# Target-less Camera-LiDAR Extrinsic Calibration Using a Bagged Dependence Estimator

Kiyoshi Irie<sup>1,2</sup>

Masashi Sugiyama<sup>3</sup>

Masahiro Tomono<sup>1</sup>

Abstract—The goal of this study is to achieve automatic extrinsic calibration of a camera-LiDAR system that does not require calibration targets. Calibration through maximization of statistical dependence using mutual information (MI) is a promising approach. However, we observed that existing methods perform poorly on outdoor data sets. Because of their susceptibility to noise, objective functions of previous methods tend to be non-smooth, and gradient-based searches fail in local optima. To overcome these issues, we introduce a novel dependence estimator called bagged least-squares mutual information (BLSMI). BLSMI is a combination of methods composed of a kernel-based dependence estimator and noise reduction by bootstrap aggregating (bagging), which can handle richer features and robustly estimate dependence. We compared ours with previous methods using indoor and outdoor data sets, and observed that our method performed best in terms of calibration accuracy. While previous methods showed degraded performance on outdoor data sets because of the local optima problem, our method exhibited high calibration accuracy both on indoor and outdoor data sets.

# I. INTRODUCTION

Combining a 3D LiDAR and camera is a popular means of acquiring colored 3D points (RGB+depth) information. Such a sensor combination is particularly useful in outdoor environments because Kinect-type RGB-D cameras have a limited range of measurement and are ineffective under direct sunlight. To assign image colors to LiDAR points correctly, identifying the geometric relationship between the sensors is necessary. This study considers the problem of extrinsic calibration (i.e., estimation of the rigid-body transform) between a 3D LiDAR and monocular camera using sensor data.

Extrinsic calibration is a general concern in using multiple sensors. Calibration problems are often formulated as registering multiple sensor data. Although two sensor data of the same modality (e.g., two color images) are relatively easy to register, registering multi-modal sensor data (e.g., a point cloud and color image) is harder because we must identify correspondences among completely different information.

To overcome this difficulty, calibration targets designed to enable easy finding of correspondences have been used [1] [2] [3] [4]. Recently, several calibration methods that do not

<sup>2</sup>Department of Computer Science, Tokyo Institute of Technology, 2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan

<sup>3</sup>Masashi Sugiyama is with Department of Complexity Science and Engineering, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan sugi@k.u-tokyo.ac.jp require special targets (so called *target-less* calibration) have been proposed [5] [6] [7]. While calibration of a system is typically done during the production process, recalibration is also important for long-life systems because sensor configurations can change over time. Target-less calibration can reduce the cost of preparing calibration targets and enable on-site and on-line recalibration.

Existing target-less calibration methods can be classified into two categories by their registration approach: edge-based and dependence-based. The method proposed by Levinson and Thrun [8] is a seminal work on edge-based calibration. Discontinuities in LiDAR scans and image edges are matched together to evaluate the calibration quality. While Levinson and Thrun [8] employed only the strength of the edges, Taylor *et al.* [10] reported the usefulness of the orientation of edges. They proposed using gradient orientation measure that can evaluate the degree to which edge orientations are aligned between a camera image and LiDAR reflectivity image.

In dependence-based methods, calibration is performed by maximizing a dependence metric. The assumption is that multiple sensor data that is observed from the same object should have dependence. For example, reflectivity of LiDAR measurements (the intensity of laser return) tends to be high on white objects and vegetation, and low on dark-colored objects. This implies that considerable dependence between reflectivity and image color exists. Several registration methods using the property have been proposed [11] [12], and Pandey *et al.* [5] as well as Taylor and Nieto [6] applied the idea to calibration. Although the method of Pandey *et al.* [5] maximizes mutual information (MI) using gradient ascent, we observed that it often got stuck in local optima with outdoor data sets as a result of the non-smoothness of the objective function.

In this paper, we propose a novel dependence-based calibration method. The major component we employ is bagged least-squares mutual information (BLSMI) that allows us to incorporate more features than previous methods and provides a considerably smoother objective function. We tested our method by a series of comparative experiments and observed significant improvements in accuracy and robustness.

#### **II. PROPOSED METHOD**

Our proposed method can be regarded as an extension of existing dependence-based calibration methods [5] [6]. Tab. I summarizes the differences between our own and previous

<sup>\*</sup>This work was partly supported by the New Energy and Industrial Technology Development Organization (NEDO).

<sup>&</sup>lt;sup>1</sup>Kiyoshi Irie and Masahiro Tomono are with Future Robotics Technology Center, Chiba Institute of Technology, 2-17-1 Tsudanuma, Narashino-shi, Chiba 275-0016, Japan {irie, tomono}@furo.org

 TABLE I

 COMPARISON OF DEPENDENCE-BASED CALIBRATION METHODS

	Metric	Optimization	LiDAR feature (dimensionality)	Image feature (dimensionality)
Pandey et al. [5]	MI	steepest gradient descent	reflectivity (1)	grayscale intensity (1)
Taylor and Nieto [6]	NMI	particle swarm optimization	surface normal (1)	grayscale intensity (1)
Proposed method	SMI	quasi-Newton (BFGS)	reflectivity (1) discontinuity (1) surface normal (3)	color (3) edge strength (1)

methods. This section describes the proposed method and provides detailed comparisons.

#### A. Overview of Dependence-based Calibration

A point cloud collected by a LiDAR and an image captured by a camera are used as input. The number of total points in the cloud is denoted by *n* and features extracted on the points are denoted by real vectors  $\{x_i\}_{i=1}^n$ . The correspondence between the LiDAR data and camera image is made by projecting the 3D points onto the image plane. By using the 6-DoF relative camera pose with respect to the LiDAR coordinate frame  $\boldsymbol{\Theta} = [t_x, t_y, t_z, r_x, r_y, r_z]$ , we calculate the projected pixel location  $(u_i, v_i)$  for each 3D point.

Image features extracted at  $(u_i, v_i)$  in the image are denoted by real vectors  $\{y_i\}_{i=1}^n$ . The LiDAR and image features are denoted by random vectors **X** and **Y**, respectively. Under the assumption that data pairs  $\{(x_i, y_i)\}_{i=1}^n$  are generated from the joint probability  $p(X, Y; \Theta)$ , we maximize an objective function f that measures statistical dependence between **X** and **Y**, to determine the calibration parameter  $\Theta$ .

$$\hat{\boldsymbol{\Theta}} = \operatorname*{argmax}_{\boldsymbol{\Theta}} f(\boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{\Theta}). \tag{1}$$

Several statistical dependence measures including MI and normalized MI (NMI) have been employed as f. Because fis generally non-convex and the optimization is not trivial, various methods have been applied to solve this maximization problem [5] [6].

## B. Features

While different sets of features have been employed in the existing dependence-based calibration methods [5] [6], we have found that using a richer set of features is important for improving the robustness against varying environments. Therefore, our method uses a combined set of features (Tab. I).

Each of the features employed previously has both advantages and disadvantages. The image grayscale intensity and laser reflectivity (intensity of laser return) employed in Pandey *et al.* [5] are less useful in outdoor scenes. For outdoor calibration, Taylor and Nieto [6] reported the effectiveness of surface normal information extracted from LiDAR measurements. We have found that the combination of these complementary features improves calibration accuracy on both indoor and outdoor datasets.

In addition, we have found that incorporating the edgebased features used in the method proposed by Levinson and Thrun [8] improve the calibration accuracy. The underlying assumption of their method is that depth changes are likely to appear as edges in an image [9]. The assumption can also be regarded as the existence of dependence between the discontinuity and edge features.

We present detailed feature extraction procedures in Section III-A.

#### C. Optimization

We employ the BFGS quasi-Newton method [13], which is a gradient-based algorithm, to maximize (1). Because the dependence function is not always smooth, a gradient search algorithm can easily get stuck in a local maximum. To overcome this problem, we follow the data aggregation approach presented in [5]. By aggregating many laser scans and images collected in different places, the dependence function becomes smooth.

A different approach to overcoming local maxima can be found in Taylor and Nieto [6]. They employed the particle swarm optimization (PSO) algorithm, which is a heuristic search used to find the global optimum of a non-convex problem. However, the computational cost of PSO is high because a large number of samples are employed during the search. Therefore, they accelerated their implementation by GPU.

#### D. Dependence Measure

We employ squared-loss mutual information (SMI) as a dependence measure. The definition of MI and SMI are given as:

$$MI(\boldsymbol{X}, \boldsymbol{Y}) := \iint p(\boldsymbol{x}, \boldsymbol{y}) \log \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})} d\boldsymbol{x} d\boldsymbol{y},$$
  

$$SMI(\boldsymbol{X}, \boldsymbol{Y}) := \frac{1}{2} \iint p(\boldsymbol{x})p(\boldsymbol{y}) \left(\frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})} - 1\right)^2 d\boldsymbol{x} d\boldsymbol{y}.$$

Both MI and SMI are non-negative and have large values when the dependence is high. One of the advantages of SMI over ordinary MI is that SMI is robust against outliers because it does not include logarithm [14]. Previous methods estimate discretized versions of MI and NMI using joint histogram-based estimators for computational efficiency [5] [6]. However, such histogram-based estimators are strongly affected by the growth of dimensionality and cannot be used with high-dimensional data. For example, if the total dimensionality of the features is four, a four-dimensional histogram must be constructed. If we discretize each feature into 256 steps and count the frequency using 32-bit integers,  $256^4 \times 4bytes = 16GB$  of memory is required. Because our method uses nine-dimensional features, applying the estimator with current computer systems is impossible (this kind of phenomenon is known as *the curse of dimensionality* [15]).

To alleviate the issue of the dimensionality, we estimate the dependence directly in the continuous space using LSMI [16], which is a kernel-based, computationally efficient SMI estimator.

# E. LSMI

Here, we briefly review LSMI [16] for the sake of completeness. In LSMI, the density ratio,

$$r(\mathbf{x},\mathbf{y}) := \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})p(\mathbf{y})}$$

is estimated using the following multiplicative kernel model:

$$r_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y}) := \sum_{l=1}^{b} \sum_{l'=1}^{b} \alpha_{l,l'} K(\boldsymbol{x},\tilde{\boldsymbol{x}}_l) L(\boldsymbol{y},\tilde{\boldsymbol{y}}_{l'}).$$

Here, K and L are kernel basis functions. We employ Gaussian kernel functions:

$$K(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \exp\left(-\frac{||\boldsymbol{x} - \tilde{\boldsymbol{x}}||^2}{2\sigma^2}\right), L(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = \exp\left(-\frac{||\boldsymbol{y} - \tilde{\boldsymbol{y}}||^2}{2\sigma^2}\right),$$

where  $\sigma > 0$  is the Gaussian bandwidth. Kernel centers  $\{(\tilde{\mathbf{x}}_l, \tilde{\mathbf{y}}_l)\}_{l=1}^b$  are randomly chosen from the input data (b = 50 in our experiments). The parameter  $\boldsymbol{\alpha}$  ( $b \times b$  matrix) is estimated to minimize the following squared loss function combined with the  $\ell_2$  regularization term:

$$J(\boldsymbol{\alpha}) := \iint (r_{\boldsymbol{\alpha}}(\boldsymbol{x}, \boldsymbol{y}) - r(\boldsymbol{x}, \boldsymbol{y}))^2 p(\boldsymbol{x}) p(\boldsymbol{y}) \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{y} + \lambda ||\boldsymbol{\alpha}||_2,$$

where  $\lambda > 0$  is the regularization parameter. By approximating the expectations by the empirical averages and setting the derivative of  $J(\alpha)$  equal to zero, we obtain the following discrete Sylvester equation about  $\alpha$ :

$$\frac{1}{n^2} \boldsymbol{K}^T \boldsymbol{K} \boldsymbol{\alpha} \boldsymbol{L}^T \boldsymbol{L} + \lambda \boldsymbol{\alpha} = \frac{1}{n} \boldsymbol{K}^T \boldsymbol{L}.$$
 (2)

Elements of **K** and **L** (both are  $n \times b$  matrices) are given by  $K_{i,l} = K(x_i, \tilde{x}_l)$ ,  $L_{i,l} = L(y_i, \tilde{y}_l)$ . Finally, estimated SMI is obtained using  $\hat{\alpha}$ , which is the solution of (2), as follows:

$$LSMI = \frac{1}{2n} \operatorname{tr}(\boldsymbol{K} \hat{\boldsymbol{\alpha}} \boldsymbol{L}^T) - \frac{1}{2}.$$
 (3)

The Gaussian width  $\sigma$  and the regularization parameter  $\lambda$  are hyper-parameters and can be determined by *k*-fold cross-validation with respect to the criterion *J*.

# F. Bagged LSMI

To further improve the smoothness of the objective function, we employ the bagging technique [17]. Bagging reduces the variance in estimation by averaging multiple solutions obtained from resampled data sets. We generate bootstrap data sets  $\{\mathcal{D}_m\}_{m=1}^M$  by randomly subsampling from the original input data  $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ . We estimate SMI for each bootstrap data set and calculate the mean of the estimated values as:

$$BLSMI(\mathscr{D}) = \frac{1}{M} \sum_{m=1}^{M} LSMI(\mathscr{D}_m).$$
(4)



Fig. 1. (a) Multisense SL used in experiments. The 2D LiDAR (Hokuyo UTM-30LX-EW) mounted on a spindle rotates continuously. (b) Reflectivity response of UTM-30LX-EW against a white paper (blue dots) and a fitted curve (red line).

We refer to (4) as *bagged LSMI (BLSMI)* and use it to measure dependence.

## III. EXPERIMENTAL SETUP

We evaluated our method using the data collected by a Multisense SL (Fig. 1 (a)) from Carnegie Robotics because its factory calibrated parameters are available. Multisense SL consists of a spinning 2D LiDAR (UTM-30LX-EW) and stereo camera. In the following experiments, we estimated the relative pose between the LiDAR and left camera (right camera images were not used).

#### A. Details of Feature Extraction

Here, we present some implementation details of the feature extraction.

1) UTM-30LX-EW Reflectivity Calibration: Hokuyo UTM-30LX-EW can output a reflectivity response for each distance measurement. The reflectivity value depends on the observed objects, especially on their material and surface orientation, and also on the distances to the objects. While the material and surface orientation should have strong dependence on image color features, the distance to the object has little dependence. Therefore, the dependence between reflectivity and distance can hinder identifying the dependence between measurements from a LiDAR and camera.

To alleviate the effect of the distance, we normalized reflectivity regarding the distance. We collected reflectivity data of a white paper measured from a perpendicular direction with different distances. Collected measurements are shown in Fig. 1 (b). We employed the least-squares method and determined that the following function fits well to the data:

$$g(x) = a \exp(bx) + c \exp(dx),$$
  
where  $a = 4207, b = -0.06993,$   
 $c = 2.814 \times 10^4, d = -1.981$ 

The normalized reflectivity is calculated using the raw reflectivity measurements  $\{r_i\}_{i=1}^n$  and the corresponding distance measurements  $\{d_i\}_{i=1}^n$  as follows.

$$r_i^{\text{normalized}} = r_i/g(d_i).$$

Exemplary raw and normalized reflectivity measurements are presented in Fig. 2 (a) and (b).



(c) LiDAR discontinuity (d) Surface normals

Fig. 2. Sample features extracted from LiDAR measurements. Green indicates higher and red indicates lower reflectivity/discontinuity. White lines in (d) indicates surface normal vectors.

2) LiDAR Scan Discontinuity: The LiDAR discontinuity feature is calculated as described in [8]. Let  $d_1, ..., d_L$  be the distance measurements in a single (2D) scan. The discontinuity feature  $e^{\text{LiDAR}}$  for the *i*-th measurement is calculated by:

$$e_i^{\text{LiDAR}} = \max(d_{i-1} - d_i, d_{i+1} - d_i, 0)^{0.5}$$

The max operator is used because closer points at the discontinuity are more likely to coincide with image edges. An example of an extracted discontinuity feature is given in Fig. 2 (c).

*3) Surface Normal:* Surface normals were extracted as follows. For each 3D point, we extracted 100 nearest-neighbor points and applied least-squares plane fitting to the following plane equation.

$$ax + by + cz + d = 0$$

The estimated coefficients a, b, c were used as threedimensional features. We used the implementation included in the Point Cloud Library [18] to calculate the plane parameters. Fig. 2 (d) shows an example of surface normal detection.

4) Image Features: In our experiments, the image size of Multisense SL's camera was  $1024 \times 544$ . RGB pixel values were used as three-dimensional color features. Another image feature employed was edge strength, which was calculated as described in [8]. For each pixel, grayscale intensity was compared with eight-neighbor pixels and the maximum intensity difference (non-negative) was used as a one-dimensional feature  $e^{image}$  for the pixel. To ensure the objective function is smooth, we applied a Gaussian filter to the extracted image features. A Gaussian width  $\sigma = 2$  was used in our experiments.

The extracted features were stored pixel-wise. However, when we project 3D points onto an image, the projected

points do not exactly correspond to single pixels (i.e. coordinates are not always integer). We have found that subpixel interpolation considerably improves the smoothness of the objective function. Instead of simply rounding the coordinates into integer, we calculated sub-pixel image features using the following linear interpolation. The feature for image location u, v is calculated by:

$$(1-\beta)((1-\alpha)I(\lfloor u \rfloor, \lfloor v \rfloor) + \alpha I(\lceil u \rceil, \lfloor v \rfloor)) + \beta((1-\alpha)I(\lfloor u \rfloor, \lceil v \rceil) + \alpha I(\lceil u \rceil, \lceil v \rceil)),$$
where  $\alpha = u - \lfloor u \rfloor$ ,  $\beta = v - \lfloor v \rfloor$ ,

and, I(i, j) denotes the image feature for pixel (i, j).

#### B. Comparative Methods

We implemented the following comparative methods.

Pandey et al. [5]: Grayscale intensity of the image and reflectivity of LiDAR measurements were used as input features. MI was employed for the objective function f and estimated in the following steps. First,  $p(X,Y;\Theta)$  was estimated by kernel density estimation (KDE) in a discretized space (256×256). Our implementation of KDE was constructing a 2D joint histogram and applying a Gaussian filter to it [19]. Discretized MI was then calculated by:

$$MI(\boldsymbol{X},\boldsymbol{Y}) = H(\boldsymbol{X}) + H(\boldsymbol{Y}) - H(\boldsymbol{X},\boldsymbol{Y}),$$
(5)

$$H(\boldsymbol{X}) = -\sum_{\boldsymbol{x}} p(\boldsymbol{x}) \log p(\boldsymbol{x}), \tag{6}$$

$$H(\boldsymbol{Y}) = -\sum_{y} p(\boldsymbol{y}) \log p(\boldsymbol{y}), \tag{7}$$

$$H(\boldsymbol{X},\boldsymbol{Y}) = -\sum_{x} \sum_{y} p(\boldsymbol{x},\boldsymbol{y}) \log p(\boldsymbol{x},\boldsymbol{y}).$$
(8)

Here,  $p(\mathbf{x})$  and  $p(\mathbf{y})$  are the marginal distributions.

*Taylor and Nieto:* Taylor and Nieto [6] proposed using surface normals calculated from LiDAR measurements. The following one-dimensional surface normal feature was calculated for each 3D measurement  $\{(p_i^x, p_i^y, p_i^z)\}_{i=1}^L$  in a single 2D scan:

$$\eta_i = \frac{(p_i^y - p_{i+1}^y)^2}{\sqrt{(p_i^x - p_{i+1}^x)^2 + (p_i^z - p_{i+1}^z)^2}}$$

Grayscale intensity was used as the image feature. NMI, which is calculated as follows, was employed as the objective function:

$$NMI(\boldsymbol{X}, \boldsymbol{Y}) = \frac{H(\boldsymbol{X}) + H(\boldsymbol{Y})}{H(\boldsymbol{X}, \boldsymbol{Y})}.$$
(9)

The joint probability was estimated similarly to that in Pandey *et al.* [5].

*Discretized SMI:* This method was introduced to demonstrate the difference of dependence measures. In this method, the same features and method described in Pandey *et al.* [5] were employed to estimate the histogram of the joint probability. The following discretized SMI was calculated from the histogram.

$$SMI(\boldsymbol{X}, \boldsymbol{Y}) = -\frac{1}{2} \sum_{x} \sum_{y} p(\boldsymbol{x}) p(\boldsymbol{y}) \left( \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x}) p(\boldsymbol{y})} - 1 \right)^{2}.$$
 (10)

*BLSMI (proposed method):* In our proposed method, we employed all the features shown in Tab. I. Bootstrap data sets for BLSMI were generated by randomly subsampling 1% of the total input points (e.g., 2,000 points when 20 scenes were used). To reduce the computation time, cross-validation in LSMI estimation was performed only on the first bootstrap data set and hyper-parameters were reused for all other data sets. In our preliminary experiments, omitting cross-validation drastically reduces the processing time without degrading calibration accuracy. The number of bagging replications was set to 200, unless otherwise noted.

LSMI without bagging: This method is exactly the same as the proposed method except that bagging was not performed. Single LSMI estimation using all input data was used as the objective function f.

*Levinson and Thrun [8]:* Because the original method described in [8] is designed for online calibration, we slightly modified their objective function for application to the offline (batch) calibration problem. The following batch version of their objective function was employed:

$$J_c = \frac{1}{n} \sum_{i=1}^{n} e_i^{\text{LiDAR}} \cdot e_i^{\text{image}}.$$
 (11)

Here,  $e_i^{\text{image}}$  is the image edge strength that corresponds to the *i*-th laser measurement. Maximizing the objective function (11) can be interpreted as maximizing *linear* dependence between LiDAR discontinuity and image edge strength. By contrast, our method employs an MI-based measure to handle *non-linear* dependence.

#### **IV. RESULTS**

We collected data in indoor and outdoor scenes and evaluated our method in two ways: qualitative evaluation of the objective functions and quantitative evaluation of the calibration accuracy. The factory calibrated parameters of the Multisense SL were used as the ground truth.

#### A. Datasets

We collected 3D scans of LiDAR and camera images from 19 indoor and 20 outdoor scenes (Fig. 3). Because the importance of nearby objects was pointed out by Pandey *et al.* [5], we included some outdoor scenes with nearby objects. Each scene contained a single 3D scan (360° rotation of the 2D LiDAR) and a color image.

Although Multisense SL's 3D LiDAR has a considerably larger field of view (FoV) than its camera, points out of the camera's FoV are not useful for calibration. To reduce the computational cost, we reduced the number of points by limiting the FoV of the 3D LiDAR to  $60^{\circ} \times 40^{\circ}$ . In addition, we randomly extracted 10,000 points from each scene. The points extracted from different scenes were combined together and used as the input for calibration.

#### B. Comparison of Objective Functions

As preliminary experiments, we investigated the objective functions of the proposed and previous methods. Exemplary objective functions around the ground truth are shown in



outdoor data

Fig. 3. Sample input data used in experiments. Top: indoor scenes. Bottom: outdoor scenes. Some of outdoor scenes contained nearby objects.

Fig. 4. The objective function employed in Pandey *et al.* [5] appeared smooth for indoor data sets, but it exhibited highly non-smooth curvature for outdoor data sets. A possible cause of the difference is noises in the data. In outdoor environments, objects were not completely motionless (e.g., tree branches were swaying in the wind). In addition, over and under exposures caused by strong sunlight would interfere in the dependence between laser reflectivity and image intensity. By contrast, the objective function of the proposed method appeared to have smooth curvatures and exhibited single maxima for both data sets.

#### C. Evaluation of Calibration Accuracy

We evaluated the calibration accuracy of our method compared to others. Note that we used only features and dependence measures from the previous methods. The same BFGS search was applied to all methods for comparison. We implemented BFGS search using the fminunc function in MATLAB [20] and the gradients were calculated numerically. To evaluate the stability, we conducted calibration experiments 100 times using different input data. At the beginning of each experiment, points were randomly extracted (again, 10,000 points per scene). To simulate errors by a hand-measured calibration, the parameters were initialized by adding uniform random noise ( $\pm 3$  cm and  $\pm 3^{\circ}$ ) to the ground truth parameters. The results are summarized in Fig. 5, showing that the proposed method exhibited the smallest repeatability error.

The quantitative calibration accuracy was measured using projection errors. 3D points in the data sets were projected onto the image plane using the estimated calibration parameters. We calculated the 2D position difference between points projected using estimated parameters and of that using ground truth parameters. Fig. 7 summarizes the results. The mean projection error of our method was 3.1 pixels using



Fig. 4. Sample visualization of objective functions.

19 indoor scans and 4.6 pixels using 20 outdoor scans, and they were significantly smaller (p < 0.01 by the paired t-test) than any other comparative methods. On outdoor data sets, the method of Pandey *et al.* [5] often got stuck in local optima and resulted in degraded performance.

Exemplary colored point clouds obtained using the calibration results (with 20 outdoor scans) are shown in Fig. 6. The quality of the colored point clouds appeared to be competitive to the ground truth. Therefore, we consider the estimated parameters to be sufficient to acquire colored 3D points.

#### D. Comparison of Features

As we integrated several features, the contribution of each feature is of our interest. We compared calibration accuracy

using the different subset of LiDAR features. We again executed calibration experiments 100 times for each subset of features, using BLSMI as the objective function. Fig. 8 summarizes the result. It can be seen that single feature is not enough for indoor-outdoor calibration; reflectivity and discontinuity were not effective on outdoor data sets and surface normal performed poorly on indoor data sets. By combining these features, we were able to compensate the weakness and also improve the calibration accuracy.

We found that reasonable calibration accuracy can be obtained without using reflectivity. Therefore we consider our method can be used even with LiDARs that do not return reflectivity.



Fig. 5. Summary of calibration results over 100 runs. The proposed method exhibited the smallest repeatability errors. The vertical axes indicate calibration errors with respect to the ground truth parameters in centimeters and degrees. Error bars show standard deviations.



Fig. 6. Sample calibration results.

#### E. Comparison of the Number of Bagging Replications

# We evaluated the performance of our method for the different number of bagging replications (M in (4)). The results are summarized in Fig. 9. Although M = 200 was used in the experiments in the previous sections, M = 50 and M = 100 also seemed to provide reasonable accuracy with much less processing time.

# V. CONCLUSION

In this paper, we presented a novel extension of dependence-based calibration methods. The major issue we tackled was the non-smoothness of the objective function, which caused the previous methods to get stuck in the local optima. Richer features had to be incorporated to address the problem. However, the previously employed dependence estimators were not feasible to handle the increased dimen-



Fig. 7. Comparison of projection error using different features and dependence estimators. Error bars indicate standard errors. Our proposed method exhibited the smallest projection error. Pandey *et al.* [5] showed degraded performance on outdoor data sets because it got stuck in local optima.



Fig. 8. Comparison of calibration accuracy using different subset of LiDAR features. BLSMI was used as the objective function.

sional features. Therefore, we introduced a novel dependence estimator, BLSMI. Consequently, we obtained a smooth objective function, and an efficient gradient-based search was successfully applied.

We compared our proposed method with previous methods by experiments using indoor and outdoor data sets. While our method showed the best calibration accuracy on all the data sets we employed for evaluation, the improvement was particularly significant on outdoor data sets. Therefore, our method should be useful for outdoor calibration.

#### REFERENCES

- Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, 2004, pp. 2301–2306.
- [2] K. H. Kwak, D. Huber, H. Badino, and T. Kanade, "Extrinsic calibration of a single line scanning lidar and a camera," in *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, 2011, pp. 3283–3289.
- [3] H. S. Alismail, L. D. Baker, and B. Browning, "Automatic calibration of a range sensor and camera system," in 2012 Second Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization & Transmission (3DIMPVT), 2012.
- [4] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012, pp. 3936–3943.
- [5] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information," in *Proceedings of the AAAI National Conference* on Artificial Intelligence, 2012, pp. 2053–2059.



Fig. 9. Comparison of projection error and processing time per dependence evaluation using different number of bagging replications (20 outdoor scans were used). Reasonable calibration accuracy was obtained by 50 or more replications.

- [6] Z. Taylor and J. Nieto, "A mutual information approach to automatic calibration of camera and lidar in natural environments," in *Proc.* of The Australian Conference on Robotics and Automation (ACRA) , 2012, pp. 3–5.
- [7] A. Napier, P. Corke, and P. Newman, "Cross-calibration of pushbroom 2d lidars and cameras in natural scenes," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013, pp. 3679–3684.
- [8] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers," in *Proc. of Robotics: Science and Systems*, 2013.
- [9] L. Baboud, M. Cadik, E. Eisemann, and H.-P. Seidel, "Automatic photo-to-terrain alignment for the annotation of mountain pictures," in *Proc. of IEEE Computer Society Conference on Vision & Pattern Recognition (CVPR)*, 2011, pp. 41–48.
- [10] Z. Taylor, J. Nieto, and D. Johnson, "Automatic calibration of multimodal sensor systems using a gradient orientation measure," in *Proc.* of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS), 2013, pp. 1293–1300.
- [11] N. Williams, K.-L. Low, C. Hantak, M. Pollefeys, and A. Lastra, "Automatic image alignment for 3d environment modeling," in *Proc. of Brazilian Symposium on Computer Graphics and Image Processing*, 2004, pp. 388–395.
- [12] A. Mastin, J. Kepner, and J. Fisher, "Automatic registration of lidar and optical images of urban scenes," in *Proc. of IEEE Computer Society Conference on Vision & Pattern Recognition (CVPR)*, 2009, pp. 2639– 2646.
- [13] C. T. Kelley, Iterative Methods for Optimization (Frontiers in Applied Mathematics). Society for Industrial and Applied Mathematics, 1987.
- [14] M. Sugiyama, "Machine learning with squared-loss mutual information," *Entropy*, vol. 15, no. 1, pp. 80–112, 2013.
- [15] R. E. Bellman, Adaptive control processes A guided tour. Princeton University Press, 1961.
- [16] T. Sakai and M. Sugiyama, "Computationally efficient estimation of squared-loss mutual information with multiplicative kernel models," *IEICE Trans. on Information and Systems*, vol. E97-D, no. 4, pp. 968–971, 2014.
- [17] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [18] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2011.
- [19] G. Pandey. (2015) Extrinsic calibration of a 3d lidar and camera. [Online]. Available: http://robots.engin.umich.edu/SoftwareData/ExtrinsicCalib. [Accessed: 3- Mar- 2016]
- [20] The MathWorks, Inc. (2015) Find minimum of unconstrained multivariable function - MATLAB fminunc. [Online]. Available: http://www.mathworks.com/help/optim/ug/fminunc.html. [Accessed: 19- Mar- 2016]