through the methods described in section II-C. Then, we tested the procedure with 1,10,20,50, and 70 pairs of frames. As shown in Fig.2, the variance of the error decreases as we increase the number of pairs we use. Because by introducing more data, the estimator is able to have better mutual information estimation of the overall semantic information. Besides, more data also increase the scene's diversity and reduce the local minimum of the optimization. Secondly, we tested the effects of the error of labels. In each test, we added random noise to both the image label and point cloud label (see Fig.3). The results are shown in Fig.4. As expected, the accuracy decreases, and the variance increases with the error of the labels.

### B. Real-world datasets

Following our experiments with simulated data, we tested on two real-world datasets, KITTI360 and RELLIS-3D, which both provide annotated synchronized LiDAR and

TABLE I: Calibration results of KITTI360

| Methods | Roll(°) | Pitch(°) | Yaw(°) | X(m) | Y(m) | Z(m) |
|---------|---------|----------|--------|------|------|------|
| KITTI360 | 73.72 | -71.37 | 64.56 | 0.26 | -0.11 | -0.83 |
| SOIC | 74.08 | -71.21 | 64.75 | 0.11 | -0.12 | -1.29 |
| PMI | 71.19 | –64.38 | 58.35 | 2.28 | -0.01 | -0.70 |
| Ours | 73.98 | -71.18 | 64.57 | 0.02 | -0.14 | -1.36 |

TABLE II: Calibration results of RELLIS3D

| Methods | Roll(°) | Pitch(°) | Yaw(°) | X(m) | Y(m) | Z(m) |
|---------|---------|----------|--------|------|------|------|
| RELLIS3D | 70.80 | 67.78 | -68.06 | -0.04 | -0.17 | -0.13 |
| SOIC | 70.08 | 68.62 | -66.68 | 0.00 | -0.15 | 4.41 |
| PMI | 75.65 | 69.87 | -64.24 | -0.08 | -0.33 | -0.10 |
| Ours | 70.24 | 67.63 | -67.32 | -0.05 | -0.19 | -0.06 |

Fig. 5: (a) KITT360 calibration; (b) SOIC Results; (c) MI using surface intensities; (d) MI using semantic information.
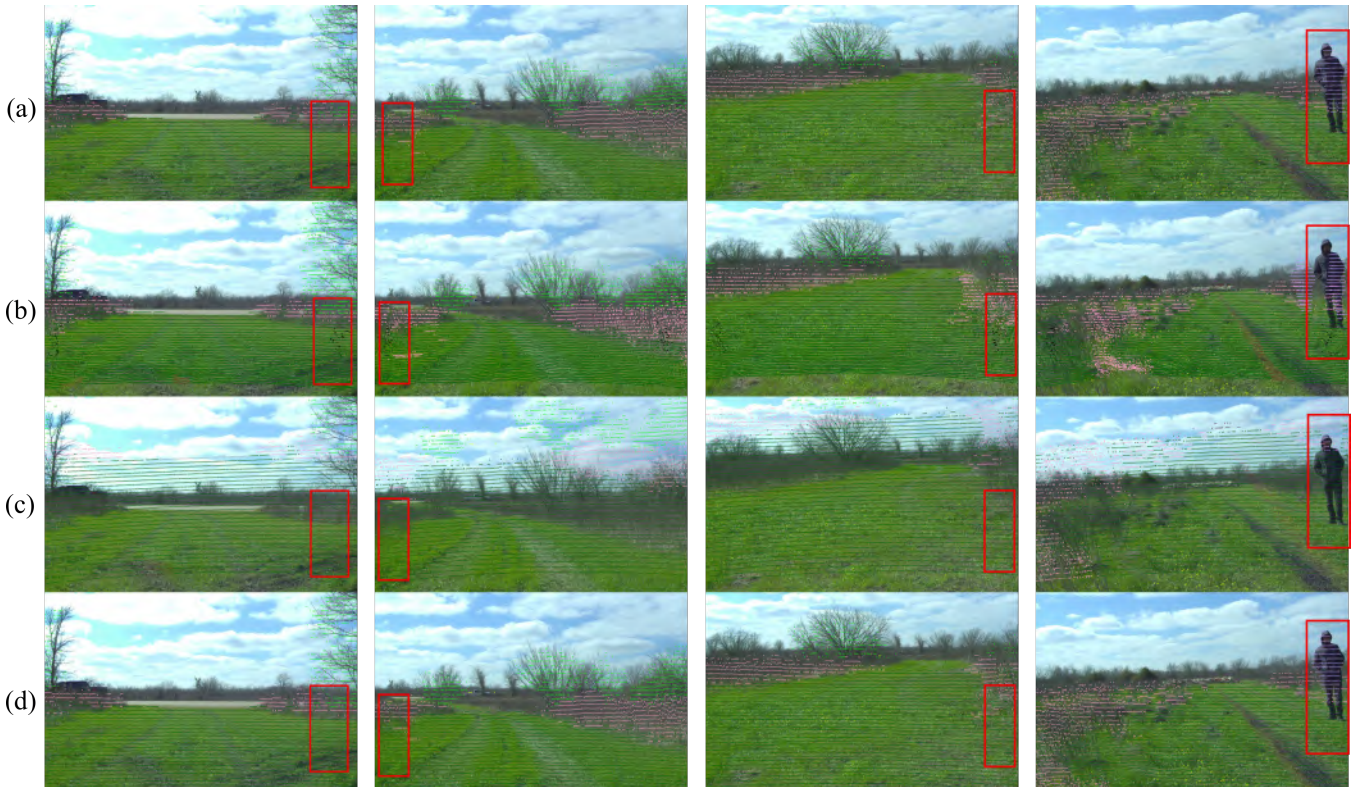


Fig. 6: (a) RELLIS-3D calibration; (b) SOIC Results; (c) MI using surface intensities; (d) MI using semantic information.

image data. Both datasets also provide calibration parameters between LiDAR and camera but they are not accurate as we can see in Fig.5 (a) and Fig.6 (a). Therefore, we apply our methods on the two datasets and qualitatively compared with SOIC[2]. We also follow Pandey's [21] method to use the mutual information between the sensor-measured surface intensities to calibrate the sensors. But we use MINE[22] to estimate the MI.The results is noted as **PMI** in TableI and II.All methods were initialized by our proposed method. As shown in Table. I-II, our method provide the closest results with the provided parameters of each dataset. Meanwhile, Fig. 5 and 6 shows that our calibration results better cross-sensor data fusion than the other two methods. More visual results can be found in the video[1].

## IV. SUMMARY AND FUTURE WORK

This paper presented a fully differential LiDAR-Camera calibration method using semantic information from both sensor measurements. By utilizing semantic information, the method doesn't need specific targets and initial guess parameters. Because the method is fully differentiable, it can be implemented using the popular deep learning framework, such as Pytorch[32] or Tensorflow[33]. Moreover, mutual semantic information was introduced used to register multi-modal data. The method has the potential to leverage deep features to calibrate LiDAR and camera. Another possible application of this method is to use this framework in deep fusion directly. By embedding this framework in the deep fusion framework, rough calibration between sensors might be enough.

# REFERENCES

[1] B. Nagy, L. Kovacs, and C. Benedek, "SFM and Semantic Information Based Online Targetless Camera-LIDAR Self-Calibration," in *Proceedings - International Conference on Image Processing, ICIP*, vol. 2019-Septe. IEEE Computer Society, sep 2019, pp. 1317–1321.

[2] W. Wang, S. Nobuhara, R. Nakamura, and K. Sakurada, "Soic: Semantic online initialization and calibration for lidar and camera," mar 2020. [Online]. Available: http://arxiv.org/abs/2003.04260

[3] Y. Zhu, C. Li, and Y. Zhang, "Online Camera-LiDAR Calibration with Sensor Semantic Information," in *Proceedings - IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., may 2020, pp. 4970–4976.

[4] X. Lv, B. Wang, D. Ye, and S. Wang, "Lidar and Camera Self-Calibration using CostVolume Network," dec 2020. [Online]. Available: http://arxiv.org/abs/2012.13901

[5] K. Yuan, Z. Guo, and Z. Jane Wang, "RGGNet: Tolerance Aware LiDAR-Camera Online Calibration with Geometric Deep Learning and Generative Model," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6956–6963, oct 2020.

[6] H. J. Chien, R. Klette, N. Schneider, and U. Franke, "Visual odometry driven online calibration for monocular LiDAR-camera systems," in *Proceedings - International Conference on Pattern Recognition*, vol. 0. Institute of Electrical and Electronics Engineers Inc., jan 2016, pp. 2848–2853.

[7] S. Mishra, G. Pandey, and S. Saripalli, "Extrinsic Calibration of a 3D-LIDAR and a Camera," in *IEEE Intelligent Vehicles Symposium, Proceedings*. Institute of Electrical and Electronics Engineers Inc., 2020, pp. 1765–1770.

[8] S. Mishra, P. R. Osteen, G. Pandey, and S. Saripalli, "Experimental evaluation of 3D-LIDAR camera extrinsic calibration," in *IEEE International Conference on Intelligent Robots and Systems*. Institute of Electrical and Electronics Engineers Inc., oct 2020, pp. 9020–9026.

[9] J. L. Owens, P. R. Osteen, and K. Daniilidis, "MSG-cal: Multi-sensor graph-based calibration," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem. Institute of Electrical and Electronics Engineers Inc., dec 2015, pp. 3660–3667.

[10] G. Zhao, J. Hu, S. You, and C. C. J. Kuo, "CalibDNN: Multimodal Sensor Calibration for Perception Using Deep Neural Networks," mar 2021. [Online]. Available: http://arxiv.org/abs/2103.14793

[11] Q. 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)*, vol. 3, 2004, pp. 2301–2306.

[12] Z. Pusztai, "Accurate Calibration of LiDAR-Camera Systems using Ordinary Boxes," Tech. Rep., 2017.

[13] J. Kang and N. L. Doh, "Automatic targetless camera–LIDAR calibration by aligning edge with Gaussian mixture model," *Journal of Field Robotics*, vol. 37, no. 1, pp. 158–179, jan 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21893

[14] Z. Taylor, J. Nieto, and D. Johnson, "Multi-Modal Sensor Calibration Using a Gradient Orientation Measure," *Journal of Field Robotics*, vol. 32, no. 5, pp. 675–695, aug 2015. [Online]. Available: http://doi.wiley.com/10.1002/rob.21523

[15] N. Schneider, F. Piewak, C. Stiller, and U. Franke, "RegNet: Multimodal sensor registration using deep neural networks," in *IEEE Intelligent Vehicles Symposium, Proceedings*. IEEE, jun 2017, pp. 1803–1810. [Online]. Available: http://ieeexplore.ieee.org/document/7995968/

[16] G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna, "CalibNet: Geometrically Supervised Extrinsic Calibration using 3D Spatial Transformer Networks," in *IEEE International Conference on Intelligent Robots and Systems*. Institute of Electrical and Electronics Engineers Inc., dec 2018, pp. 1110–1117.

[17] J. Zhang, X. Zhao, Z. Chen, and Z. Lu, "A Review of Deep Learning-Based Semantic Segmentation for Point Cloud," pp. 179 118–179 133, 2019.

[18] H. Yu, Z. Yang, L. Tan, Y. Wang, W. Sun, M. Sun, and Y. Tang, "Methods and datasets on semantic segmentation: A review," *Neurocomputing*, vol. 304, pp. 82–103, aug 2018.

[19] F. P. Oliveira and J. M. R. Tavares, "Medical image registration: A review," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 17, no. 2, pp. 73–93, jan 2014. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/10255842.2012.670855

[20] A. Nan, M. Tennant, U. Rubin, and N. Ray, "DRMIME: Differentiable mutual information and matrix exponential for multi-resolution image registration," pp. 527–543, jan 2020. [Online]. Available: http://arxiv.org/abs/2001.09865

[21] 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, aug 2015. [Online]. Available: http://doi.wiley.com/10.1002/rob.21542

[22] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm, "Mutual information neural estimation," in *35th International Conference on Machine Learning, ICML 2018*, vol. 2, jan 2018, pp. 864–873. [Online]. Available: http://arxiv.org/abs/1801.04062

[23] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[24] J. Xie, M. Kiefel, M.-T. Sun, and A. Geiger, "Semantic instance annotation of street scenes by 3d to 2d label transfer," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[25] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "Rellis-3d dataset: Data, benchmarks and analysis," 2020.

[26] C. Wachinger and N. Navab, "Simultaneous registration of multiple images: Similarity metrics and efficient optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1221–1233, 2013.

[27] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial Transformer Networks," *Advances in Neural Information Processing Systems*, vol. 2015-January, pp. 2017–2025, jun 2015. [Online]. Available: http://arxiv.org/abs/1506.02025

[28] S. Mukherjee, H. Asnani, and S. Kannan, "CCMI : Classifier based conditional mutual information estimation," 2019.

[29] A. K. Mondal, A. P. Prathosh, A. Bhattacharjee, S. Kannan, S. Mukherjee, and H. Asnani, "C-MI-GAN: Estimation of conditional mutual information using minmax formulation," in *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence, UAI 2020*, 2020, pp. 869–878.

[30] 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. 155–166, feb 2009. [Online]. Available: https://link.springer.com/article/10.1007/s11263-008-0152-6

[31] P. Jiang and S. Saripalli, "LiDARNet: A Boundary-Aware Domain Adaptation Model for Point Cloud Semantic Segmentation," *arXiv*, mar 2020. [Online]. Available: http://arxiv.org/abs/2003.01174

[32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," 2019.

[33] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016*, 2016, pp. 265–283. [Online]. Available: https://tensorflow.org.