2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Online Camera-LiDAR Calibration with Sensor Semantic Information

Yufeng Zhu[1], Chenghui Li[1,2] and Yubo Zhang[1]

***Abstract*— As a crucial step of sensor data fusion, sensor calibration plays a vital role in many cutting-edge machine vision applications, such as autonomous vehicles and AR/VR. Existing techniques either require quite amount of manual work and complex settings, or are unrobust and prone to produce suboptimal results. In this paper, we investigate the extrinsic calibration of an RGB camera and a light detection and ranging (LiDAR) sensor, which are two of the most widely used sensors in autonomous vehicles for perceiving the outdoor environment. Specifically, we introduce an online calibration technique that automatically computes the optimal rigid motion transformation between the aforementioned two sensors and maximizes their mutual information of perceived data, without the need of tuning environment settings. By formulating the calibration as an optimization problem with a novel calibration quality metric based on semantic features, we successfully and robustly align pairs of temporally synchronized camera and LiDAR frames in real time. Demonstrated on several autonomous driving tasks, our method outperforms state-of-the-art edge feature based auto-calibration approaches in terms of robustness and accuracy.**

I. INTRODUCTION

Autonomous driving has attracted an increasing amount of attention from both academia and industrial partners in recent years. A safe and robust navigation system heavily relies on accurate perception of visual objects in the environment. Unfortunately, none of the existing sensors is able to guarantee the perceptual reliability in all cases. To overcome such hardware limitations, recent autonomous robot systems utilize multiple sensor modalities, which provide complementary environmental information, and adopt sensor fusion techniques to reduce data uncertainty. An essential step to fuse information from heterogeneous devices is to accurately estimate their relative rigid body transformations through extrinsic calibration. In this paper, we focus on calibration between a camera and a LiDAR, which is one of the most effective complementary pairs of sensors for robotic perception and is pervasive in modern autonomous vehicles. Unlike calibration between two sensors of the same modality, calibrating multi-modal sensor data is more difficult because we must identify correspondences among completely different sensor data, such as point cloud and image frames. Tedious manual work and special assumptions, such as artificial markers, are often required to overcome such difficulty. This procedure is laborious and time consuming especially for autonomous vehicles, because their sensor position slightly drifts over the time and needs periodic recalibration.

Recent approaches for extrinsic calibration between camera and LiDAR have focused on automatic and targetless methods to reduce the setup cost or complexity and enable online recalibration. While significant effort has been devoted to aligning edge features in different sensors data, we show that aligning semantic features instead can be more effective and robust. In this work, we present an online calibration method tailored to automotive sensor setups as shown in Figure 1. The performance and quality of our approach is demonstrated on real-time automotive sensor calibration tasks, where we have observed significant improvements in accuracy and robustness over existing approaches.

Fig. 1. Our sensors setup is composed of multiple RGB camera and LiDAR devices mounted to the roof of autonomous vehicle. In this work, we focus on calibration problem of single camera-LiDAR pair and our algorithm can be directly generalized to multiple pairs setting.

II. RELATED WORK

Recent years have witnessed the application of camera-LiDAR based sensor fusion to a growing repertoire of autonomous vehicles. To enable reliable robotic perception, there has been a number of proposed solutions to this multi-sensor calibration problem over the past few years. Each sensor can be characterized by its intrinsic parameters, such as camera lens distortion, and extrinsic parameters. In this work, we only focus on extrinsic calibration and assume their intrinsic parameters have been accurately estimated. Extrinsic calibration is the process of estimating the rigid body transformation between two or more sensors. With a proper calibration between a camera and a LiDAR, laser measurements of the LiDAR can be associated with color pixels by being projected onto the camera frame. Conversely, pixels in the camera frame can be given depth values by querying the nearest laser returns. In general, most calibration approaches, including our proposed approach, are fundamentally based on identifying and matching features detected in LiDAR frames and camera frames to determine the calibration parameters. Traditional calibration techniques are realized by placing fixed markers or targets, e.g., a checkerboard, in the scenes, but these approaches suffer from complicated setup requirement and are limited to offline usages. To overcome this limitation, more recent work ex-

[1]Pony.ai
[2]Carnegie Mellon University

plores automatic calibration using features presented in the observed scene, without any preset targets.

A. Target Based Approach

The extrinsic calibration of a color camera and a laser rangefinder was first addressed by Zhang and Pless [1], who used a fixed checkerboard as the calibration target. They solved for the calibration parameters by forming a non-linear optimization problem through normals of the checkerboard surface. This method was then generalized by Unnikrishnan and Hebert [2] who manually selected point features from both sensors data and modeled the calibration task as a linear least-squares problem. Shortly after, Scaramuzza et al. [3] proposed to find the optimal calibration estimation through perspective-from-n-points (PnP) algorithm given manually established point feature correspondences. Nunnez et al. [4] later modified Zhang's method to calibrate the two sensors by detecting a checkerboard pattern. Other calibration targets have also been explored to fulfill this task, including right-angled triangular checkerboard [5], trirectangular trihedron [6], ordinary box [7], custom-made planar target [8], [9], v-shaped calibration target [10] or an arbitrary trihedron [11], etc. Kassir and Peynot [12] eliminated the effort of manual work during calibration by providing a reliable corner detection procedure based on Bouguet's camera calibration Matlab toolbox. Bok et al. [13] used bridging sensor for calibration without overlap between camera and LiDARs' field of view. Geiger [14] presented a calibration toolbox with web interface for multiple cameras and a multi-beam laser using a single shot of multiple checkerboard patterns placed across a room.

B. Targetless Approach

Many attempts have been made in recent years to develop an automatic and flexible camera-LiDAR calibration system without any preset targets. Some interesting and seminal work in this area were published in 2012 by two groups of researchers [15], [16], who proposed to estimate extrinsic calibration parameters by maximizing the mutual information between the reflectivity of point clouds and the intensity of images. This idea was first applied to registration problems given the assumption that data acquired by multiple sensors on the same object should have correlation [17], [18]. For example, reflectivity of LiDAR measurements (the intensity of laser return) tends to be higher on white objects and vegetation, and lower on dark-colored objects. This implies considerable correlation between reflectivity in LiDAR frames and color in camera frames. However, Pandey's method [16] is easily stuck in local optima due to the non-smoothness of objective function. This problem is later addressed by Irie et al. [19] who developed a bagged least-squares mutual information method that enables them to incorporate more features to construct a considerably smoother objective function than previous ones.

Another early attempt was made by Bileschi [20] who proposed to align LiDAR frames to camera images by contour matching. The edge points in each laser scan are identified and projected onto the camera frame. The extrinsic parameters are then adjusted accordingly to improve the alignment of projected edge points to object contours detected in camera frames. A similar idea was proposed and verified later in Levinson and Thrun's work [21]. An objective function is defined to capture the correlation of discontinuities in point clouds and edges in camera frames. An inverse distance transform (IDT) is applied on the edge camera frame to produce a smoother energy map. While they employed only the strength of the edges, Taylor et al. [22] reported the usefulness of the orientation of edges. They proposed using gradient orientation measurement that can evaluate the degree to which edge orientations are aligned between a camera frame and LiDAR reflectivity image. To overcome the difficulty of making direct edge alignment between data acquired by different sensors with significantly different sampling pattern, Castorena et al. [23] jointly fused the data and estimated the calibration parameters. In this work, we propose to align semantic features instead of edge features to improve online calibration robustness, especially for low-resolution LiDAR and noisy inputs.

C. Semantic Image Segmentation

Semantic image segmentation is a well established research area and has been evolved successfully for decades. As we utilize segmentation techniques in our method, here we will briefly introduce most related previous work. For interested readers on this topic, we refer you to some recent survey [24] for a more comprehensive overview. Semantic image segmentation predicts dense labels for each pixel in the image, and is regarded as a very important task that can help deep understanding of scenes, objects, and humans. Traditional methods [25] adopt handcrafted features, while recent convolutional neural networks (CNN) based methods largely improve the performance and make remarkable progress [26]–[28]. In this work, we adopt Pyramid Scene Parsing Network (PSPNet) [27] to semantically segment each camera frame and use it to construct an optimization objective for optimal calibration parameters estimation.

III. Calibration Quality Metric

The LiDAR device produces point clouds which are distance measurements defined in the local coordinate system around the LiDAR. To fuse information from two different device sources, correlation between sensors data has to be established through mappings from one device to the other. For example, a 3D laser point can be represented as a 3×1 vector, $\mathbf{p}_L \in \mathbb{R}^3$, within the LiDAR coordinate system, which can be transformed to the camera coordinate system as $\mathbf{p}_C$ via rigid motion,

$$\mathbf{p}_C = \mathbf{R}\mathbf{p}_L + \mathbf{t}. \tag{1}$$

where $\mathbf{R} \in \mathbb{R}^{3\times3}$ and $\mathbf{t} \in \mathbb{R}^3$ are the relative rotation and translation between two device coordinate system, which are to be figured out in extrinsic calibration. Similar to most previous work, we adopt the pinhole camera model and project the camera coordinates $\mathbf{p}_C$ to image coordinates

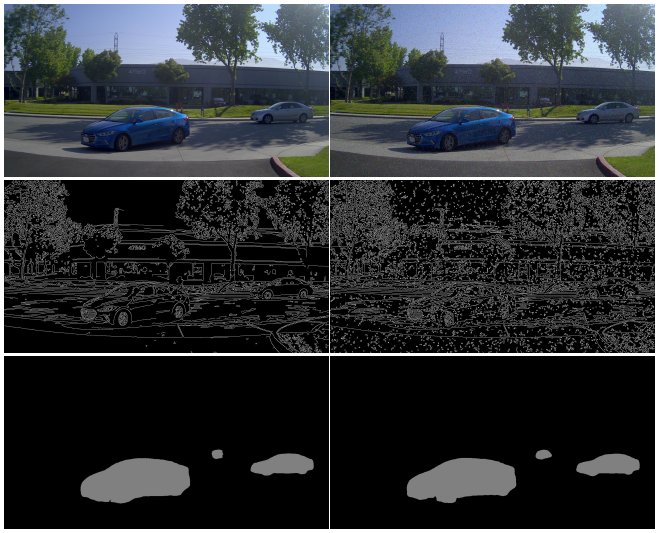

Fig. 2. Edge detection is more sensitive to color variations and noise in camera image than semantic segmentation. In **left** column, we show the edge detection and semantic segmentations for cars in a camera frame. Edge detection results in more redundant and inconsistent features, e.g., on the boundary of shadow on the ground, car windows, caused by local changes in color. In **right** column, we add random noise to the input image, which has a significant impact on the edge detection output. Semantic segmentation provides consistent results across two different noise levels.

$\mathbf{p}_I \in \mathbb{R}^2$. In reality, there is often a considerable amount of radial and tangential lens distortion in camera frames, which makes this projection a nonlinear mapping $\mathcal{K} : \mathbb{R}^3 \to \mathbb{R}^2$,

$$\mathbf{p}_I = \mathcal{K}(\mathbf{p}_C). \tag{2}$$

In this work, we focus on solving extrinsic calibration parameters, $\mathbf{R}$ and $\mathbf{t}$, assuming that $\mathcal{K}(\cdot)$ has already been determined in intrinsic calibration.

We propose a quality metric for extrinsic calibration, which has higher value for more accurate calibration parameters. With this metric, the process of solving extrinsic parameters can be modeled as a non-linear iterative optimization process. One of the popular quality metrics explored and adopted by existing online calibration approaches is based on aligning edge features [21]. These approaches find the optimal parameters by seeking for the best alignment of edge features between point clouds and camera images, which is demonstrated to work well for automatic extrinsic calibration but lack robustness especially when the parameters' initial value is far from optimal. One important cause for this problem is that such metrics highly rely on pixel intensity in camera frames. For example, variations of objects' color in the camera frame may have significant impact on the edge detection results, as shown in Figure 2. Moreover, edge features are usually sensitive to sensor data quality. Noisy or sparse information may easily lead to problematic edge detection results, as shown in Figure 3. In order to avoid the aforementioned problems and improve calibration robustness, we propose a novel quality metric based on semantic features detected in sensor data and develop an iterative nonlinear subgradient solver to efficiently estimate the optimal results.

Fig. 3. Edge feature (**right**) in LiDAR point cloud is vulnerable to noise interference (notice the detected edge feature in red on the ground plane), especially for low resolution laser data as shown in **left** figure. In such case, the detected edge feature is too sparse to identify outliers (false positive) when being aligned with camera image.

In our algorithm, the semantic features of interest are obstacles that can reflect laser beams, e.g., cars, trees, pedestrians, and traffic signs. In recent years, huge progress has been made in the field of image semantic segmentation since the successful application of CNN to image detection tasks [29]. We adopt PSPNet [27] to semantically segment camera frames and consider the segmentation result as a reward mask to guide where the laser points reported by LiDAR device are most likely to fall on when projected onto the camera frame. Unlike edge feature based methods that rely on color information in camera frames, semantic segmentation is more robust to variations of lighting conditions and noise, as demonstrated in Figure 2. In this paper, we use cars as our major features to demonstrate the robustness of our calibration method, as they commonly appear as obstacles in road tests for autonomous vehicles. Other types of semantic features can fit into our proposed framework as long as they are supported by the adopted segmentation algorithms.

With the pixel-wise obstacle/background segmentation produced by PSPNet, we construct a height map $\mathcal{H} : \mathbb{R}^2 \to \mathbb{R}$,

$$h = \mathcal{H}(\mathbf{p}_I), \tag{3}$$

To encourage laser points to fall on the pixels labeled as obstacles, we define our quality metric by measuring the overlap between the height map and the laser points projected onto the camera frame. For LiDAR point cloud data $\mathbb{P} \subset \mathbb{R}^3$, we simply remove ground points $\mathbb{G} \subset \mathbb{R}^3$, as they don't necessarily contribute to the calibration optimization process. Instead of extracting edge features from point cloud based on depth discontinuities [21], we use the point cloud itself. Each point, $\mathbf{p}_L \in \mathbb{P} \setminus \mathbb{G}$, will contribute height value of the pixel it falls on to the final quality metric,

$$\phi(\mathbf{R}, \mathbf{t}) = \sum_{\mathbf{p}_L \in \mathbb{P} \setminus \mathbb{G}} \mathcal{H} \circ \mathcal{K}(\mathbf{R}\mathbf{p}_L + \mathbf{t}). \tag{4}$$

Such strategy is more robust when working with relatively low resolution LiDAR device and noise interference as illustrated in Figure 3. Ideally, when the extrinsic calibration is exact, we should expect the objective ϕ to be maximized as pointed out by the work of mutual information maximization [15], [16] that reflected laser points are more likely to fall on obstacle pixels when projected into the camera image space.

As shown in Figure 4b, the most simple height map is a binary segmentation mask, in which obstacle pixels $\mathbb{O} \subset \mathbb{R}^2$

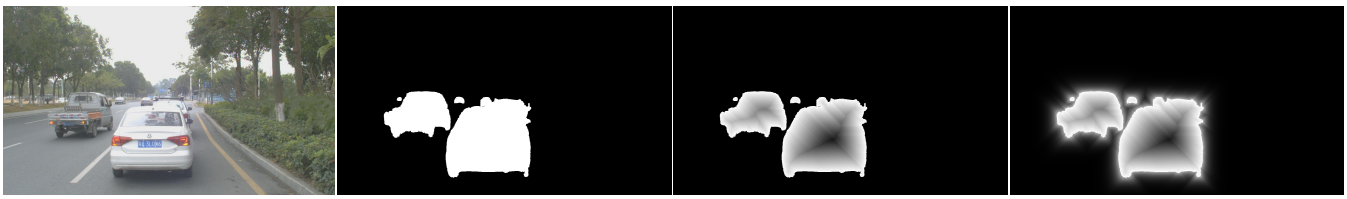

Fig. 4. To avoid influence over quality metric from color variation and noise perturbation, we construct a height map $\mathcal{H}$ from input camera image (**a**). First, we extract the semantic information from RGB image data and set every pixel's value in the segmented region to 1 and others to 0 (**b**). Then we apply distance transform to decay segmented region from boundary to its interior (**c**). Finally, we compute inverse distance transformation of the background region to construct a smooth height map (**d**).

and background pixels $\mathbb{B} \subset \mathbb{R}^2$ are set to one and zero, respectively.

$$h_{SS} = \mathcal{H}_{SS}(\mathbf{p}_I) = \begin{cases} 0, & \mathbf{p}_I \in \mathbb{B} \\ 1, & \mathbf{p}_I \in \mathbb{O} \end{cases}. \tag{5}$$

However, this binary map $\mathcal{H}_{SS}(\cdot)$ has a null space along the camera viewing axis as shown in Figure 5. Namely, any perturbation along that direction will produce local optimal solutions. To enforce uniqueness of local optimality, we decay each segmentation mask from its boundary gradually to its interior as shown in Figure 4c, which stops the extrinsic parameters from perturbing along the viewing axis freely. The interior decaying operation is achieved by distance transformation,

$$\begin{aligned} h_{DT} &= \mathcal{H}_{DT}(\mathbf{p}_I) = \alpha_1 \mathcal{H}_{SS}(\mathbf{p}_I) \\ &+(1-\alpha_1) \max_{\mathbf{q}_I \in \mathbb{B}} \mathcal{H}_{SS}(\mathbf{q}_I) \gamma_1^{\|\mathbf{p}_I - \mathbf{q}_I\|_1}, \mathbf{p}_I \in \mathbb{O}, \end{aligned} \tag{6}$$

where α_1 and γ_1 are set as 0.93 and 0.59 respectively. When LiDAR points fall on background region, they will suffer from the vanishing gradient problem [30] and no longer contribute to improve the calibration quality. Thus we follow a similar objective function smoothing strategy described by Levinson and Thrun [21] using inverse distance transform,

$$\begin{aligned} h_{IDT} &= \mathcal{H}_{IDT}(\mathbf{p}_I) = \alpha_0 \mathcal{H}_{SS}(\mathbf{p}_I) \\ &+(1-\alpha_0) \max_{\mathbf{q}_I \in \mathbb{O}} \mathcal{H}_{SS}(\mathbf{q}_I) \gamma_0^{\|\mathbf{p}_I - \mathbf{q}_I\|_1}, \mathbf{p}_I \in \mathbb{B}, \end{aligned} \tag{7}$$

in order to make the quality metric more friendly to optimization solvers (see Figure 4d). Such smoothed height map will be easier to handle and also share the same optimal calibration solution as the non-smoothed one. Finally, the height map in Equation 3 is defined as

$$h = \mathcal{H}(\mathbf{p}_I) = \begin{cases} h_{IDT}, & \mathbf{p}_I \in \mathbb{B} \\ h_{DT}, & \mathbf{p}_I \in \mathbb{O} \end{cases}. \tag{8}$$

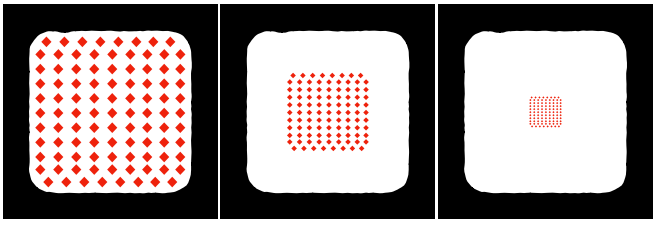

Fig. 5. Binary map has null space problem as the quality metric will be indistinguishable when same amount of LiDAR points fall on the segmented region. The above three configurations differ with each other along the viewing direction, but they have the same calibration quality using binary map. Thus it's impossible to judge which one is the best (**left** in this case).

IV. Calibration Optimization

The objective function introduced in the previous section provides one way to measure the quality of current extrinsic calibration. An accurate calibration configuration should be a stationary point and achieve local optimality of this calibration objective,

$$\mathbf{R}^*, \mathbf{t}^* = \underset{\mathbf{R},\mathbf{t}}{\operatorname{argmax}}\, \phi(\mathbf{R}, \mathbf{t}). \tag{9}$$

Any perturbation of this configuration would lead to decrement of the quality metric. To automatically calibrate between camera and LiDAR, we propose a nonlinear and non-smooth optimization solver to efficiently improve the calibration quality starting from a reasonable initial configuration $\mathbf{R}_0$ and $\mathbf{t}_0$. As 3D rotation matrix $\mathbf{R}$ lies in the Lie group $SO(3)$, it needs to satisfy the constraints,

$$\mathbf{R}^T\mathbf{R} = \mathbf{I}, \quad \det(\mathbf{R}) = 1 \tag{10}$$

where $\mathbf{I}$ is the 3×3 identity matrix and $\det(\cdot)$ is matrix determinant. However, directly optimizing over $\mathbf{R}$ would be too cumbersome as it not only has too many redundant degrees of freedom (9 for matrix of $\mathbb{R}^{3\times3}$) but also introduce non-trivial equality constraints to our optimization problem. Instead we reparameterize the rotation matrix as a rotation vector so that we can get rid of equality constraints and only work on a relatively easy unconstrained optimization problem. A rotation vector $\mathbf{r}$ is simply a vector in $\mathbb{R}^3$, whose unit direction $\frac{\mathbf{r}}{\|\mathbf{r}\|}$ is the rotation axis and length $\|r\|$ is the rotation speed around the axis. As rotation vector space is isomorphic to the Lie algebra $\mathfrak{so}(3)$ of 3D rotation group, conversion between $\mathbf{R}$ and $\mathbf{r}$ can be defined by corresponding exponential and logarithm map. We adopt the Rodrigues formula, $\mathbf{R} = \exp(\mathbf{r})$, to rephrase our optimization problem in terms of $\mathbf{r}$ and $\mathbf{t}$,

$$\Phi(\mathbf{r}, \mathbf{t}) = \phi(\exp(\mathbf{r}), \mathbf{t}). \tag{11}$$

In order to solve for these six degrees of freedom, we propose a new non-monotonic subgradient ascent algorithm as shown in Algorithm 1.

Algorithm 1: Extrinsic Calibration Optimization Solver

Input: $\mathbf{r}_0$, $\mathbf{t}_0$, $\mathcal{H}$, $\mathcal{K}$, $\mathbb{P} \setminus \mathbb{G}$, ω, τ, δ
Output: $\mathbf{r}^*$, $\mathbf{t}^*$
$\mathbb{E}_0 = \{\Phi_0\}$, $\beta = 0.5$
for n = 0 **to** τ
  $\mathbb{E}_n = \mathbb{E}_n \setminus \{\Phi_{n-\omega}\}$
  $\eta = 1.0$, $\Delta\mathbf{r}_n \in \partial_{\mathbf{r}_n}\Phi$, $\Delta\mathbf{t}_n \in \partial_{\mathbf{t}_n}\Phi$
  iter = 0, $back\_tracking$ = **true**
  while $back\_tracking$ **and** *iter* $< \delta$ **do**
    if $\Phi(\mathbf{r}_n + \eta\Delta\mathbf{r}_n, \mathbf{t}_n + \eta\Delta\mathbf{t}_n) > \min\mathbb{E}_n$
      $\mathbf{r}_{n+1} = \mathbf{r}_n + \eta\Delta\mathbf{r}_n$
      $\mathbf{t}_{n+1} = \mathbf{t}_n + \eta\Delta\mathbf{t}_n$
      $back\_tracking =$ **false**
    else
      $\eta = \beta \cdot \eta$
    end if
    iter $=$ *iter* $+ 1$
  end while
  $\mathbb{E}_{n+1} = \mathbb{E}_n \cup \{\Phi_{n+1}\}$
  if $\min\mathbb{E} == \max\mathbb{E}$
    break
  end if
end for
$\mathbf{r}^* \approx \mathbf{r}_n$, $\mathbf{t}^* \approx \mathbf{t}_n$

The inputs to our solver are camera frames and LiDAR frames as well as initial value for calibration parameters, $\mathbf{r}_0$ and $\mathbf{t}_0$. After processing sensors data as discussed in the previous section, we start an iterative procedure to gradually improve the calibration parameters. Unlikely convex unconstrained optimization problem, in which objective gradient plays a crucial role to update solution iteratively, our objective function is neither convex nor differentiable because height map is sampled discretely. Thus we have to adopt the subdifferential strategy by picking one direction within the subdifferential cone $\partial\Phi$. We choose coordinate ascent direction to update $\mathbf{r}$ and $\mathbf{t}$ during each iteration. Then the step size η is determined through backtracking line search in order to stabilize the optimization process. Due to lack of objective differential information, we only compare objective value during line search. An immediate idea would be picking the step size whenever the objective value is larger than the current one as we backtrack along the chosen search direction. Such approach will enforce the solver to converge as the objective value is guaranteed to increase monotonically. However, we observed that such monotonic line search strategy may sometimes lead us to suboptimal solution. This is due to the non-differentiable and non-convex properties of our objective function. To alleviate such problem, we choose non-monotonic line search [31] by comparing the objective along search direction against the minimum objective value among a sequence of past iterations, $\mathbb{E}_n$, instead of just the current, Φ_n. Such modification will guarantee our solver to converge but also provide it the ability to recover from being stuck by the suboptimal solution. The cardinality of set $\mathbb{E}_n$ is decided by user provided parameter ω.

V. EXPERIMENTS

A. Implementation

All evaluations and experiments are performed on an Intel Xeon Gold 6148 CPU and a NVIDIA TITAN V GPU with our C++ and CUDA implementation. Our testing data comes from recordings of real world self-driving road tests. The input sensor data are a 960×510 downsampled RGB camera image and 32-beam LiDAR point cloud. Time cost of our algorithm is dominated by semantic feature extraction from camera images. The semantic segmentation model we adopt, PSPNet [27], was a large CNN model tuned for accuracy rather than efficiency. To achieve the goal of real-time online calibration, we use NVIDIA high performance inference engine, TensorRT [32]. In order to gain the most speedup without sacrificing noticeable accuracy, we customize some of the PSPNet layers on TensorRT. For instance, we replace one of the most computational expensive layers in the model, resize-bilinear, with our own CUDA implementation to enable multi-channel linear resize operation in FP16 precision. We experimented with two different precisions, FP32 and FP16, for PSPNet inference. When using the FP16 precision, our final PSPNet model takes only 167.744 milliseconds per camera image frame. On Pascal VOC 2012 [33], the segmentation precision of our modified PSPNet implementation for cars is 0.8016 and 0.8012 when using FP32 and FP16 precisions, respectively. This demonstrates that while FP16 bring considerable performance gain, it does not cause noticeable degradation on segmentation precision nor calibration accuracy. To achieve converged solution as reported in the paper, we set τ to 500 and δ to 100. In Table I, we list and compare our pipeline's performance statistics on CPU and GPU.

Levinson and Thrun [21] suggested that multiple frames can be used for calibration optimization. We compare using 1, 5, and 10 frames. We synthetically apply the same perturbation to precalibrated extrinsic parameters (using offline method) of 10 examples as initial solutions, and measure the quality of converged solutions by the residual with respect to the precalibrated parameter configuration in terms of conventional L_2 norm for $\mathbb{R}^6$, which provides concise upper bound for both translational and rotational geometric terms. As shown in Table II, we observe more frames generally need more iterations to converge, but it does not bring noticeable quality improvement. Similarly, we also compare the results of choosing different non-monotonic line search window size ω. As shown in the table, more iterations are needed in

	CPU	GPU(FP32)	GPU(FP16)
Image Segmentation	3.419 s	232.708 ms	167.744 ms
Image Processing	2.669 s	24.42 ms	6.986ms
Optimization	86.573 ms	N/A	N/A

TABLE I
PERFORMANCE OF THE METHOD WHEN DEPLOYED ON DIFFERENT COMPUTATIONAL PLATFORMS.

	Frame Size	ω = 2	ω = 5	ω = 10
Iteration	**1**	37/20	72/22	77/17
	5	72/33	85/27	81/18
	10	88/24	83/24	80/17
Residual	**1**	0.96/1.45	1.17/0.99	1.02/0.80
	5	0.96/0.91	0.85/0.53	0.95/0.70
	10	0.98/0.95	1.06/0.84	0.95/0.64

TABLE II

CONVERGENCE AND QUALITY PERFORMANCE (MEAN/STANDARD DEVIATION) OF OUR OPTIMIZATION SOLVER WHEN CHOOSING DIFFERENT SENSOR DATA FRAME WINDOW SIZE (1, 5, 10) AND DIFFERENT NON-MONOTONIC LINE SEARCH WINDOW SIZE (ω = 2, 5, 10). LARGER WINDOW SIZE GENERALLY IMPLIES SLOWER CONVERGENCE.

general to converge to local optimal solution when larger ω is chosen. Therefore, we adopt single frame calibration and set ω to 5 for real time application purpose. Multiple frames and larger ω may be more reasonable when robustness is a major concern.

B. Edge Feature vs. Semantic Feature

In this section, we compare our semantic feature based approach with previous edge feature based approach and demonstrate its performance through practical examples. To evaluate the impact of different approaches, we apply the same optimization solver as described in section IV to both of them for fair comparison. Similar to the previous experiment, we synthetically perturb precalibrated extrinsic parameters, and apply both approaches to see if they can recover the original correct solution. Compare the residual between the precalibrated parameter configuration and converged solutions, and plot the progress of both approaches in the same figure as shown in Figure 6. The edge based approach has more non-trivial local optima and the solver easily gets stuck and stopped making any progress. On the other hand, semantic based approach is more solver friendly and is able to recover original extrinsic parameter configuration in this case.

Next we compare two approaches on 10 more examples with various degree of perturbation. We randomly perturb

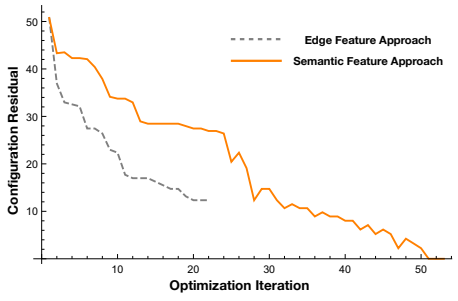

Fig. 6. Convergence performance of our optimization method when applied to edge feature based metric and semantic feature based metric. The edge-feature approach has many local optima that can make optimization solvers easily stuck, while semantic-feature approach is more friendly to solvers. The edge-feature approach converges faster according to our stopping criteria but produces suboptimal solutions.

Perturbation	Edge Feature	Semantic Feature
0-1	3.777e-3/1.927e-3	8.869e-3/7.365e-3
1-5	3.429/3.660	1.219e-1/2.701e-1
5-15	11.495/7.263	9.058e-1/8.944e-1
15-30	23.025/9.766	2.502/2.746
30-60	52.179/14.836	10.604/8.865

TABLE III

RESIDUAL STATISTICS (MEAN/STANDARD DEVIATION) OF THE CONVERGED SOLUTION WHEN USING THE EDGE FEATURE-BASED METRIC AND THE SEMANTIC-FEATURE BASED METRIC FOR CALIBRATION TASKS WITH INCREASING DIFFICULTY.

each example's precalibrated extrinsic configuration with residual ranging from $[0, 1]$, $[1, 5]$, $[5, 15]$, $[15, 30]$ and $[30, 60]$. As the initial solution of optimization solver gets farther and farther away from the expected solution, the difficulty of the calibration task increases. We demonstrate how both approaches behave in this situation. As shown in Table III, we compare the mean and standard deviation of both methods' output residual with respect to the precalibrated one. As shown in the table, when the perturbation is small, both approaches are able to recover the extrinsic parameters quite accurately. As the miscalibration gets more severe, the edge feature based approach quickly starts to produce poor calibration results. On the other hand, the semantic feature based approach has a much more robust and stable performance.

Similar to many other algorithms, we require obstacles with different range of distance to our autonomous vehicle in order to obtain accurate results. Small errors will be amplified when fused data has large depth value or lens distortion is no longer ignorable. By taking more obstacle types into account, we are expecting to improve our results further, as obstacles with larger spectrum of distance to our vehicle will contribute to the online calibration process.

VI. CONCLUSIONS

We demonstrate the use of camera's semantic features and LiDAR point clouds to construct a solver friendly extrinsic calibration quality metric. Such measurement can be used to automatically determine if the current extrinsic configuration is accurate or not, which is an essential step for self driving vehicles to detect and report calibration errors during operation. We expect future exploration on combining LiDAR semantic information to further improve our method. By combining non-monotonic line search and subgradient ascent, we are able to estimate the optimal calibration parameters robustly and efficiently. Our experiments with real world self driving tasks show promising performance improvement compared to existing algorithms and the ability to robustly recover from miscalibrated configurations. Moreover, it can also be combined with better frame selection approaches, such as RANSAC and other heuristic methods. Finally, poor segmentation quality or coarse LiDAR scan will degrade the calibration accuracy, but our method can still be used to provide good initial estimation for offline calibration approaches.

REFERENCES

[1] Q. Zhang and R. Pless, "Extrinsic calibration of camera and laser range finder," *IEEE International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2301–2306, 2004.

[2] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," Carnegie Mellon University, Tech. Rep. CMU-RI-TR-05-09, 2005.

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

[4] P. Nunez, P. Drews, R. Rocha, and J. Dias, "Data fusion calibration for a 3d laser range finder and a camera using inertial data," in *ECMR*, 2009.

[5] G. Li, Y. Liu, L. Dong, X. Cai, and D. Zhou, "An algorithm for extrinsic parameters calibration of a camera and a laser range finder using line features," *IEEE International Conference on Intelligent Robots and Systems*, pp. 3854–3859, 2007.

[6] Z. Hu, Y. Li, N. Li, and B. Zhao, "Extrinsic calibration of 2-d laser rangefinder and camera from single shot based on minimal solution," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 4, pp. 915–929, 2016.

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

[8] V. Martin, Z. Michal, and H. Adam, "Calibration of rgb camera with velodyne lidar," *International Conference on Computer Graphics, Visualization and Computer Vision*, 2014.

[9] C. Guindel, J. Beltran, D. Martin, and F. Garcia, "Automatic extrinsic calibration for lidar-stereo vehicle sensor setups," *International Conference on Intelligent Transportation Systems*, pp. 1–6, 2017.

[10] S. Sim, J. Sock, and K. Kwak, "Indirect correspondence-based robust extrinsic calibration of lidar and camera," *Sensors*, vol. 16, no. 6, p. 933, 2016.

[11] X. Gong, Y. Lin, and J. Liu, "3d lidar-camera extrinsic calibration using an arbitrary trihedron," *Sensors*, vol. 13, no. 2, pp. 1902–1918, 2013.

[12] A. Kassir and T. Peynot, "Reliable automatic camera-laser calibration," *Proceedings of the 2010 Australasian Conference on Robotics & Automation*, 2010.

[13] Y. Bok, D. Choi, and I. Kweon, "Extrinsic calibration of a camera and a 2d laser without overlap," *Robotics and Autonomous Systems*, vol. 78, pp. 17–28, 2016.

[14] 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.

[15] Z. Taylor and J. Nieto, "A mutual information approach to automatic calibration of camera and lidar in natural environments," *Australian Conference on Robotics and Automation*, pp. 3–5, 2012.

[16] G. Pandey, J. McBride, S. Savarese, and R. Eustice, "Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information," *AAAI Conference on Artificial Intelligence*, 2012.

[17] N. Williams, K. Low, C. Hantak, M. Pollefeys, and A. Lastra, "Automatic image alignment for 3d environment modeling," *Brazilian Symposium on Computer Graphics and Image Processing*, pp. 388–395, 2004.

[18] A. Mastin, J. Kepner, and J. Fisher, "Automatic registration of lidar and optical images of urban scenes," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2639–2646, 2009.

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

[20] S. Bileschi, "Fully automatic calibration of lidar and video streams from a vehicle," *International Conference on Computer Vision Workshops*, pp. 1457–1464, 2009.

[21] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers," *Robotics: Science and Systems*, vol. 2, 2013.

[22] Z. Taylor, J. Nieto, and D. Johnson, "Automatic calibration of multi-modal sensor systems using a gradient orientation measure," *International Conference on Intelligent Robots and Systems*, pp. 1293–1300, 2013.

[23] J. Castorena, U. Kamilov, and P. Boufounos, "Autocalibration of lidar and optical cameras via edge alignment," *International Conference on Acoustics, Speech and Signal Processing*, pp. 2862–2866, 2016.

[24] X. Liu, Z. Deng, and Y. Yang, "Recent progress in semantic image segmentation," *Artificial Intelligence Review*, vol. 52, pp. 1089–1106, 2019.

[25] G. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 2368–2382, 2011.

[26] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

[27] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2881–2890, 2017.

[28] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," *Proceedings of the European Conference on Computer Vision*, pp. 405–420, 2018.

[29] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[30] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," *A Field Guide to Dynamical Recurrent Neural Networks. IEEE Press*, 2001.

[31] H. Zhang and W. Hager, "A non-monotone line search technique and its application to unconstrained optimization," *SIAM Journal on Optimization*, vol. 14, no. 4, pp. 1043–1056, 2004.

[32] NVIDIA, "TensorRT," 2018. [Online]. Available: https://developer.nvidia.com/tensorrt

[33] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results." [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html