Artificial Intelligence Review (2023) 56:9949–9987
https://doi.org/10.1007/s10462-022-10317-y

Automatic targetless LiDAR–camera calibration: a survey

Xingchen Li[1] · Yuxuan Xiao[1] · Beibei Wang[2] · Haojie Ren[1] · Yanyong Zhang[1] · Jianmin Ji[1]

Published online: 23 November 2022

Abstract
The recent trend of fusing complementary data from LiDARs and cameras for more accurate perception has made the extrinsic calibration between the two sensors critically important. Indeed, to align the sensors spatially for proper data fusion, the calibration process usually involves estimating the extrinsic parameters between them. Traditional LiDAR–camera calibration methods often depend on explicit targets or human intervention, which can be prohibitively expensive and cumbersome. Recognizing these weaknesses, recent methods usually adopt the autonomic targetless calibration approach, which can be conducted at a much lower cost. This paper presents a thorough review of these automatic targetless LiDAR–camera calibration methods. Specifically, based on how the potential cues in the environment are retrieved and utilized in the calibration process, we divide the methods into four categories: information theory based, feature based, ego-motion based, and learning based methods. For each category, we provide an in-depth overview with insights we have gathered, hoping to serve as a potential guidance for researchers in the related fields.

Keywords Calibration · LiDAR · Camera · Automatic · Targetless

✉ Jianmin Ji
jianmin@ustc.edu.cn

Xingchen Li
starlet@mail.ustc.edu.cn

Yuxuan Xiao
xiaoyx@mail.ustc.edu.cn

Beibei Wang
wbb@iai.ustc.edu.cn

Haojie Ren
rhj@mail.ustc.edu.cn

Yanyong Zhang
yanyongz@ustc.edu.cn

[1] School of Computer Science and Technology, University of Science and Technology of China, Jinzhai Road, Hefei 230026, Anhui, China

[2] Department of Fundamental Research, Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, West Wangjiang Road, Hefei 230088, Anhui, China

1 Introduction

In modern autonomous systems such as self-driving vehicles, accurate perception of the surrounding environment is an important capability and a prerequisite for making subsequent decisions. In order to further improve perception accuracy, autonomous systems usually apply different types of sensors and combine their advantages through data fusion (Cui et al. 2022; Feng et al. 2021; Wang et al. 2020). Among them, a most typical multi-sensor fusion is for a LiDAR sensor and an RGB camera, as shown in Fig. 1.

Currently, LiDAR–camera fusion has been widely applied to a variety of challenging tasks, such as object detection and tracking (Chen et al. 2017; Vora et al. 2020; Kim et al. 2021), simultaneous localization and mapping (SLAM) (Graeter et al. 2018; Zuo et al. 2019), and navigation (Hussein et al. 2016).

In order to fuse the data from a LiDAR sensor and a camera, it is critical to first calculate the extrinsic transformation between the two sensors in a common frame of reference. The process of the parameter estimation for the transformation between sensor coordinate systems, including rotation and translation, is called *LiDAR–camera extrinsic calibration*.

The extrinsic calibration involves finding the correspondence between data from the two sensors. LiDAR point clouds and camera images are of two distinct modalities, which differ in dimension, resolution, field of view, etc., bringing great challenges to the calibration process. Traditionally, the two sensors are calibrated by placing definite targets, such as checkerboards, polygonal boards, and boxes, in specific scenes, or by manually extracting and matching particular features from the sensor outputs. However, these methods require pricey and lengthy manual operations, which is expensive to compensate for the drifts of calibration parameters caused by displacements in the location of sensors on moving vehicles.

To approach this problem, a recent trend is to extract features or other discriminative information, such as the common attribute probability distribution and the motion trajectory, that can be used to calibrate the sensors in the actual driving environment. This approach does not require any calibration target or manual effort. As a result, it is referred to as *automatic targetless calibration*.

Automatic targetless calibration promises to revolutionize the way calibration is done, but it also brings great challenges to the design of the system. In particular, without a clear calibration target, the difficulties of feature extraction and matching increase significantly. In this paper, we carefully review recent automatic targetless calibration methods, with a special focus on how they tackle this challenge mathematically. Specifically, based on how these methods exploit potential cues in the environment, we divide them into four

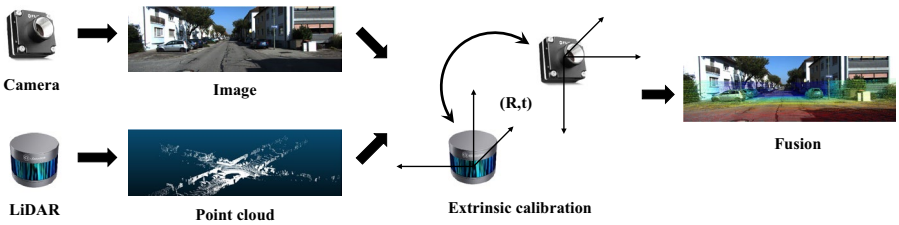


Fig. 1 The process for LiDAR–camera fusion based on extrinsic calibration, the image and the point cloud are taken from the KITTI public dataset (Geiger et al. 2013), as in the following figures

categories: (1) information theory based methods that measure the statistical similarity between joint histogram values of several common properties between the two modalities, (2) feature based methods that extract geometric, semantic or motion features from the environment, (3) ego-motion based methods that make use of the sensor-movement related information, and (4) learning based methods that use neural network models to estimate the extrinsic parameters.

Though there are already a few surveys on LiDAR–camera calibration such as Nie et al. (2021), Yaopeng et al. (2021), Khurana and Nagla (2021), and Wang et al. (2021), they usually cover a wide range of traditional calibration methods and only provide a relatively brief review of automatic targetless calibration. Given its rapidly increasing popularity, we believe a thorough and insightful review on automatic targetless LiDAR–camera calibration is both imperative and important. To summarize, our contributions are as follows:

1. We provide an accurate and inclusive automatic targetless LiDAR–camera calibration classification. Based on how the potential information in the environment is utilized to solve the extrinsic calibration problem, we divide these methods into four categories, i.e., information theory based methods, feature based, ego-motion based, and learning based methods. Then we further split each category according to their different choices for the implementation.
2. We present an extensive and detailed introduction to related studies that fall within the scope of automatic targetless calibration. Then we carefully classify and introduce these papers in detail.
3. We provide elaborate comparisons and discussions for the four categories on their characteristics and advantages, as well as their limitations.

The remainder of this paper is organized as follows. The overall structure is presented in Fig. 2. In Sect. 2 we explain the mathematical principles of extrinsic calibration between a LiDAR sensor and a camera, and introduce the criteria for the classification of current methods, then we present our automatic targetless LiDAR–camera calibration framework. Section 3 provides a review of LiDAR–camera extrinsic calibration

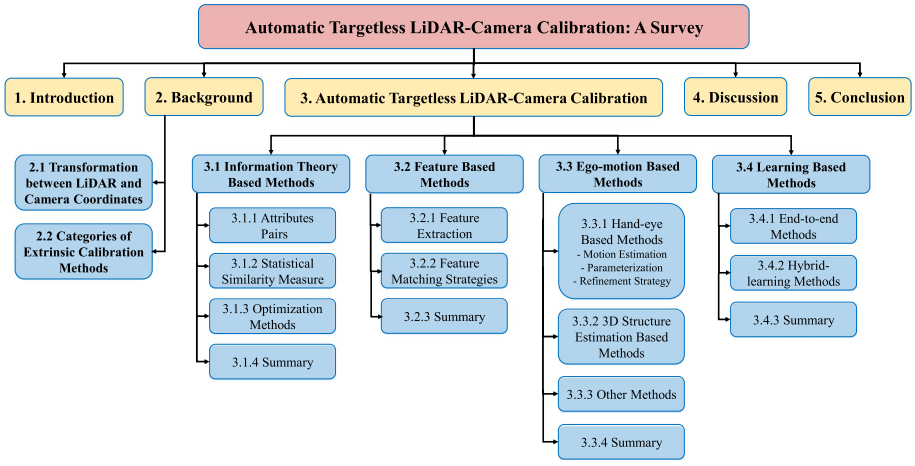


Fig. 2 Global structure of this survey

methods from the four previously-mentioned categories. In Sect. 4 we summarize and compare the four categories of methods on their pros and cons. Section 5 provides the conclusion of the paper.

2 Background

The calibration for LiDAR and camera is aimed at obtaining the transformation between the two sensors' coordinates, which enables the conversion of the data from the LiDAR sensor and the camera into the same coordinate system. Fusion of the calibrated data is crucial to improve performance for perception tasks, such as object detection, classification, tracking, and so on (Chen et al. 2017; Zhang et al. 2019).

In this section, we specify the concepts of intrinsic and extrinsic calibration parameters and review the mathematics for transformation between LiDAR and camera coordinates. Then we introduce four categories of extrinsic calibration methods, that can be summarized according to the need for calibration targets and whether human intervention is required. In this paper, we focus on automatic targetless extrinsic calibration between a LiDAR sensor and a camera.

It should be noted that the LiDAR–camera calibration here refers to the alignment of the sensor at the spatial level, which is to obtain a rigid transformation relationship between the two sensor coordinate systems. In addition, a concept related to calibration is temporal calibration or time synchronization, which is the alignment of sensors at the temporal level. Since each sensor often has different sampling frequencies, some methods need to be used to synchronize the data of multiple sensors to a unified timestamp. In this paper, we assume that temporal calibration has been performed well.

2.1 Transformation between LiDAR and camera coordinates

The transformation relationship between the coordinate systems of a LiDAR sensor and a camera is specified by *extrinsic parameters* in LiDAR–camera calibration. Meanwhile, the camera is treated as a classical pinhole camera model. Then a 3D point in the camera coordinate system is projected onto a 2D point in the image plane, where *intrinsic parameters* specifies the projection. We use both extrinsic and intrinsic parameters to transform a 3D point in the LiDAR coordinate system to a 2D pixel in the image plane and vice versa, which defines the correspondence between points and pixels. Notice that, the intrinsic parameters represent the internal properties of the camera such as focal length and principal point, which can be measured offline. Then we only need to estimate extrinsic parameters online for the transformation.

As illustrated in Fig. 3, we use O_L and O_C to denote the origin of coordinate systems attached to the LiDAR sensor (L) and the camera (C), respectively. The position coordinates of a point P w.r.t. L and C can be denoted as $P_L = [X_L\ Y_L\ Z_L]^\top$ and $P_C = [X_C\ Y_C\ Z_C]^\top$, respectively. The point P is also projected on the image plane at $p_C = [u\ v]^\top$. Then the projection between the 3D point P_C in the camera coordinates and the 2D point p_C on the image plane, i.e., intrinsic parameters, is specified by the following equation:

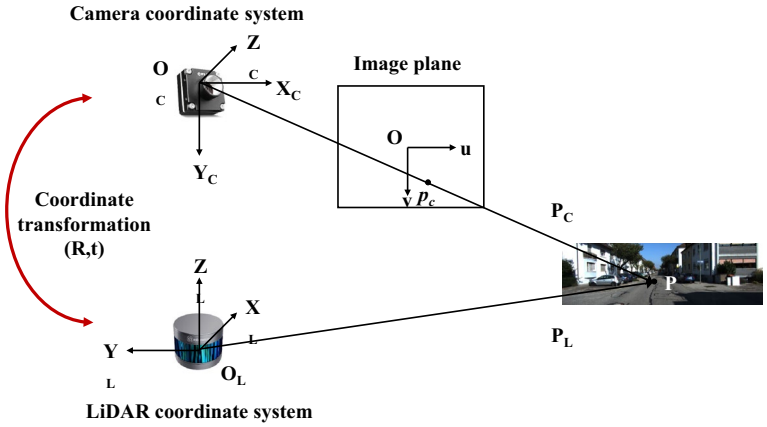


Fig. 3 Transformation between a LiDAR sensor and a camera using extrinsic parameters. A point P in 3D world scene is observed by a LiDAR sensor and a camera, denoted as P_L in the LiDAR coordinate system and P_C in the camera coordinate system. The coordinate transformation of these two points P_L and P_C is performed through extrinsic parameters $\mathbf{R}$ and $\mathbf{t}$

$$Z_C \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \mathbf{K}_C \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix}, \tag{1}$$

where f_x and f_y denote the focal length in pixels on the x and y axes respectively, (u_0, v_0) denotes the optical center (the principal point), and s denotes the skew coefficient, which is non-zero if the image axes are not perpendicular. Meanwhile, Z_C denotes the depth scale factor.

The transformation between the point P_L in the LiDAR coordinates and the point P_C in the camera coordinates, i.e., extrinsic parameters, is specified by the following equation:

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} + \mathbf{t} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} X_L \\ Y_L \\ Z_L \\ 1 \end{bmatrix}, \tag{2}$$

where $\mathbf{R}$ and $\mathbf{t}$ denote the rotation matrix and the translation vector between LiDAR and camera coordinates, respectively. Let $\mathbf{T} = [\mathbf{R}\ \mathbf{t}]$. In the following, we use $\mathbf{T}$ to denote the extrinsic parameters.

At last, we can specify the transformation between P_L and p_C by combining Eqs. (1) and (2):

$$Z_C \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}_C \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} X_L \\ Y_L \\ Z_L \\ 1 \end{bmatrix}. \tag{3}$$

In the following, we use $proj_{\mathbf{T}}(P_L) = p_C$ to denote the projection function from the 3D LiDAR point P_L to the 2D point p_C on the image plane w.r.t. the extrinsic parameters $\mathbf{T}$. With a slight abuse of the notion, we also use $proj_{\mathbf{T}}(\mathbf{P}_L) = \mathbf{p}_C$ to denote the projection

function from a set of 3D LiDAR points $\mathbf{P}_L$ to an image $\mathbf{p}_C$, where for each $P_L \in \mathbf{P}_L$, $proj_{\mathbf{T}}(P_L) \in \mathbf{p}_C$, and vice versa.

2.2 Categories of extrinsic calibration methods

According to the need for calibration targets and whether human intervention is required, extrinsic calibration between a LiDAR sensor and a camera can be divided into the following four categories:

Manual target-based These extrinsic calibration methods require engineers to manually specify the correspondences between the LiDAR point clouds and camera images based on one or more calibration targets, like checkerboard patterns (Zhang and Pless 2004; Geiger et al. 2012; Zhou and Deng 2012), ArUco tags (Dhall et al. 2017; Yoo et al. 2018), custom-made planar targets (Vel'as et al. 2014; Guindel et al. 2017), ordinary boxes (Pusztai and Hajder 2017; Hassanein et al. 2016).

These specified calibration targets impose geometric constraints between corresponding 3D points in point clouds and pixels in images, which enable the agent to estimate extrinsic parameters. For example, Zhang and Pless (2004) proposed to use a checkerboard from multiple views to calibrate a 2D LiDAR sensor and a camera, where the extrinsic parameters were estimated by solving a nonlinear least-squares iterative minimization problem. Later, Unnikrishnan and Hebert (2005) extended the work to calibrate 3D LiDARs and cameras following a similar procedure.

Automatic target-based Different from manual target-based methods, these methods do not require human intervention, where the correspondences between point clouds and images are automatically estimated using various features w.r.t. the calibration targets.

There are various calibration methods in this category. For instance, Geiger et al. (2012) presented an automatic extrinsic calibration method using a single shot only. In specific, the method required finding several checkerboards in different places, other than taking several shots on one checkerboard located differently. Toth et al. (2020) uses a spherical target for automatic extrinsic calibration. The method calculates the sphere center of the target using the detected surfaces and contour from point clouds and images respectively, and estimates extrinsic parameters via the geometric constraint for the same sphere center.

Manual targetless Extrinsic parameters may need to be adjusted online in some real-world applications, like self-driving (Levinson and Thrun 2013). Then targetless calibration methods are required to estimate extrinsic parameters in the real world without specified targets.

Manual targetless methods consider the problem by manually specifying the correspondences between point clouds and images, which often require a set of predefined rules or patterns for selecting the correspondences. For example, Scaramuzza et al. (2007) proposed a targetless calibration method. The method first manually selects a set of pairs between 3D points in point clouds and pixels in images. Then it estimates extrinsic parameters using the PnP (Perspective from n Points) algorithm (Quan and Lan 1999) followed by an iterative least-squares refinement.

Automatic targetless Automatic targetless calibration methods estimate extrinsic parameters by exploiting useful information from surrounding environments automatically. These approaches neither require any specified calibration targets nor heavy manual work. In the next section, we summarize various existing automatic targetless extrinsic calibration methods according to which information they are used for the estimation.

Notice that, automatic targetless calibration is widely applied for lots of practical applications for autonomous systems, like intelligent vehicles, drones, and robots (Li et al. 2017;

Liu et al. 2018). There exist multiple surveys for LiDAR–camera calibration. Nie et al. (2021) categorized calibration methods into offline and online methods. Yaopeng et al. (2021) divided them into manual calibration and automatic calibration solutions. Later, Khurana and Nagla (2021) classified existing calibration methods based on: (1) 2D or 3D LiDAR, (2) target-based or targetless, and (3) manual or automatic. Wang et al. (2021) Considered the need for targets and geometric constraints as transverse and longitudinal dimensions to classify the calibration methods. Different from these previous works, we focus on automatic targetless extrinsic LiDAR–camera calibration methods and consider their potential applications in various autonomous systems.

3 Automatic targetless LiDAR–camera calibration

Automatic targetless LiDAR–camera calibration methods intend to estimate the extrinsic parameters between LiDAR and camera automatically, by exploiting useful information from surrounding environments online, without any human intervention.

According to three specific sources of information exploited from environments, there are three categories of automatic targetless LiDAR–camera calibration methods, i.e., information theory based methods, feature based methods, and ego-motion based methods. Different from them, learning based methods use neural networks to implicitly capture useful information from environments for the calibration.

The reason that we group existing automatic targetless LiDAR-camera calibration methods in these four categories, is to indicate how the information from surrounding environment is utilized for calibration.Moreover, these four categories also suggest calibration methods for different application scenarios.

In specific, information theory based methods are preferred for environments with few features, as they are to maximize the similarity between (the projection of) the set of all 3D points from the LiDAR and the whole image from the camera, rather than certain kinds of features.On the other hand, feature based methods are suitable for scenes that provide sufficient features such as urban environments with rich geometric and semantic features.

Ego-motion based methods can be performed for scenarios when both LiDAR and camera are moving during the calibration process, like the case that both sensors are mounted on a moving car.Correspondingly, ego-motion based methods should not be applied for scenarios when both sensors are static during the calibration process, like the applications of roadside sensing systems.Different from the above three categories, the applications of learning based methods require large sets of training data and enough computing resources for online inference.

In this section, we summarize most recent automatic targetless calibration methods into four categories, i.e., information theory based methods, feature based methods, ego-motion based methods, and learning based methods. For each category, we introduce the basic principles of the methods and further explore their differences by specifying multiple choices for the implementation.

3.1 Information theory based methods

Information theory based methods estimate the extrinsic parameters by maximizing the similarity transformation between the LiDAR sensor and the camera, which is measured by various information metrics. In specific, the basic principles of information theory based methods can be summarized as the following equation:

$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmax}}\, \mathrm{IM}(proj_{\mathbf{T}}(\mathbf{P}_L),\, \mathbf{p}_C\,), \tag{4}$$

where $\mathbf{P}_L$ denotes the set of 3D points generated by the LiDAR sensor, $\mathbf{p}_C$ denotes the image generated by the camera, $proj_{\mathbf{T}}$ denotes the project function from the set of 3D points to the image w.r.t. the extrinsic parameters $\mathbf{T}$, and IM denotes the corresponding information metric that measures the similarity between $proj_{\mathbf{T}}(\mathbf{P}_L)$ and $\mathbf{p}_C$.

Following the statement in Eq. (4), an information theory based method for LiDAR–camera calibration consists of three steps:

3D–2D projection for LiDAR points $proj_{\mathbf{T}}$ projects the set $\mathbf{P}_L$ of 3D LiDAR points to the image $proj_{\mathbf{T}}(\mathbf{P}_L)$ w.r.t. the extrinsic parameters $\mathbf{T}$.

Statistical similarity measure IM measures the statistical similarity between the 2D projected image $proj_{\mathbf{T}}(\mathbf{P}_L)$ and the camera image $\mathbf{p}_C$ w.r.t. some features that share the similar distribution between the sensor data obtained by LiDAR and camera. Notice that, different choices of these features and corresponding statistical dependence measures would result in different LiDAR–camera calibration methods.

Optimization The statistical dependence measure IM is usually a non-convex function, which requires an optimization method to reach global optima. A typical pipeline of an information theory based approach is shown in Fig.4.

Note that, there are several attributes of the sensor data obtained by LiDAR and camera that share a similar distribution. For instance, LiDAR data points with high reflectivity usually correspond to bright surfaces in the image, and points with low reflectivity correspond to dark areas (Pandey et al. 2012). The correlation between the LiDAR reflectivity and camera intensity is often applied to measure the similarity between the data of LiDAR and camera. Besides reflectivity and intensity, gradient magnitude and orientation extracted from both LiDAR points clouds and camera images can also be considered here (Taylor et al. 2013).

3.1.1 Pairs of point cloud and image attributes

We summarize pairs of attributes for LiDAR point clouds and images that are commonly adopted in existing information theory based methods and specify them in the form of "Point cloud attribute – Image attribute".

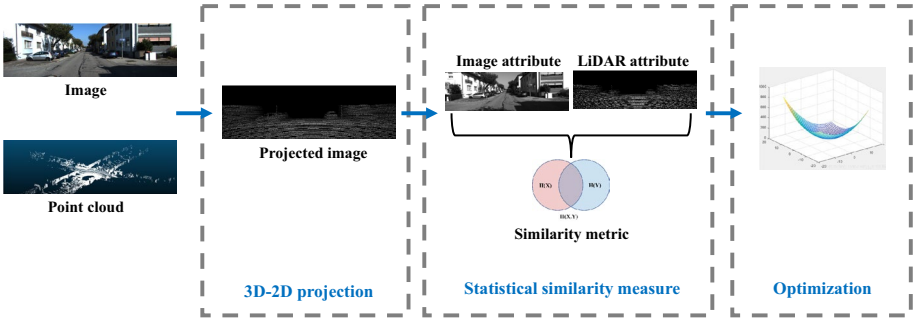


Fig. 4 A typical pipeline of information theory based Methods. In this figure, the LiDAR–camera attribute pair is chosen as reflectivity—grayscale intensity, and the statistical similarity measure is chosen as MI

- *Reflectivity—Grayscale intensity* The reflectivity of a LiDAR point is recorded as the return strength of a laser beam, and grayscale intensity denotes the intensity of the pixel in a grayscale image. When the camera and LiDAR simultaneously observe the environment, there would be a statistical similarity between the reflectivity of the LiDAR point clouds and the grayscale intensity of the image, as both attributes mainly depend on the same surface property of the objects (Pandey et al. 2014). Similarly, other pairs of attributes, like *Reflectivity—Hue* (Zhao et al. 2016), *Reflectivity—Visible light wavelengths* (Pascoe et al. 2015) and *Reflectivity—color* (Irie et al. 2016), also depend on the same surface property of the objects.
- *Surface normal—Grayscale intensity* Given the light sources in the environment, the surface normal will affect the grayscale intensity of the corresponding pixels in the image. Then, there is a statistical relation between the surface normal obtained from the LiDAR point clouds and the grayscale intensity of the image. The surface normal can be estimated from either dense or sparse LiDAR point clouds via various methods (Taylor and Nieto 2012). Given the normal vector of a point, the corresponding angle between the horizontal plane can also be calculated (Taylor and Nieto 2013). It often assumes that most of the light is coming from above, then this angle results in the largest influence on the intensity, which implies the statistical relation between the surface normal and the grayscale intensity.
- *Gradient magnitude and orientation—Gradient magnitude and orientation* When comparing two multi-modal images, a camera picture and a LiDAR depth image for example, if the pixel intensity of a patch in one image differs significantly from its surroundings, then the strength of the corresponding site in the other modality is likely to change accordingly (Taylor et al. 2013). This correlation exists as changes in these intensities typically represent differences between the background and the detected material or object. For 2D images, the magnitude and orientation of its pixel gradient can be calculated using the Sobel operator (Taylor et al. 2014). As for point clouds, each pixel is first projected onto a sphere, then the gradient is computed using its nearest 8 neighbors based on the algorithm proposed in Taylor et al. (2014).
- *3D semantic label–2D semantic label* Due to the fact that the semantic label of each 3D point is the same as its corresponding image pixel if exists, we should be able to perform data association using such information (Jiang et al. 2021). The point-wise semantic labels in an image and a point cloud can be predicted respectively, in a segmentation task using neural network models (Takikawa et al. 2019; Cortinhal et al. 2020).
- *Combination of 3D–2D attribute-pairs* Instead of relying on one specific pair of 3D–2D attributes to estimate the pixel similarity, some methods found that using a mixture of features is advantageous for improving algorithmic robustness against varying environments (Irie et al. 2016). They compute similarity measurements using a combined set of 3D–2D attribute pairs with appropriate weights assigned to each. These attribute sets are usually a combination of some of the above attribute pairs, such as reflectivity, surface normal, and gradient in the point cloud, and grayscale intensity and gradient in the image. They compute similarity measurements with a combined set of 3D–2D attribute pairs and assign them appropriate weights.

3.1.2 Statistical similarity measure

Based on the above attribute pairs for LiDAR point clouds and camera images, we can use various statistical dependence measures to measure the statistical similarity between

them, where larger measure values lead to better correspondences. In the following, we summarize statistical dependence measures that are commonly applied in existing information theory based methods.

- *Mutual Information (MI)* MI provides a means to measure statistical dependence between two random variables or the amount of information that one variable contains about the other. Under the Shannon entropy (Shannon 2001), MI is defined as:

$$MI(X, Y) = H(X) + H(Y) - H(X, Y),$$

where $H(X)$ and $H(Y)$ are the individual entropies of random variables X and Y, and $H(X, Y)$ is the joint entropy of the two random variables, i.e.,

$$H(X) = -\sum_{x \in X} p_X(x) \log p_X(x),$$
$$H(Y) = -\sum_{y \in Y} p_Y(y) \log p_Y(y),$$
$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p_{XY}(x, y) \log p_{XY}(x, y),$$

where $p_X(x), p_Y(y), p_{XY}(x, y)$ denote the marginal and joint probabilities of these random variables, respectively. In practice, we can use, for example, the reflectivity value of each LiDAR point and the intensity of each image pixel as two random variables X and Y. Then the probability distribution of both random variables can be estimated using methods, like kernel density estimation (KDE) Scott (1992).

- *Normalized Mutual Information (NMI)* Notice that, MI can be influenced by the total amount of information contained in both LiDAR points and the image. Then the preferred similarity transformations between the LiDAR sensor and the camera, i.e., the extrinsic parameters for the calibration, may not result in larger MI measure values (Studholme et al. Jan 1999). NMI addresses the problem by normalizing the value in MI, i.e.,

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

- *Gradient Orientation Measure (GOM)* GOM operates by calculating how well the orientation of the gradients is aligned between two images (Taylor et al. 2013). The magnitude of the gradient is also considered as the weight. There is a major difference between NMI and GOM. GOM uses the gradients of points rather than their intensity, so it takes into account the values of neighboring points and the geometry present in the image.
- *Normalised Information Distance (NID)* NID (Li et al. 2004) is a similarity metric that can be used to match the modalities of different sensors. The normalization property of NID brings similar advantages over MI metrics, as it does not depend on the total information content of the two images, thus, it does not detrimental to global image alignment due to matching between highly textured image regions.
- *Bagged Least-squares Mutual Information (BLSMI)* BLSMI (Irie et al. 2016) is a combination of methods composed of a kernel-based dependence estimator and noise reduction by bootstrap aggregating (bagging). One of the advantages of

BLSMI over ordinary MI is that BLSMI is robust against outliers because it does not include a logarithm.

- *Mutual Information and Distance between Histogram of Oriented Gradients (MIDHOG)* MIDHOG is a metric that combines NMI and Distance between Histogram of Oriented Gradients (DHOG) to measure the consistency between images (Guislain et al. 2017). MIDHOG is defined as a parameter representing the weight α:

$$MIDHOG = (2.0 - NMI) + \alpha \cdot DHOG.$$

When applied to images with only a few textures, DHOG performs much better than NMI. However, on images with a lot of textures, NMI gives more accurate results. Thus, MIDHOG is able to deal with different scenarios by inheriting the properties of MI and DHOG.

- *Mutual Information Neural Estimation (MINE)* MINE (Belghazi et al. 2018) use neural networks to estimate the mutual information between high dimensional continuous random variables. MINE is scalable, flexible, and completely trainable via back-propm, and it can be used in mutual information estimation, maximization, and minimization. MINE uses the Donsker-Varadhan (DV) duality to represent MI as:

$$I(X, Y) = \sup_{\theta \in \Theta} \mathbb{E}_{P(X,Y)}[F_\theta] - \log(\mathbb{E}_{P(X)P(Y)}[e^{F_\theta}]).$$

F_θ is a function parameterized by a neural network, where θ are the parameters of the neural network.

3.1.3 Optimization methods

We also summarize optimization methods that are most commonly adopted in existing information theory based methods.

- *Barzilai–Borwein steepest descent method* The Barzilai–Borwein steepest descent method (Barzilai and Borwein 1988) is a gradient method with an adaptive step size in the direction of the gradient of the cost function.
- *Nelder–Mead downhill simplex method* The Nelder–Mead method (Nelder and Mead 1965) is a direct search method and is often applied to nonlinear optimization problems for which derivatives may not be known.
- *Levenberg–Marquardt algorithm* The Levenberg–Marquardt algorithm (Levenberg 1944) is a commonly used iterative algorithm to solve non-linear minimization problems.
- *Particle swarm optimization* Particle swarm optimization (Kennedy and Eberhart 1995) is a global optimization algorithm. It works by placing an initial population of particles randomly in the search space, then iteratively optimizing to solve the problem.
- *Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton method* The BFGS quasi-Newton method (Kelley 1999) is a gradient-based algorithm to maximize the objective function.
- *The Bound Optimization BY Quadratic Approximation (BOBYQA) algorithm* The BOBYQA algorithm (Powell 2009) is a deterministic, derivative-free optimization algorithm that relies on an iteratively constructed quadratic approximation.

3.1.4 Summary of information theory based methods

Following the above discussion, we summarize information theory based methods in Table 1 and group the methods by corresponding 'Information metric', 'LiDAR attribute – Image attribute', and 'Optimization method'.

When using combined 3D–2D attribute pairs, the specific attribute pairs are selected differently for each method. Taylor and Nieto (2013) and Guislain et al. (2017) both choose 'reflectivity - grayscale intensity' and 'surface normal - grayscale intensity' the two attribute pairs.

Irie et al. (2016) also used 'depth discontinuity – edge' attribute pair. The assumption of this method is that depth changes in the point cloud are likely to appear as edges in the image, which will be described in detail later in the feature based method section. Zhao et al. (2016) used reflectivity, surface normal, and curvature as LiDAR attributes and intensity, hue, and gradient as image attributes. Here, the curvature attribute in the point cloud is used to correspond to the gradient attribute in the image. Besides reflectivity and surface normal for the point cloud and color for the image, the curvature attribute in the point cloud corresponds to the gradient attribute in the image.

3.2 Feature based methods

Different from information theory based methods, feature based methods for automatic targetless LiDAR–camera calibration directly extract and match the features from LiDAR and camera images, without optimizing their statistical similarities.

Features that are commonly adopted in these methods can be sorted into three categories, including geometric, semantic and motion features. They need to be acquired online from both LiDAR points and camera images of surrounding environments. In specific, geometric features are constructed by a set of geometric elements like points or edges in environments. Semantic features are high-level representations that often specify semantic-aware components of the environment, such as skylines, cars, and poles. Motion features describe the characteristics of moving objects, including pose, velocity, acceleration, etc.

As illustrated in Fig. 5, the process of a typical feature based method often contains three steps, i.e., feature extraction, feature matching, and transformation estimation.

Feature extraction Feature extraction aims to automatically detect stable and unique features from both point clouds and images. These features usually represent specific geometric or semantic elements in the surrounding environments.

Feature matching Feature matching intends to provide the correspondence between the features extracted from the point cloud and the image. For this purpose, various feature descriptors as well as spatial relationships between features are applied.

Transformation estimation This step estimates the transformation relationship, i.e., extrinsic parameters, for LiDAR–camera calibration, based on feature correspondences provided by feature matching. Singular Value Decomposition (SVD) is a widely applied algorithm for the step.

Meanwhile, many methods also combine the steps of feature matching and transformation estimation (Levinson and Thrun 2013; Li et al. 2017; Zhu et al. 2020). They estimate the transformation relationship while looking for the feature correspondences.

In the following, we summarize typical features and corresponding extraction methods for feature extraction and commonly applied strategies for feature matching. Then we

Table 1 Summary of information theory based methods

Method	Information metric	LiDAR attribute—Image attribute	Optimization method	Method complexity	Application environment	Open Source
Pandey et al. (2012, 2014)	MI	Reflectivity—Grayscale intensity	Barzilai-Borwein steepest descent method	+++	Urban	No
Wang et al. (2012)	MI	Reflectivity—Grayscale intensity	Nelder-Mead downhill simplex method	+++	Urban	No
Miled et al. (2016)	MI	Reflectivity—Grayscale intensity	Levenberg-Marquardt algorithm	+++	Urban	No
Taylor and Nieto (2012)	NMI	Surface normal—Grayscale intensity	Particle swarm optimization	++	Natural	No
Taylor and Nieto (2013)	NMI	Combination of 3D–2D attribute-pairs i.e., Reflectivity—Grayscale intensity, Surface normal—Grayscale intensity	Particle swarm optimization	++	Natural	No
Zhao et al. (2016)	NMI	Combination of 3D–2D attribute-pairs i.e., Reflectivity—Grayscale intensity, Reflectivity—Hue, Surface normal—0Grayscale intensity, Curvature—Gradient magnitude and orientation	Nelder-Mead downhill simplex method	+++	Urban	No
Igelbrink et al. (2018)	NMI	Reflectivity—Grayscale intensity	Nelder-Mead downhill simplex method	+++	Natural	No
Taylor et al. (2013, 2014)	GOM	Gradient magnitude and orientation—Gradient magnitude and orientation	Particle swarm optimization	++	Urban	Yes
Pascoe et al. (2015)	NID	Reflectivity—Visible light wavelengths	BFGS quasi-Newton method	++	Urban	No
Irie et al. (2016)	BLSMI	Combination of 3D–2D attribute-pairs i,e., Reflectivity—Grayscale intensity, Surface normal—Grayscale intensity, Depth discontinuity—Edge	BFGS quasi-Newton method	+++	Indoor/ Urban	No

Table 1 (continued)

Method	Information metric	LiDAR attribute—Image attribute	Optimization method	Method complexity	Application environment	Open Source
Guislain et al. (2017)	MIDHOG	Combination of 3D–2D attribute-pairs i.e., Reflectivity—Grayscale intensity, Surface normal—Grayscale intensity	BOBYQA algorithm	++	Urban	No
Jiang et al. (2021)	MINE	3D semantic label—2D semantic label	Gradient descent method	+++++	Natural/ Urban	No

We list popular information theory methods in terms of information metrics, chosen attributes of point clouds and images, and optimization methods. In addition, we also list the complexity of the method, application environment, and whether it is open source

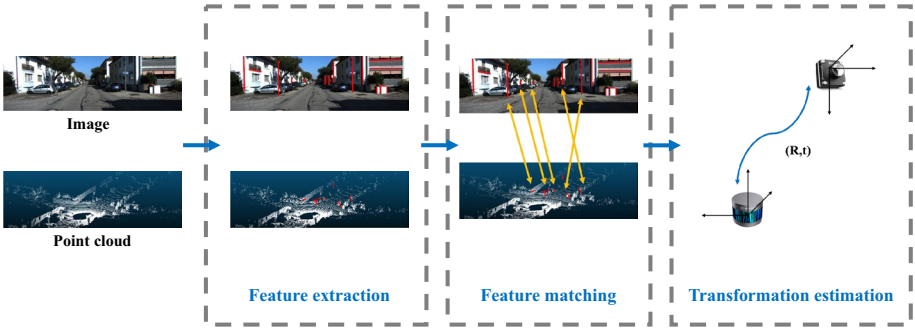


Fig. 5 A typical pipeline of feature based methods. Here we use the extraction and matching of line features as an example

summarize feature based methods for automatic targetless LiDAR–camera calibration in the literature.

3.2.1 Feature extraction

In the early stage, features in camera images and point clouds are specified by hand (Scaramuzza et al. 2007), which are often used in some manual methods for the LiDAR–camera calibration problem. With the development of computer vision and the requirement for automatic matching, many feature detection methods have been developed to extract unique and robust features from both images and point clouds.

There are a number of feature detectors for point clouds and images, respectively. In LiDAR–camera calibration, we need a pair of feature detectors for both point clouds and images. In the following, we summarize the pairs that are commonly employed in existing feature based methods and address them in the form of "point cloud feature extractor

– image feature extractor". We collect these pairs into three categories, i.e., both features in the pair are geometric features, semantic features, and motion features, respectively.

We first summarize the pairs of feature detectors for geometric features, where points of interest and edges are widely applied.

Points of interest are geometric features that are widely applied in LiDAR–camera calibration. A point of interest may have a special attribute that can be significantly different from its neighbors, such as color or brightness. It may also have an explicit location in the image space, e.g., intersection points of geographic edges (Willis and Sui 2009). Point features can be calculated regularly and reliably to provide effective detection results.

- *Förstner operator–Förstner operator* Förstner operator (Förstner and Gülch 1987) is a fast operator for detecting and precisely locating distinct points and corners. The algorithm extracts junction and circular points from a image with subpixel accuracy. Interest points extraction in the LiDAR point cloud can be performed by projecting the points into a range image and applying the Förstner operator (González-Aguilera et al. 2009).
- *Corner–Corner* Defined as the intersection of horizontal and vertical edges, corners can be naturally found in the data of urban scenes. These edges can be detected by the edge detectors such as Sobel operator (Sobel et al. n.d.). Similar to the above case for Förstner operator, the corner feature extraction can be performed on both the camera image and the projected intensity image from the point cloud.
- *SIFT–SIFT* Scale Invariant Feature Transform (SIFT) (Lowe 1999) is a popular operator that detects and matches local features in images. With SIFT, the extracted point features are invariant to image translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projection. There are several variants of SIFT available for point feature extraction. For instance, Speeded Up Robust Features (SURF) is a faster version of SIFT (Bay et al. 2006), Affine SIFT (ASIFT) (Morel and Yu 2009) extends the SIFT method to fully affine invariant. For images and point clouds, interest point extraction based on SIFT or its variants is performed on both pristine and projected images.

Besides points of interest, edges are another type of geometric feature that is widely applied in LiDAR–camera calibration. These edges in point clouds and images contain geometric information of environments that are useful, especially for environments when point features disclose their instability (Yu et al. 2020).

- *Depth discontinuity—Intensity difference* Edges in LiDAR point clouds can be extracted using depth discontinuities. In specific, these edges are recognized from the points by calculating the differences in depth between neighboring points and filtering out points whose values of differences are below a pre-set threshold (Levinson and Thrun 2013). This idea has been widely applied in various edge extraction methods (Blaga and Nedevschi 2017; Banerjee et al. 2018; Munoz-Banon et al. 2020; Ma et al. 2021; Wang et al. 2018; Xu et al. 2019). The idea can be further extended by first generating a dense depth map by upsampling the point cloud, then identifying the edges by calculating gradient changes in depth (Castorena et al. 2016). Meanwhile, edges in images can be extracted by detecting shape changes in pixel intensity. It is often assumed that edges extracted by depth discontinuity in point clouds are one-to-one corresponded to edges extracted by intensity difference in images.

- *Depth discontinuity—Sobel operator* Edges in LiDAR point clouds are still extracted by depth discontinuity. Meanwhile, edges in images are extracted by performing Sobel operator (Sobel et al. n.d.), which is an operator that detects edges based on changes of image grayscales. In specific, Sobel operator combines Gaussian smoothing and differentiation to compute the approximation of the gradient of the image intensity function. Besides Soble operator, Canny edge detector (CANNY 1987) and LSD algorithm (von Gioi et al. 2012) also provide methods to extract edges in images. In particular, Canny edge detector uses a multi-stage algorithm to detect a wide range of edges in images, which involves steps of noise reduction, intensity gradient estimation, non-maximum suppression, and hysteresis thresholding. On the other hand, LSD is an edge detection algorithm based on the gradient of the grayscale image. Therefore, *Depth discontinuities—Canny detector* and *Depth discontinuities—LSD* are also possible pairs for feature based methods.
- *3D line detector—LSD* 3D line detector (Yu et al. 2020) provides an alternative way for extracting edges in point clouds. In specific, based on the point cloud segmentation and a 2D line detector (Lu et al. 2019), a 3D line detector utilizes a simple 3D point cloud segment detection algorithm for structured environments. Meanwhile, edges in images are extracted by LSD.
- *Depth continuity—Canny detector* We can extract two kinds of edges in point clouds, i.e., edges with depth discontinuity and edges with depth continuity. In specific, depth-discontinuous edges are those whose depth values changed dramatically w.r.t. their neighboring points, which often refer to edges between foreground and background objects. In contrast, depth-continuous edges are those with continuously varying depth values, which tend to suggest planar intersection lines. These depth-continuous edges can be extracted from a dense point cloud, like the one generated by a solid-state LiDAR. In particular, these edges, i.e., plane intersection lines, can be extracted using point cloud voxel partitioning and plane fitting, which divides the point cloud into small voxels of given sizes and repeatedly uses RANSAC to fit and extract planes in these voxels (Yuan et al. 2021). Meanwhile, edges in images can be extracted by Canny detector.
- *Depth continuity—L-CNN*Bai et al. (2020) reports that the edges of buildings often have sharp edges and explicit line textures, which can be easily extracted in both point clouds and images. In specific, the planes of corresponding buildings can be identified by various point cloud segmentation methods (Nurunnabi et al. 2012; Vo et al. 2015; Xu et al. 2015). Then the edges, i.e., plane intersection lines, in the point cloud can be conveniently obtained by a line detection algorithm based on these segmented 3D planes. On the other hand, an end-to-end neural model, named L-CNN (Zhou et al. 2019), can be trained to output a vectorized wireframe that contains semantically significant and geometrically salient lines and junctions.

Notice that, real-world environments often contain a large number of similar geometric features, which would increase the difficulty of LiDAR–camera calibration. On the other hand, semantic features often reflect high-level characteristics that account for semantic-aware constraints of the environments, such as skyline (Hofmann et al. 2014), vehicles (Zhu et al. 2020), and road lanes (Ma et al. 2021). This semantic information is consistent across data modalities. Then they can be extracted from both LiDAR point clouds and camera images and used for LiDAR–camera calibration.

- *Skyline–Skyline* A skyline is a curve or contour between the sky and other objects in urban environments. This semantic feature is evident in both LiDAR point clouds and images and can be extracted for calibration. In the point cloud, the skyline can be obtained from the contour plot involving the foreground and the sky, since LiDAR sensors only get distance measurements for objects and receive no response from the open sky (Hofmann et al. 2014). Other methods first generate a projected image of the point cloud, then identify the highest pixel in the image from the bottom in a column-wise fashion. Once such a pixel is found, the corresponding point is considered to be on the skyline (Zhu et al. 2018). The skyline in an image can be determined from all world objects based on a given brightness threshold and alpha shapes (Edelsbrunner et al. 1983). Alternatively, based on the large difference in image pixel values between the sky and other objects, along with some prior information on the skyline's location in the image, the desired skyline points can be retrieved by performing a column-wise search for the first pixel point with a jump in grayscale values from top to bottom.
- *Lane and Pole–Lane and Pole* Lanes and poles are objects with distinct linear shapes in the images and point clouds. Most road lanes are outlined with high reflectivity paint to enhance visibility in the dark, which gives the LiDAR a stronger signal. Lanes on the road can be extracted by choosing a threshold for the pixel intensity (Ma et al. 2021). Meanwhile, the road poles in the point cloud can be extracted using their obvious feature of being perpendicular to the ground, which is often calculated by setting a height threshold within the field of the image. BiSeNet-V2 (Yu et al. 2018) has been used as an image semantic segmentation network for such tasks, which is further refined by improving the contours using a Dense CRF operator (Krähenbühl and Koltun 2011).
- *3D Semantic Centroid–2D Semantic Centroid* Recent convolutional neural networks (CNN) based methods have made make remarkable progress and largely improved the performance of semantic segmentation tasks (Liu et al. 2018). Multiple methods have applied such networks to extract semantic information for LiDAR–camera calibration. Wang et al. (2020) proposed such a calibration method, where PointR-CNN (Shi et al. 2019) and Nvidia Semantic Segmentation (Zhu et al. 2019) were applied respectively to obtain the semantic information from the point cloud and the image. Later, the 3D and 2D Semantic Centroids are calculated based on these segmentation results.

Besides geometric and semantic features for static elements in environments, motion features such as trajectories of moving objects can also be used to calibrate multiple sensors.

- *Object trajectory–Object trajectory* Based on multiple detection and tracking algorithms, we can receive estimated trajectories of moving objects from LiDAR and camera respectively. Notice that, we can obtain two trajectories for a moving object from LiDAR and camera respectively. These two trajectories should be as closely matched as possible, which helps us to calibrate the LiDAR and camera (Peršić et al. 2020).

3.2.2 Feature matching strategies

Feature matching intends to establish the correspondences between the points in LiDAR point clouds and the pixels in images, which are identified by feature extraction. Here we summarise popular feature matching strategies, which are specified for certain kinds of features by considering descriptor similarities and spatial geometric relationships, respectively.

- *Descriptors similarity* Descriptor similarity based matching methods are usually applied for geometric features focusing on points of interest. Based on extracted feature points, a description, i.e., a compact representation of the neighborhood of the points, is often used to compute a descriptor for each feature point. This strategy matches the feature points with the most similar descriptors between the image and the projected image from the point cloud. Brute force matching calculates the similarity of the matched features w.r.t. the reference feature set. On the other hand, the nearest neighbor fast search method can alleviate the problem. Meanwhile, Euclidean distance is often used as the distance metric. Since there are many incorrectly matched points in established correspondences, it is usually necessary to apply a random sampling consistent random optimization algorithm (RANSAC) to eliminate these incorrectly matched point pairs (González-Aguilera et al. 2009; Li-Chee-Ming et al. 2010).
- *Spatial geometrical relation* This kind of feature matching strategy aims to establish the correspondences from two given feature sets by directly using spatial geometrical relations and optimization methods. They align features, such as line features distributed at different locations in space, as much as possible and assume that the calibration parameters reach the optimal solution when these features are perfectly aligned (Levinson and Thrun 2013).
- *Semantic relation* The semantic relation based matching strategies intend to find the correspondences by matching the features at the semantic level as much as possible. For example, the points reflecting the vehicle in the point cloud can be matched to the pixels for the vehicle in the image (Wang et al. 2020; Zhu et al. 2020).
- *Trajectory relation* The basic method for trajectory association is based on matching the locations in both trajectories with the same time stamp. Moreover, velocity and curvature can also be used for matching these trajectories. Peršić et al. (2020) determine the trajectory association by observing two criteria: the average of the velocity norm difference and the average of the position norm difference. The track pairs are required to satisfy these two criteria and not exceed a predefined threshold.

3.2.3 Summary of feature based methods

In Table 2, we summarize feature based methods for LiDAR–camera calibration and group them by categories of their features.

Besides the methods listed in Table 2, there exist multiple methods (Bileschi 2009; Banerjee et al. 2018; Blaga and Nedevschi 2017; Xiao et al. 2017; Li et al. 2017; Jiang et al. 2018; Munoz-Banon et al. 2020; Ma et al. 2021) that extract line features in point clouds and images respectively using depth discontinuities with image edge detectors.

Besides only considering line features alone, there are also methods that use the combination of line features and depth information from both images and point clouds for calibration. These methods assume that the depth difference between the measured LiDAR data and the image should be minimized. The depth information from images can be obtained either from the point cloud projection (Castorena et al. 2016) or from the monocular depth estimation (Vaida and Nedevschi 2019).

For the methods that extract lines using depth continuity in the point cloud, the point cloud is often segmented into uniform size voxels in Yuan et al. (2021). Different from these methods, Liu et al. (2021) implements the adaptive voxelization to dynamically segment the LiDAR point cloud into voxels of different sizes.

Table 2 Summary of feature based methods

Method	Feature type	Feature extraction	Feature matching	Method complexity	Application environment	Open Source
González-Aguilera et al. (2009)	Point	Förstner operator–Förstner operator	Descriptors similarity	++	Natural	Yes
Li-Chee-Ming and Armenakis (2010)	Point	Corner–Corner	Descriptors similarity	++	Urban	No
Böhm and Becker (2007)	Point	SIFT–SIFT	Descriptors similarity	++	Urban	No
Zhang et al. (2015)	Point	SURF–SURF	Descriptors similarity	++	Urban	No
Alba et al. (2012)	Point	SIFT/SURF–SIFT/SURF	Descriptors similarity	++	Urban	No
Moussa et al. (2012)	Point	ASIFT–ASIFT	Descriptors similarity	++	Urban	No
Levinson and Thrun (2013)	Edge	Depth discontinuity—Intensity difference	Spatial geometrical relation	++	Urban	No
Li et al. (2017)	Edge	Depth discontinuity—Sobel operator	Spatial geometrical relation	++	Urban	No
Hsu et al. (2018)	Edge	Depth discontinuity—Canny detector	Spatial geometrical relation	++	Urban	No
Zhang et al. (2021)	Edge	Depth discontinuity—LSD	Spatial geometrical relation	++	Urban	No
Yu et al. (2020)	Edge	3D line detector—LSD	Spatial geometrical relation	++	Urban	No
Bai et al. (2020)	Edge	Depth continuity—L-CNN	Spatial geometrical relation	++	Urban	No
Yuan et al. (2021)	Edge	Depth continuity—Canny	Spatial geometrical relation	++	Urban	Yes
Hofmann et al. (2014)	Semantic	Skyline–Skyline	Semantic relation	+++	Urban	No
Wang et al. (2020)	Semantic	3D semantic information–2D semantic information	Semantic relation	+++++	Urban	No
Ma et al. (2021)	Semantic	Pole + road lane–Pole + road lane	Semantic relation	+++++	Urban	No
Peršić et al. (2020)	Motion	Object trajectory–Object trajectory	Trajectory relation	++++	Urban	No

We list popular feature based methods in terms of feature type, feature extraction of point clouds and images, and feature matching strategy. In addition, we also list the complexity of the method, application environment, and whether it is open source

As an alternative method for extracting feature points, Nieto et al. (2010) used the SIFT extractor automatically and matches the features by looking for the two closest features in the space of SIFT descriptors. As an alternative way to use semantic features, Zhu et al. (2020) applied semantic masks of vehicles in the image and constructs a height map to encourage LiDAR points to fall on the pixels labeled as vehicles. In this work, semantic segmentation is performed only in the image.

3.3 Ego-motion based methods

Ego-motion based methods exploit the motion of sensors mounted on the traveling vehicle to estimate the extrinsic parameter. In this scope, some methods try to find the correspondence between the trajectories generated by LiDARs and those by cameras, with LiDARs and visual odometry techniques, or IMU and GNSS measurements (Taylor and Nieto 2015; Ishikawa et al. 2018; Park et al. 2020). There are also methods that make use of the structure from motion (SfM) approach to estimate the 3D structure from the image sequences, thus converting the 3D–2D LiDAR–camera data registration into a 3D–3D case (Swart et al. 2011; Nagy et al. 2019a). In accordance with how the ego-motion information between sensors is used, ego-motion based methods can be divided into hand-eye based and 3D structure estimation based ones.

3.3.1 Hand-eye based methods

Hand-eye calibration problem is a fundamental and critical issue in robot vision applications. It is a problem in determining the transformation between a robot base and a camera, in the case where the camera (the "eye") is mounted on an arm (the "hand") of the robot, or fixed elsewhere other than the arm. The mathematical formulation of this problem also takes the form of $AX = XB$, where A and B describe the motions of the arm and the camera respectively, and X is the desired unknown transformation matrix. Methods discussed in this section extend the traditional hand-eye calibration to the LiDAR–camera calibration problem, although the rigid-mounted robot sensors should satisfy the same traditional constraints.

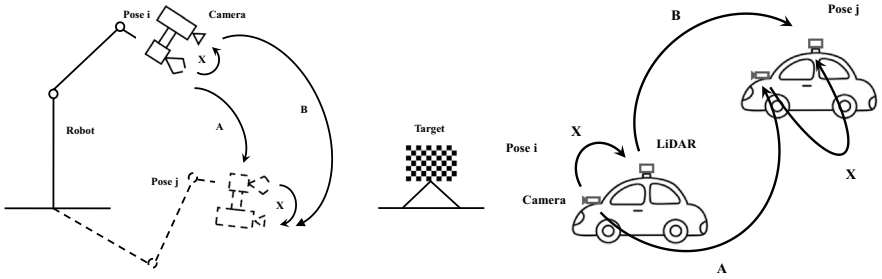
Given the following notation:

T : The transformation between a LiDAR sensor and a camera.
T_L^i : The motion or the transformation of the LiDAR sensor from timestamp t_i to timestamp t_{i+1}.
T_C^i : The motion of the camera from timestamp t_i to timestamp t_{i+1}.

Then the extrinsic parameter between a LiDAR sensor and a camera can be formulated by the hand-eye calibration:

$$T_C^i T = T T_L^i. \tag{5}$$

A depiction of the hand-eye calibration problem is shown in Fig. 6. Hand-eye based LiDAR–Camera calibration procedure can be roughly split into three stages:



(a) Standard hand–eye calibration

(b) Hand-eye based LiDAR-camera calibration

Fig. 6 **a** Standard hand-eye calibration problem. The camera "eye" is mounted on the robot gripper "hand", and the robot is performing a series of movements. The transformation between the camera and the gripper is calculated by solving the equation $AX = XB$. **b** LiDAR–camera calibration formulated as the hand-eye calibration problem. The two sensors are mounted on the vehicle. As the carrier vehicle moves, each sensor's motion is estimated. The extrinsic parameter between the two sensors is determined by the same equation above

Estimation of each sensor's motion In the first stage, the state transformation matrices for the LiDAR and the camera, i.e. T_L^i and T_C^i, are estimated with rotation and translation considered between neighboring frames for each sensor. For the LiDAR, Iterative Closest Point (ICP) and LiDAR odometry are popular algorithms to compute T_L^i (Taylor and Nieto 2014; Shi et al. 2019), while for the camera, SfM and visual odometry are commonly used methods to find T_C^i (Taylor and Nieto 2015; Park et al. 2020).

Estimation of the extrinsic parameter Since the motion of each sensor is estimated independently, the transformation between the LiDAR sensor and the camera can be obtained by solving the homogeneous equation defined by Eq. (5).

Solutions to the transformation equation can be categorized based on whether the rotation and translation parameters are estimated separately or simultaneously. In a hand-eye based extrinsic calibration problem, the separated solution is frequently used due to its simplicity (Taylor and Nieto 2015).

T_C^i, T_L^i, and T are 4×4 transformation matrices which can be written as:

$$T_C^i = \begin{bmatrix} R_C^i & t_C^i \\ 0 & 1 \end{bmatrix}, \quad T_L^i = \begin{bmatrix} R_L^i & t_L^i \\ 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} R_T & t_T \\ 0 & 1 \end{bmatrix}.$$

The matrix T can be divided into two parts: R^T and t^T. Thus, Eq. (5) yields the following two equations. First, the rotation R^T is determined by:

$$R_C^i R_T = R_T R_L^i. \tag{6}$$

Once R^T is known, Eq. (7) becomes linear and t^T can then be calculated by:

$$(R_C^i - I)t_T = R_T t_L^i - t_C^i. \tag{7}$$

Refinement of extrinsic parameter In hand-eye based methods, the external parameter is usually initialized by solving the homogeneous transform equation. However, deviations in the motion estimation can affect the calibration results and lead to inaccuracies (Taylor and Nieto 2015). The appearance information in the surroundings, such as geometric edge alignment, can be useful to reduce such errors. Liao and Liu (2019) utilizes the line

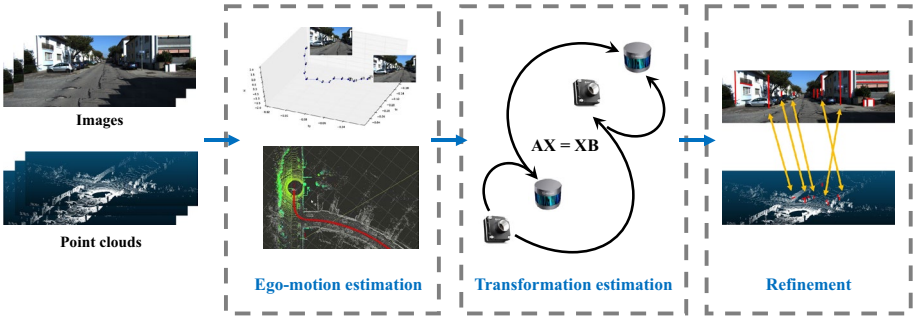


Fig. 7 The hand-eye based LiDAR–Camera calibration procedure can be roughly divided into three stages: the estimation of each sensor's ego-motion, the estimation of the transformation according to $AX = XB$, and the refinement of the estimated transformation. In the refinement stage, we use the line feature alignment method as an example

features in both the image and the point cloud to refine the calibration parameter by feature matching. A typical pipeline of a hand-eye based method is shown in Fig. 7.

To further discuss the above ego-motion based calibration pipeline, we focus on the selections of algorithms in each step. For the first motion-estimation step, we introduce the widely used LiDAR and camera motion estimation algorithms (Besl and McKay 1992; Zhang and Singh 2014; Mur-Artal et al. 2015); for the second equation-solving step, if the rotational parameter R_T is given, then Eq. (7) for the translational parameter t_T becomes linear that can be solved straightforwardly. Therefore, we focus on the way that R_T behaves in the solution. For the final calibration-refinement step, we present exactly what kinds of appearance information are involved.

To calculate the extrinsic parameter, we first estimate the pose of the LiDAR and the camera respectively for each paired data. Various methods are applied depending on the type of sensors. We summarize multiple popular methods for sensor motion estimation in hand-eye based calibration tasks.

- *LiDAR motion estimation* The iterative Closest Point (ICP) algorithm (Besl and McKay 1992) is a classical approach for point cloud registration. It iteratively queries the closest points between two sets of point clouds and minimizes the distance between the corresponding points. The output of ICP is a rigid transformation that associates the two point clouds. In addition, some variants of ICP have been developed for both point clouds and images (Oishi et al. 2005; Pomerleau et al. 2013). The motion of LiDAR sensors can also be estimated by LiDAR odometry methods (Shi et al. 2019; Park et al. 2020). For example, LOAM (Zhang and Singh 2014) is a simple and efficient 3D algorithm for this task, which matches the corresponding feature edges and planes. From each trajectory, LOAM extracts a set of relative transformations and utilizes them for extrinsic calibration.
- *Camera motion estimation* Using the SfM approach, a set of transformations that describe the movement of the camera can be calculated, up to scale ambiguity (Ullman 1979). Given 2-D images, SfM estimates the camera pose and retrieves a sparse reconstruction simultaneously. The camera motion transformations can also be found using a standard visual odometry approach, which estimate the motion of a camera in real time using sequential images (i.e., ego-motion). As an example, ORB-

SLAM (Mur-Artal et al. 2015) is a feature-based monocular simultaneous localization and mapping (SLAM) system that is frequently mentioned (Shi et al. 2019; Liao and Liu 2019). Note that, the motion estimation purely based on visual estimation faces the problem of scale ambiguity and requires the use of some additional methods to estimate the scale (Taylor and Nieto 2016; Ishikawa et al. 2018).

As mentioned above, Eq. (7) can be solved as a linear equation for translational transformation parameter t_T with known rotational parameter R_T. Here we focus mainly on different parameterization techniques for R_T, including rotation matrix, Angle-axis (Shiu and Ahmad 1989), Lie algebra (Park and Martin 1994) and Quaternions (Chou and Kamel 1991).

- *Rotation matrix* A rotation matrix is determined by a 3×3 matrix. Although not as compact as other representations, a matrix uniquely defines a 3D rotation. Park et al. (2020) found the rotation matrix of the equation by decomposing the covariance matrix of camera-LiDAR relative poses, after aligning the correspondences in the continuous-time trajectories of the sensors.
- *Angle-axis* The axis-angle representation parameterizes a rotation by two quantities: a unit vector, i.e., the rotation axis, pointing to the direction of the rotation, along with an angle indicating the magnitude of the rotation about this axis. In solving the homogeneous transformation equation for the rotation parameter, the use of an angle-axis representation can simplify the process (Taylor and Nieto 2016).
- *Lie algebra* The rotation parameters can also be expressed in the form of Lie algebras (Xu et al. 2019), which is suitable for optimization problems. Lie algebra specifies the extrinsic parameter through a vector with 6 degrees of freedom (DoF) variables. The 6 DoF parameters include a rotation vector $r = (r_1, r_2, r_3)$ and a translation vector $t = (x, y, z)$.
- *Quaternion* Quaternion provides a simple and unique representation for describing finite rotations in 3D space. Liao and Liu (2019) presented the rotation with quaternion which reduced the variable number from nine to four. Given the rotation, the translation parameter can be found by solving the linear equation (7).

Several methods based on environmental information have been reported useful for refining the calibrations between LiDARs and cameras, such as aligning edges (Levinson and Thrun 2013) or correlating the data intensity of the two modalities (Pandey et al. 2012). There are also methods to continuously optimize the estimation for camera motion and extrinsic parameters alternatively by sensor fusion odometry (Ishikawa et al. 2018).

- *Edge alignment* Line features in natural scenes can be used for optimizing the extrinsic parameter (Taylor and Nieto 2014; Liao and Liu 2019). The correspondence between 3D lines in point clouds and 2D lines in images can be derived from the line-to-line constraints, thus refining the results obtained from the motion estimation.
- *Intensity matching* An intensity alignment approach based on the statistical dependence measure can also be used to further refine extrinsic parameters. Shi et al. (2019) aligned the LiDAR reflectivity with the camera image intensity through the metric of mutual information. The hypothesis for this matching is that the LiDAR reflectivity is usually similar to the image intensity in the environment.

- *Depth matching* The correspondence between the depth images generated by LiDAR and the camera respectively, also adds to the optimization of the extrinsic parameters (Xu et al. 2019). The LiDAR depth map is created by projecting the LiDAR point cloud from the initial extrinsic parameter and the camera depth map is produced from the monocular depth estimation. The principle is that an arbitrary point in the LiDAR depth map should is bound to a pixel in the camera depth map at the same pixel coordinates and their depth values should be identical.
- *Color matching* This refinement method works by assuming that the points in the point cloud are of the same color as in the camera images in two consecutive frames (Taylor and Nieto 2016). It operates by first projecting points onto the image to obtain the corresponding colors of the local pixels, then the same points are projected onto the next frame of the image, with the time offset being compensated by the estimated motion information. By minimizing the average difference between the color of the points in current and previous frames, a more accurate extrinsic parameter can be obtained.
- *3D–2D point matching*Park et al. (2020) refined the extrinsic parameter by reducing the 3D–2D projection error. In their work, the 3D coordinates of 2D features are computed by triangulation instead of directly from 3D LiDAR points. After the 3D–2D projection is performed with LiDAR–camera extrinsic parameter, the result is improved using non-linear optimization.

We summarize hand-eye based methods for LiDAR–camera calibration in Table 3.

3.3.2 3D structure estimation based methods

Another way for LiDAR–camera calibration based on motion information is to estimate the 3D structure of the surrounding environment from images, one of the most commonly used methods is structure from motion (SfM) (Ullman 1979).

SfM is a technique to estimate the 3D structure of a scene from 2D image sequences, that has been applied in many occasions, such as 3D modeling, augmented reality, visual SLAM, etc. 3D structure estimation based approaches use SfM to generate 3D point clouds from a set of images recorded by the camera on the moving vehicle, which converts the LiDAR–camera calibration problem into a registration task in the 3D domain.

Swart et al. (2011) described an approach to register panoramic images and LiDAR point clouds. They generate a sparse 3D point cloud from images and match it to a dense 3D point cloud from LiDAR using a non-rigid ICP process. The results were then polished by adding SIFT interest points corresponding to the framework. Moussa et al. (2012) proposed a bundle block adjustment method to determine the accurate 3D–3D correspondences.

Corsini et al. (2012) divided the calibration into coarse and fine-grained alignment procedures. After obtaining the intermediate results by applying the ICP algorithm to the LiDAR generated point clouds, they use a global refinement method based on mutual information to improve the accuracy of the fine 2D–3D alignment.

Wang et al. (2018) used sequential scene information from the vehicle motion to obtain the initial extrinsic parameter. The method uses the SfM algorithm to calculate 3D points from 2D image sequences, and registers the SfM points with LiDAR points through the ICP algorithm to estimate the primary result. Then by projecting the 3D LiDAR points to the 2D image plane, they use feature points of edges with a combined optimization method to further promote the accuracy of the extrinsic parameter.

Table 3 Summary of hand-eye based methods

Method	Motion estimation	Rotation parameterization	Refinement Strategy	Method complexity	Application environment	Open Source
Taylor and Nieto (2014)	ICP–SfM	Angle-axis	Edge alignment	++++	Urban	No
Taylor and Nieto (2015)	ICP – SFM	Angle-axis	Color matching	++++	Urban	No
Taylor and Nieto (2016)	ICP – Visual odometry	Angle-axis	Color matching	++++	Urban	No
Ishikawa et al. (2018)	ICP – Visual odometry	Angle-axis	Intensity matching	++++	Indoor/Urban	No
Shi et al. (2019)	LiDAR odometry – Visual odometry	Angle-axis	Intensity matching	++++	Indoor/Urban	No
Liao and Liu (2019)	ICP – Visual odometry	Quaternion	Edge alignment	++++	Indoor	No
Xu et al. (2019)	ICP – Visual odometry	Lie algebra	Depth matching + Edge alignment	++++	Urban	No
Park et al. (2020)	LiDAR odometry – Visual odometry	Rotation matrix	3D–2D point matching	++++	Indoor/Urban	No

We list the differences between hand-eye based methods in terms of estimation method of motion trajectory, rotation parameterization, and refinement strategy, In addition, we also list the complexity of the method, application environment, and whether it is open source

However, the ICP algorithm may fail when the density of SfM cloud points is very different from the LiDAR ones. To address this challenge, Li et al. (2018) designed an automatic registration method based on semantic features extracted from panoramic images and point clouds. They use GPS and IMU aiding the SfM algorithm to obtain rotation parameters, then extract parked vehicles from two modalities to estimate translation parameters by maximizing the overlapping area of corresponding target pairs.

Nagy et al. (2019a) proposed an extrinsic calibration method with an object-level registration. First, they use SfM to generate point clouds from consecutive camera images that can be used for alignment and registration, then they introduce a target-level alignment between the generated and the LiDAR point clouds base on object detection results. Nagy et al. (2019b) introduced similar work and used semantic information in the point clouds registration stage.

Later, Nagy and Benedek (2020) made an extension to their previous work mainly in terms of optimization for the registration stage. They manage to diminish the registration error using the point-level ICP method after the object-level registration step, then they introduce a curve-based non-rigid point cloud registration refinement step build on the non-uniform rational basis spline approximation.

3.3.3 Other methods

Besides generating 3D point clouds through the motion trajectory of the sensors and recovering 3D structures from image sequences, there are alternative ideas of motion based methods to solve the LiDAR–camera calibration problem.

Bileschi (2009) made an early attempt to associate video streams with LiDAR data from a moving vehicle. The initial calibration parameter is obtained with the help of the IMU motion signal and then is refined by matching 2D and 3D contours in camera images and LiDAR point clouds.

Chien et al. (2016) developed a LiDAR–engaged visual odometry framework and embed the ego-motion estimation problem into LiDAR–camera calibration. Their idea is based on the idea that the performance of the estimated ego-motion is directly related to the quality of extrinsic parameters. Specifically, if the extrinsic parameter deviates far from the ground truth, then the ego-motion estimation would also lose effectiveness. Combining the ego-motion estimation problem with LiDAR–camera calibration will form a bi-level optimization structure, this method introduces data constraints such as intensity and discontinuity restrictions to solve such a problem.

Under the Gaussian noise assumption, Huang and Stachniss (2017) applied the Gauss-Helmert model to multi-sensor extrinsic calibration. With constraints between the motions of the individual sensors given, they jointly optimize the extrinsic parameter and reduce the pose observation error using the Gauss-Helmert paradigm.

Castorena et al. (2020) proposed a motion-guided method for automatic calibration of the two multi-modal sensors. With a sequence of time-synchronized point clouds from LiDAR and the corresponding images from the camera, they compute the motion vector for each modality independently, then estimated the extrinsic parameter.

When using sensor movement information for extrinsic calibration, the motion must satisfy constraints such as moving in all directions and rotating around all the axes. If the sensors are mounted on a mobile robot performing planar motions, some parameters are rendered as unobservable. Zuniga-Noel et al. (2019) estimated the extrinsic parameter of multiple heterogeneous sensors mounted on a mobile robot subjected to such movements. The method computes the 2D parameters (x, y, yaw) from sensors' incremental motions,

and used the observation of the ground plane to estimate the remaining 3 parameters ($z, pitch, roll$).

Horn et al. (2021) used dual quaternions (DQs) to represent translation and rotation with fewer parameters. Based on DQs, they confine the optimization to planar calibration only, and combine a fast local and a global optimization approach for estimating the result.

3.3.4 Summary of ego-motion based methods

Ego-motion based methods use the motion of sensors from LiDAR and camera data sequences. They do not require initial calibration parameters and overlapping field of view for hand-eye based methods and turn the 2D–3D registration problem into a 3D–3D registration problem for 3D structure estimation based methods. However, the accuracy of motion estimation for the sensors often affects the performance of these methods.

For SFM based methods 1. Accurate calibration results

3.4 Learning based methods

Recently, deep learning has made breakthroughs in automatic feature engineering, and achieved excellent performance on multiple tasks, like detection tasks in images and LiDAR point clouds, respectively. Learning-based methods require no artificial definition of features, which can learn useful information using neural networks. These methods can also be applied in LiDAR–camera calibration.

3.4.1 End-to-end methods

End-to-end methods use network models to process input camera images and LiDAR point clouds, then directly output the extrinsic parameters. These methods achieve optimal calibration parameters by minimizing corresponding loss functions.

End-to-end methods rely heavily on the training data. In the training phase, pairs of point clouds and images accompanied with ground truth extrinsic parameters are fed to the model. However, referring to the ground truth of hundreds of thousands of different relative positions of laser scanners and cameras can be bothersome. Therefore, Schneider et al. (2017) reformulated the problem as determining the mis-calibration $\phi_{mis\text{-}calib}$ between the initial calibration parameter T_{init} and the ground truth parameter T_{gt}. With the mis-calibrated extrinsic parameter T_{init} and camera matrix K, the LiDAR points were projected to the camera frame as depth images. The mis-calibration $\phi_{mis\text{-}calib}$ can be varied randomly to get a huge amount of training data.

For end-to-end methods, their network architectures can be classified into three categories:

Regression: Methods in this category take RGB pictures and depth images as inputs. Their networks often have two branches to extract features from RGB and depth images respectively. Then the features from both modalities are fused by the feature matching component. Finally, the global information extracted from both modalities is regressed to obtain the mis-calibration parameters. The common architecture of regression methods is shown in Fig. 8.

RegNet (Schneider et al. 2017) is one of the first deep learning methods that integrate feature extraction, feature matching, and global regression into a convolutional neural

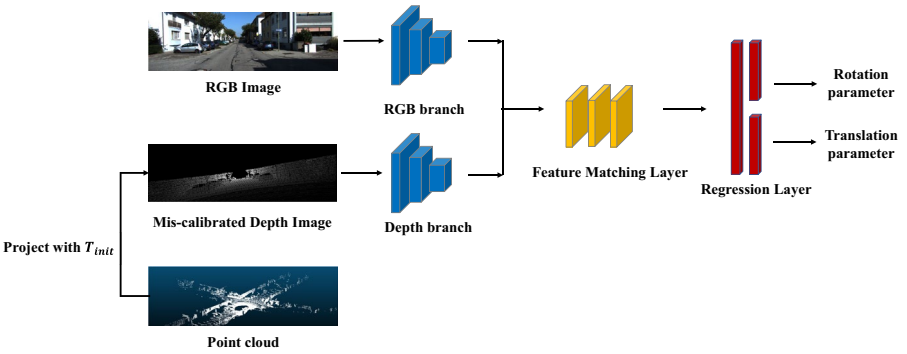


Fig. 8 The common architecture of regression methods for the estimation of the extrinsic calibration parameters for LiDAR–camera calibration. The point cloud is projected to the image plane using an initial calibration T_{init}. The RGB and Depth branches extract the features for matching separately, and then the features are matched in the second part. Lastly, the regression layer regresses the mis-calibration parameters by gathering global information

network, for estimating extrinsic parameters between the LiDAR and the camera. In RegNet, blocks of Network in Network (NiN) (Lin et al. 2013) were used to extract and match the features of LiDAR depth maps and camera RGB images.

Based on RegNet, Liu et al. (2018) presented an online calibration method for visual and depth sensors. The depth camera and the LiDAR are first calibrated and fused as a virtual depth sensor, then this virtual sensor is calibrated with the camera.

Iyer et al. (2018) proposed CalibNet, which takes the geometry information into account and introduces a 3D spatial transformer layer in the model. The RGB branch is the convolutional layers of a pre-trained ResNet-18 (He et al. 2016), and the depth branch is a similar network but with the number of filters halved. The two outputs are then concatenated and passed through the global aggregation block. CalibNet performs end-to-end training by maximizing the geometric and photometric consistency between the image and the point cloud.

Yuan et al. (2020) designed RGGNet. This method considers Riemannian geometry and employs deep generative models to build a tolerance-aware loss function. RGGNet not only considers the calibration error, but also focuses on the tolerance within the error bounds.

Shi et al. (2020) created and demonstrated CalibRCNN, which combines CNN with LSTM. The output features from the two branches were fused then fed into the LSTM layer to extract temporal features for sequential learning. CalibRCNN not just added pose constraints between consecutive frames, but uses the geometric and photometric loss to refine the calibration accuracy of the predicted transformation parameters.

Zhao et al. (2021) proposed CalibDNN and applied it to a complex dataset with diverse scenarios. As a simple system with one model and a single iteration, CalibDNN considers transformation loss and geometric loss to maximize the consistency of multimodal data.

Lv et al. (2021) presented LCCNet for extrinsic calibration of a LiDAR and a camera. To match the features between depth image and RGB image, cost volume layer is constructed instead of concatenating the features directly. In addition to the smooth L1-Loss as supervision for the ground truth, a point cloud constraint is also added to the loss function.

Calibration Flow: The concept of optical flow refers to the movement of target pixels in an image due to the behaviors of objects or the motions of the camera in two consecutive frames. The calibration flow is similar to the optical flow, which includes two channels and represents the horizontal and vertical offsets. Methods in this category take 2D pictures and LiDAR depth maps as inputs. Images from the two modalities are fed into an optical flow network to predict the flow between mis-calibrated depth map and the RGB image, then get the correspondence between cloud points and image pixels. Finally, the initial extrinsic parameters can be optimized by minimizing the projection errors.

Lv et al. (2021) showcased CFNet, which can generate a refined calibration flow. A group of accurate 2D–3D correspondences can be constructed and the EPnP algorithm with the RANSAC scheme is applied to estimate the extrinsic parameters.

Jing et al. (2022) presented DXQ-Net, which predicts the calibration flow with uncertainty. The network architecture of DXQ-Net is derived from RAFT (Teed and Deng 2021), and a differentiable pose estimation module is used to compute the extrinsic parameters.

Keypoints: Unlike end-to-end learning methods in the above two categories, keypoint methods directly point clouds as inputs, along with camera images. The network extracts feature descriptors from the input data, then finds the corresponding 2D points on the image for each 3D keypoint. Finally, the extrinsic parameters between the LiDAR and camera can be estimated.

Ye et al. (2022) offered RGKCNet model, a 2D–3D pose estimation network based on keypoints. This network extracts sparse keypoints and matches them, then a weighted nonlinear PnP solver is applied to estimate the pose. RGKCNet uses extrinsic calibration constraints to solve the data association problem of 2D–3D points. The optimizer in the network is based on geometric constraints.

3.4.2 Hybrid-learning methods

Different from end-to-end methods, hybrid-learning methods use neural networks only to extract information such as geometric and semantic features, while feature association and extrinsic parameter calculation procedures are still based on non-learning methods.

As introduced in Sect. 3.2, Wang et al. (2020) designed SOIC with the introduction of semantic centroids, to ease the demand for prior knowledge of initial calibration. In SOIC, 2D and 3D semantic centroids are calculated based on semantic segmentation of images and LiDAR points. Thus, the LiDAR–Camera calibration initialization is transformed into a PnP problem. Furthermore, the optimal calibration parameter was obtained by minimizing the cost function based on the semantic elements.

Zhu et al. (2020) suggested aligning semantic features instead of edge features to improve LiDAR–camera calibration robustness, especially for low-resolution LiDAR and

noisy inputs. They extracted cars from both the image and the point cloud, and the extrinsic calibration was optimized through a cost function under semantic constraints.

3.4.3 Summary of learning based methods

Learning based methods use neural networks to find potential features of LiDAR and camera data, these methods can obtain suitable features and achieve good results if there are sufficient data for training. However, existing learning algorithms for calibration usually require a large number of training calculations, which results in a great deal of computational cost. Moreover, they are demanding the conditions of application, which implies that the algorithms need broadly similar scenes for training and validation/test, thus their generalization performance needs to be improved urgently.

4 Discussion

This paper provides a systematic review of automatic targetless methods for extrinsic calibration between LiDAR sensors and cameras. In literature, current targetless LiDAR–camera calibration paradigms can be divided into four categories, i.e., information theory based methods, feature based methods, ego-motion based methods, and learning based methods.

In specific, information theory based methods evaluate the statistical similarity of data from LiDAR and camera. They calculate precise calibration parameters by maximizing a similarity measurement. However, the accuracy is susceptible to some environmental factors, such as occlusion between objects or the presence of shadows (Pandey et al. 2012) (Parmehr et al. 2014).

Feature based methods extract information from the natural environment and find correspondence between images and point clouds after the feature matching phase. Distinguishable features are available in LiDAR data and optical images. Features can be separated into geometric, semantic, and motion ones. Geometric features such as points and lines are easy to extract (González-Aguilera et al. 2009; Zhang et al. 2021), while semantic features have more differentiation degrees and are simple to match (Wang et al. 2020). However, LiDAR data and optical images often capture different characteristics of the environment and feature extraction can be easily affected by random factors such as noise and occlusion (Zhu et al. 2020). Some methods use the motion trajectory of the detected object as a motion feature. This motion feature introduces dynamic information to allow for temporal calibration, however, this variety of methods requires lots of moving objects to generate sufficient trajectories to be tracked (Peršić et al. 2020).

Comparing the information theory based method with feature based methods, the former runs on the entire 2D–3D data, which avoids the problem of unstable feature extraction and matching, and yields alignment information over the whole data. On the other hand, the latter uses features extracted from 2D and 3D data, which are more discriminative and lead to a simpler optimization.

Ego-motion based methods exploit the motion information generated from the two sensors. These methods can be divided into hand-eye and 3D structure estimation based

methods, in accordance with how motion information is used to transform calibration into different problems. Using the trajectories of the LiDAR sensor and the camera, hand-eye methods can proceed without an initial guess for the extrinsic parameters. They require no overlapping field of view (Park et al. 2020), as they do not need to extract features or compute the statistical similarity of the corresponding attributes. However, the accuracy of these algorithms strongly depends on sufficient estimation performance, which usually needs to be refined by other methods (Shi et al. 2019). Since methods based on ego-motion introduces dynamic information, they also need to solve the problem of time synchronization.

The most typical learning based methods are end-to-end methods. End-to-end approaches transform several calibration steps into single-step methods using neural network models. They employ such models to learn useful features by themselves instead of defining features by hand. With the help of high-performance neural networks, these methods can achieve satisfactory calibration results. However, datasets for calibration are difficult to obtain (Schneider et al. 2017). End-to-end methods rely heavily on labeled data for training, and their performance often ends up being unstable in unseen environments. There are also hybrid approaches where that use semantic segmentation networks to extract more robust features while using classical algorithms for subsequent matching and optimization. We list the strengths and problems of the four methods in Table 4.

However, achieving accurate and robust automatic targetless LiDAR–camera calibration for different types of scenarios remains a challenge for future efforts. Different methods have their applicable and inapplicable scenarios. It is a challenging task to design a method that can be used in indoor, urban, and natural environments. On the other hand, for online calibration, some unanswered questions still remain: how to quickly detect the offset of the calibration parameters? At what rate should the calibration data be updated? How much does the offset of the calibration parameters affect the perception results? At the same time, an ideal calibration solution should be able to run on various platforms regardless of their computational constraints, so there is also a difficult task to balance accuracy, efficiency, and resources.

Hybrid methods provide a promising way to improve the performance.For example, the combination of hand-craft and learning based methods can take advantage of deep learning capabilities while maintaining theoretical modeling. Moreover, such methods can reduce their computational cost while maintaining acceptable performance. There is also the combination of SLAM technology and the integration of different sensors such as IMU. SLAM technology is very similar in feature extraction and matching, and with the help of other sensors, additional information can be brought to help calibration task. In learning based methods, the successful application of semi-supervised or unsupervised learning would be helpful due to the hard-to-obtain nature of the ground truth for calibration parameters.

5 Conclusion

This paper reviews the existing calibration algorithms for automatic targetless calibration between LiDARs and cameras. Unmanned intelligent perception systems are usually equipped with a combination of LiDAR sensors and cameras, taking advantage of the two sensors to better perceive the surrounding environment. A key pre-step of data fusion is to calibrate the extrinsic parameters of the sensor. Traditional methods either rely on

Table 4 Comparison of automatic targetless LiDAR–camera calibration methods

Methods	Strength	Problem
Information theory based methods	∘ Obtain alignment information for the entire 2D–3D data. ∘ Accurate calibration results.	∗ Accuracy is susceptible to some environmental factors, such as occlusion by objects or the presence of shadows. ∗ Require large area of overlapping information, panoramic cameras are usually require.
Feature based methods	∘ A large number of features exist in the environment. ∘ Geometric features are easy to extract while semantic features have better recognition and are easy to match. ∘ Accurate calibration results	∗ Feature extraction is easily affected by environmental factors, such as noise and occlusion. ∗ Geometric features matching across modalities is diffcult while semantic segmentation is costly.
Ego-motion based methods	For hand-eye based methods: ∘ Do not require initial calibration parameter and overlapping field of view. ∘ Do not depend on specific sensor types and suitable for many sensor types such as cameras, LiDARs, radars, or IMUs. For 3D structure estimation based methods: ∘ Accurate calibration results	For hand-eye based methods: ∗ Rough calibration result and usually need refined by other methods. ∗ Require precise temporal registration between sensors. For 3D structure estimation based methods: ∗ Problems such as as inaccurate scaling effects may occur in the SfM process;
Learning based methods	∘ Use neural network to learn useful features by themselves instead manually defined features. ∘ Accurate calibration results	∗ Difficult to provided a dataset with ground truth calibration parameters. ∗ Heavily rely on the labeled datasets for training, whose performance turns to be unstable in new environments.

calibration objects or require manual interaction. Automatic targetless methods spontaneously obtain information from the surrounding environments in the data, thus eliminating the requirements for calibration targets and human efforts.

The current automatic targetless LiDAR–camera calibration methods can be categorized into four categories, i.e., information theory based, feature-based, ego-motion based, and learning based methods. Methods in the first category measure the statistical similarity between the LiDAR data and optical images. The feature-based methods extract geometric or semantic features of the environment, instead of running on the entire 2D–3D data as inputs. The ego-motion based methods exploit the motion of sensors from LiDAR point and camera image sequences. At last, learning based methods use neural network models to learn useful features rather than define features manually.

Acknowledgements The work is partially supported by the 2030 National Key AI Program of China 2018AAA0100500, Guangdong Province R&D Program 2020B0909050001, Anhui Province Development and Reform Commission 2020 New Energy Vehicle Industry Innovation Development Project and 2021 New Energy and Intelligent Connected Vehicle Innovation Project, CAAI-Huawei MindSpore Open Fund, Shenzhen Yijiahe Technology R&D Co., Ltd., and Huawei Cloud Computing Technologies Co., Ltd.

Author contributions Had the idea for the article: XL, JJ; Performed the literature search and data analysis: XL, YX; Drafted the work: XL, YX, BW, HR; Critically revised the work: BW, YZ, JJ.

Declarations

Competing interests The authors declare no competing interests.

References

Alba M, Barazzetti L, Scaioni M, Remondino F (2012) Automatic registration of multiple laser scans using panoramic RGB and intensity images. Int Arch Photogramm Remote Sens Spat Inf Sci XXXVIII–5/W12:49–54. https://doi.org/10.5194/isprsarchives-xxxviii-5-w12-49-2011

Bai Z, Jiang G, Xu A (2020) LiDAR-camera calibration using line correspondences. Sensors 20(21):6319. https://doi.org/10.3390/s20216319

Banerjee K, Notz D, Windelen J, Gavarraju S, He M (2018) Online camera LiDAR fusion and object detection on hybrid data for autonomous driving. In: IEEE intelligent vehicles symposium (IV). IEEE. https://doi.org/10.1109/ivs.2018.8500699

Barzilai J, Borwein JM (1988) Two-point step size gradient methods. IMA J Numer Anal 8(1):141–148. https://doi.org/10.1093/imanum/8.1.141

Bay H, Tuytelaars T, Gool LV (2006) SURF: speeded up robust features. In: Computer vision – ECCV 2006. Springer, Berlin, pp 404–417. https://doi.org/10.1007/11744023_32

Belghazi MI, Baratin A, Rajeshwar S, Ozair S, Bengio Y, Courville A, Hjelm D (2018) Mutual information neural estimation. In: International conference on machine learning. PMLR, pp 531–540

Besl PJ, McKay ND (1992) Method for registration of 3-d shapes. In: Schenker PS (ed) Sensor fusion IV: control paradigms and data structures. SPIE. https://doi.org/10.1117/12.57955

Bileschi S (2009) Fully automatic calibration of LIDAR and video streams from a vehicle. In: 2009 IEEE 12th international conference on computer vision workshops, ICCV workshops. IEEE. https://doi.org/10.1109/iccvw.2009.5457439

Blaga B-C-Z, Nedevschi S (2017) Online cross-calibration of camera and LIDAR. In: 2017 13th IEEE international conference on intelligent computer communication and processing (ICCP). IEEE. https://doi.org/10.1109/iccp.2017.8117020

Böhm J, Becker S (2007) Automatic marker-free registration of terrestrial laser scans using reflectance. In: Proceedings of the 8th conference on optical 3D measurement techniques, Zurich, Switzerland, pp 9–12

CANNY J (1987) A computational approach to edge detection, pp 184–203. https://doi.org/10.1016/b978-0-08-051581-6.50024-6

Castorena J, Kamilov US, Boufounos PT (2016) Autocalibration of lidar and optical cameras via edge alignment. In: 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE. https://doi.org/10.1109/icassp.2016.7472200

Castorena J, Puskorius GV, Pandey G (2020) Motion guided LiDAR-camera self-calibration and accelerated depth upsampling for autonomous vehicles. J Intell Robot Syst 100(3–4):1129–1138. https://doi.org/10.1007/s10846-020-01233-w

Chen X, Ma H, Wan J, Li B, Xia T (2017) Multi-view 3d object detection network for autonomous driving. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2017.691

Chien H-J, Klette R, Schneider N, Franke U (2016) Visual odometry driven online calibration for monocular LiDAR-camera systems. In: 2016 23rd international conference on pattern recognition (ICPR). IEEE. https://doi.org/10.1109/icpr.2016.7900068

Chou JCK, Kamel M (1991) Finding the position and orientation of a sensor on a robot manipulator using quaternions. Int J Robot Res 10(3):240–254. https://doi.org/10.1177/027836499101000305

Corsini M, Dellepiane M, Ganovelli F, Gherardi R, Fusiello A, Scopigno R (2012) Fully automatic registration of image sets on approximate geometry. Int J Comput Vis 102(1–3):91–111. https://doi.org/10.1007/s11263-012-0552-5

Cortinhal T, Tzelepis G, Aksoy EE (2020) SalsaNext: fast, uncertainty-aware semantic segmentation of LiDAR point clouds. In: Advances in visual computing. Springer, pp 207–222. https://doi.org/10.1007/978-3-030-64559-5_16

Cui Y, Chen R, Chu W, Chen L, Tian D, Li Y, Cao D (2022) Deep learning for image and point cloud fusion in autonomous driving: a review. IEEE Trans Intell Transp Syst 23(2):722–739. https://doi.org/10.1109/tits.2020.3023541

Dhall A, Chelani K, Radhakrishnan V, Krishna KM (2017) Lidar-camera calibration using 3d-3d point correspondences. arXiv preprint arXiv:1705.09785

Edelsbrunner H, Kirkpatrick D, Seidel R (1983) On the shape of a set of points in the plane. IEEE Trans Inf Theory 29(4):551–559. https://doi.org/10.1109/tit.1983.1056714

Feng D, Haase-Schutz C, Rosenbaum L, Hertlein H, Glaser C, Timm F, Wiesbeck W, Dietmayer K (2021) Deep multi-modal object detection and semantic segmentation for autonomous driving: datasets, methods, and challenges. IEEE Trans Intell Transp Syst 22(3):1341–1360. https://doi.org/10.1109/tits.2020.2972974

Förstner W, Gülch E (1987) A fast operator for detection and precise location of distinct points, corners and centres of circular features. In: Proc. ISPRS intercommission conference on fast processing of photogrammetric data, vol 6. Interlaken, pp 281–305

Geiger A, Moosmann F, Car O, Schuster B (2012) Automatic camera and range sensor calibration using a single shot. In: 2012 IEEE international conference on robotics and automation. IEEE. https://doi.org/10.1109/icra.2012.6224570

Geiger A, Lenz P, Stiller C, Urtasun R (2013) Vision meets robotics: the KITTI dataset. Int J Robot Res 32(11):1231–1237. https://doi.org/10.1177/0278364913491297

González-Aguilera D, Rodríguez-Gonzálvez P, Gómez-Lahoz J (2009) An automatic procedure for co-registration of terrestrial laser scanners and digital cameras. ISPRS J Photogramm Remote Sens 64(3):308–316. https://doi.org/10.1016/j.isprsjprs.2008.10.002

Graeter J, Wilczynski A, Lauer M (2018) LIMO: Lidar-monocular visual odometry. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE. https://doi.org/10.1109/iros.2018.8594394

Guindel C, Beltran J, Martin D, Garcia F (2017) Automatic extrinsic calibration for lidar-stereo vehicle sensor setups. In: 2017 IEEE 20th international conference on intelligent transportation systems (ITSC). IEEE. https://doi.org/10.1109/itsc.2017.8317829

Guislain M, Digne J, Chaine R, Monnier G (2017) Fine scale image registration in large-scale urban LIDAR point sets. Comput Vis Image Underst 157:90–102. https://doi.org/10.1016/j.cviu.2016.12.004

Hassanein M, Moussa A, El-Sheimy N (2016) A new automatic system calibration of multi-cameras and lidar sensors. ISPRS XLI–B1:589–594. https://doi.org/10.5194/isprs-archives-xli-b1-589-2016

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2016.90

Hofmann S, Eggert D, Brenner C (2014) Skyline matching based camera orientation from images and mobile mapping point clouds. ISPRS Ann Photogramm Remote Sens Spat Inf Sci II–5:181–188. https://doi.org/10.5194/isprsannals-ii-5-181-2014

Horn M, Wodtko T, Buchholz M, Dietmayer K (2021) Online extrinsic calibration based on per-sensor ego-motion using dual quaternions. IEEE Robot Autom Lett 6(2):982–989. https://doi.org/10.1109/lra.2021.3056352

Hsu C-M, Wang H-T, Tsai A, Lee C-Y (2018) Online recalibration of a camera and lidar system. In: 2018 IEEE international conference on systems, man, and cybernetics (SMC). IEEE. https://doi.org/10.1109/smc.2018.00687

Huang K, Stachniss C (2017) Extrinsic multi-sensor calibration for mobile robots using the gauss-helmert model. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE. https://doi.org/10.1109/iros.2017.8205952

Hussein A, Marin-Plaza P, Martin D, de la Escalera A, Armingol JM (2016) Autonomous off-road navigation using stereo-vision and laser-rangefinder fusion for outdoor obstacles detection. In: IEEE intelligent vehicles symposium (IV). IEEE. https://doi.org/10.1109/ivs.2016.7535372

Igelbrink F, Wiemann T, Pütz S, Hertzberg J (2018) Markerless ad-hoc calibration of a hyperspectral camera and a 3d laser scanner. In: Intelligent autonomous systems, vol 15. Springer, pp 748–759. https://doi.org/10.1007/978-3-030-01370-7_58

Irie K, Sugiyama M, Tomono M (2016) Target-less camera-LiDAR extrinsic calibration using a bagged dependence estimator. In: 2016 IEEE international conference on automation science and engineering (CASE). IEEE. https://doi.org/10.1109/coase.2016.7743564

Ishikawa R, Oishi T, Ikeuchi K (2018) LiDAR and camera calibration using motions estimated by sensor fusion odometry. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE. https://doi.org/10.1109/iros.2018.8593360

Iyer G, Ram RK, Murthy JK, Krishna KM (2018) CalibNet: geometrically supervised extrinsic calibration using 3d spatial transformer networks. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE. https://doi.org/10.1109/iros.2018.8593693

Jiang J, Xue P, Chen S, Liu Z, Zhang X, Zheng N (2018) Line feature based extrinsic calibration of LiDAR and camera. In: 2018 IEEE international conference on vehicular electronics and safety (ICVES). IEEE. https://doi.org/10.1109/icves.2018.8519493

Jiang P, Osteen P, Saripalli S (2021) SemCal: semantic LiDAR-camera calibration using neural mutual information estimator. In: 2021 IEEE international conference on multisensor fusion and integration for intelligent systems (MFI). IEEE. https://doi.org/10.1109/mfi52462.2021.9591203

Jing X, Ding X, Xiong R, Deng H, Wang Y (2022) DXQ-Net: differentiable lidar-camera extrinsic calibration using quality-aware flow. arXiv preprint arXiv:2203.09385

Kelley CT (1999) Iterative methods for optimization. SIAM

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, vol 4. IEEE, pp 1942–1948

Khurana A, Nagla KS (2021) Extrinsic calibration methods for laser range finder and camera: a systematic review. Mapan 36(3):669–690. https://doi.org/10.1007/s12647-021-00500-x

Kim A, Osep A, Leal-Taixe L (2021) EagerMOT: 3d multi-object tracking via sensor fusion. In: 2021 IEEE international conference on robotics and automation (ICRA). IEEE. https://doi.org/10.1109/icra48506.2021.9562072

Krähenbühl P, Koltun V (2011) Efficient inference in fully connected crfs with Gaussian edge potentials. Advances in neural information processing systems 24

Levenberg K (1944) A method for the solution of certain non-linear problems in least squares. Q Appl Math 2(2):164–168. https://doi.org/10.1090/qam/10666

Levinson J, Thrun S (2013) Automatic online calibration of cameras and lasers. In: Robotics: science and systems IX. Robotics: Science and Systems Foundation. https://doi.org/10.15607/rss.2013.ix.029

Li M, Chen X, Li X, Ma B, Vitányi PM (2004) The similarity metric. IEEE Trans Inf Theory 50(12):3250–3264

Li T, Fang J, Zhong Y, Wang D, Xue J (2017) Online high-accurate calibration of rgb+ 3d-lidar for autonomous driving. In: Lecture notes in computer science. Springer, pp 254–263. https://doi.org/10.1007/978-3-319-71598-8_23

Li J, Yang B, Chen C, Huang R, Dong Z, Xiao W (2018) Automatic registration of panoramic image sequence and mobile laser scanning data using semantic features. ISPRS J Photogramm Remote Sens 136:41–57. https://doi.org/10.1016/j.isprsjprs.2017.12.005

Liao Q, Liu M (2019) Extrinsic calibration of 3d range finder and camera without auxiliary object or human intervention. In: 2019 IEEE international conference on real-time computing and robotics (RCAR). IEEE. https://doi.org/10.1109/rcar47638.2019.9044146

Li-Chee-Ming J, Armenakis C, Fusion of optical and terrestrial laser scanner data. In: The (2010) Canadian geomatics conference and symposium of commission I. ISPRS Convergence in Geomatics-Shaping Canada's Competitive Landscape, Citeseer, p 2010

Lin M, Chen Q, Yan S (2013) Network in network. arXiv preprint arXiv:1312.4400
Liu X, Deng Z, Yang Y (2018) Recent progress in semantic image segmentation. Artif Intell Rev 52(2):1089–1106. https://doi.org/10.1007/s10462-018-9641-3
Liu H, Liu Y, Gu X, Wu Y, Qu F, Huang L (2018) A deep-learning based multi-modality sensor calibration method for USV. In: 2018 IEEE fourth international conference on multimedia big data (BigMM). IEEE. https://doi.org/10.1109/bigmm.2018.8499349
Liu X, Yuan C, Zhang F (2021) Fast and accurate extrinsic calibration for multiple lidars and cameras. arXiv preprint arXiv:2109.06550
Lowe D (1999) Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision. IEEE. https://doi.org/10.1109/iccv.1999.790410
Lu X, Liu Y, Li K (2019) Fast 3d line segment detection from unorganized point cloud. arXiv preprint arXiv:1901.02532
Lv X, Wang S, Ye D (2021a) CFNet: LiDAR-camera registration using calibration flow network. Sensors 21(23):8112. https://doi.org/10.3390/s21238112
Lv X, Wang B, Dou Z, Ye D, Wang S (2021b) LCCNet: LiDAR and camera self-calibration using cost volume network. In: 2021 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW). IEEE. https://doi.org/10.1109/cvprw53098.2021.00324
Ma H, Liu K, Liu J, Qiu H, Xu D, Wang Z, Gong X, Yang S (2021a) Simple and efficient registration of 3d point cloud and image data for an indoor mobile mapping system. JOSA A 38(4):579–586. https://doi.org/10.1364/josaa.414042
Ma T, Liu Z, Yan G, Li Y (2021b) Crlf: automatic calibration and refinement based on line feature for lidar and camera in road scenes. arXiv preprint arXiv:2103.04558
Miled M, Soheilian B, Habets E, Vallet B (2016) Hybrid online mobile laser scanner calibration through image alignment by mutual information. ISPRS Ann Photogramm Remote Sens Spat Inf Sci III–1:25–31. https://doi.org/10.5194/isprsannals-iii-1-25-2016
Morel J-M, Yu G (2009) ASIFT: a new framework for fully affine invariant image comparison. SIAM J Imag Sci 2(2):438–469. https://doi.org/10.1137/080732730
Moussa W, Abdel-Wahab M, Fritsch D (2012) Automatic fusion of digital images and laser scanner data for heritage preservation. In: Progress in cultural heritage preservation. Springer, Berlin, pp 76–85. https://doi.org/10.1007/978-3-642-34234-9_8
Munoz-Banon MA, Candelas FA, Torres F (2020) Targetless camera-LiDAR calibration in unstructured environments. IEEE Access 8:143692–143705. https://doi.org/10.1109/access.2020.3014121
Mur-Artal R, Montiel JMM, Tardos JD (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans Rob 31(5):1147–1163. https://doi.org/10.1109/tro.2015.2463671
Nagy B, Benedek C (2020) On-the-fly camera and lidar calibration. Remote Sens 12(7):1137. https://doi.org/10.3390/rs12071137
Nagy B, Kovacs L, Benedek C (2019a) Online targetless end-to-end camera-LIDAR self-calibration. In: 2019 16th international conference on machine vision applications (MVA). IEEE. https://doi.org/10.23919/mva.2019.8757887
Nagy B, Kovacs L, Benedek C (2019b) SFM and semantic information based online targetless camera-LIDAR self-calibration. In: 2019 IEEE international conference on image processing (ICIP). IEEE. https://doi.org/10.1109/icip.2019.8804299
Nelder JA, Mead R (1965) A simplex method for function minimization. Comput J 7(4):308–313. https://doi.org/10.1093/comjnl/7.4.308
Nie J, Pan F, Xue D, Luo L (2021) A survey of extrinsic parameters calibration techniques for autonomous devices. In: 2021 33rd Chinese control and decision conference (CCDC). IEEE. https://doi.org/10.1109/ccdc52312.2021.9602601
Nieto JI, Monteiro ST, Viejo D (2010) 3d geological modelling using laser and hyperspectral data. In: 2010 IEEE international geoscience and remote sensing symposium. IEEE. https://doi.org/10.1109/igarss.2010.5651553
Nurunnabi A, Belton D, West G (2012) Robust segmentation in laser scanning 3d point cloud data. In: 2012 international conference on digital image computing techniques and applications (DICTA). IEEE. https://doi.org/10.1109/dicta.2012.6411672
Oishi T, Nakazawa A, Kurazume R, Ikeuchi K (2005) Fast simultaneous alignment of multiple range images using index images. In: Fifth international conference on 3-D digital imaging and modeling (3DIM'05). IEEE. https://doi.org/10.1109/3dim.2005.41
Pandey G, McBride JR, Savarese S, Eustice RM (2012) Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. In: Twenty-sixth AAAI conference on artificial intelligence. https://doi.org/10.1609/aaai.v26i1.8379

Pandey G, McBride JR, Savarese S, Eustice RM (2014) Automatic extrinsic calibration of vision and lidar by maximizing mutual information. J Field Robot 32(5):696–722. https://doi.org/10.1002/rob.21542
Park F, Martin B (1994) Robot sensor calibration: solving AX=XB on the euclidean group. IEEE Trans Robot Autom 10(5):717–721. https://doi.org/10.1109/70.326576
Park C, Moghadam P, Kim S, Sridharan S, Fookes C (2020) Spatiotemporal camera-LiDAR calibration: a targetless and structureless approach. IEEE Robot Autom Lett 5(2):1556–1563. https://doi.org/10.1109/lra.2020.2969164
Parmehr EG, Fraser CS, Zhang C, Leach J (2014) Automatic registration of optical imagery with 3d lidar data using statistical similarity. ISPRS J Photogramm Remote Sens 88:28–40
Pascoe G, Maddern W, Newman P (2015) Direct visual localisation and calibration for road vehicles in changing city environments. In: 2015 IEEE international conference on computer vision workshop (ICCVW). IEEE. https://doi.org/10.1109/iccvw.2015.23
Peršić J, Petrović L, Marković I, Petrović I (2020) Online multi-sensor calibration based on moving object tracking. Adv Robot 35(3–4):130–140. https://doi.org/10.1080/01691864.2020.1819874
Pomerleau F, Colas F, Siegwart R, Magnenat S (2013) Comparing ICP variants on real-world data sets. Auton Robot 34(3):133–148. https://doi.org/10.1007/s10514-013-9327-2
Powell MJ (2009) The bobyqa algorithm for bound constrained optimization without derivatives. Cambridge NA Report NA2009/06, University of Cambridge, Cambridge, 26
Pusztai Z, Hajder L (2017) Accurate calibration of lidar-camera systems using ordinary boxes. In: 2017 IEEE international conference on computer vision workshops (ICCVW). IEEE. https://doi.org/10.1109/iccvw.2017.53
Quan L, Lan Z (1999) Linear n-point camera pose determination. IEEE Trans Pattern Anal Mach Intell 21(8):774–780. https://doi.org/10.1109/34.784291
Scaramuzza D, Harati A, Siegwart R (2007) Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes. In: 2007 IEEE/RSJ international conference on intelligent robots and systems. IEEE. https://doi.org/10.1109/iros.2007.4399276
Schneider N, Piewak F, Stiller C, Franke U (2017) RegNet: multimodal sensor registration using deep neural networks. In: IEEE intelligent vehicles symposium (IV). IEEE. https://doi.org/10.1109/ivs.2017.7995968
Scott DW (1992) Multivariate density estimation: theory, practice and visualisation. Wiley, New York
Shannon CE (2001) A mathematical theory of communication. ACM SIGMOBILE Mobile Comput Commun Rev 5(1):3–55
Shi C, Huang K, Yu Q, Xiao J, Lu H, Xie C (2019a) Extrinsic calibration and odometry for camera-LiDAR systems. IEEE Access 7:120106–120116. https://doi.org/10.1109/access.2019.2937909
Shi S, Wang X, Li H (2019b) PointRCNN: 3d object proposal generation and detection from point cloud. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2019.00086
Shi J, Zhu Z, Zhang J, Liu R, Wang Z, Chen S, Liu H (2020) CalibRCNN: calibrating camera and LiDAR by recurrent convolutional neural network and geometric constraints. In: 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE. https://doi.org/10.1109/iros45743.2020.9341147
Shiu Y, Ahmad S (1989) Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB. IEEE Trans Robot Autom 5(1):16–29. https://doi.org/10.1109/70.88014
Sobel I, Duda R, Hart P Sobel-feldman operator
Studholme C, Hill D, Hawkes D (1999) An overlap invariant entropy measure of 3d medical image alignment. Pattern Recogn 32(1):71–86. https://doi.org/10.1016/s0031-3203(98)00091-0
Swart A, Broere J, Veltkamp R, Tan R (2011) Refined non-rigid registration of a panoramic image sequence to a LiDAR point cloud. In: Photogrammetric image analysis. Springer, Berlin, pp 73–84. https://doi.org/10.1007/978-3-642-24393-6_7
Takikawa T, Acuna D, Jampani V, Fidler S (2019) Gated-SCNN: gated shape CNNs for semantic segmentation. In: 2019 IEEE/CVF international conference on computer vision (ICCV). IEEE. https://doi.org/10.1109/iccv.2019.00533
Taylor Z, Nieto J (2012) A mutual information approach to automatic calibration of camera and lidar in natural environments. In: Australian conference on robotics and automation, pp 3–5
Taylor Z, Nieto J (2013) Automatic calibration of lidar and camera images using normalized mutual information. In: 2013 IEEE international conference on robotics and automation (ICRA). Citeseer
Taylor Z, Nieto J (2014) Parameterless automatic extrinsic calibration of vehicle mounted lidar-camera systems. In: International conference on robotics and automation: long term autonomy workshop, number October, pp 3–6

Taylor Z, Nieto J (2015) Motion-based calibration of multimodal sensor arrays. In: 2015 IEEE international conference on robotics and automation (ICRA). IEEE. https://doi.org/10.1109/icra.2015.7139872
Taylor Z, Nieto J (2016) Motion-based calibration of multimodal sensor extrinsics and timing offset estimation. IEEE Trans Rob 32(5):1215–1229. https://doi.org/10.1109/tro.2016.2596771
Taylor Z, Nieto J, Johnson D (2013) Automatic calibration of multi-modal sensor systems using a gradient orientation measure. In: 2013 IEEE/RSJ international conference on intelligent robots and systems, pp 1293–1300. IEEE
Taylor Z, Nieto J, Johnson D (2014) Multi-modal sensor calibration using a gradient orientation measure. J Field Robot 32(5):675–695. https://doi.org/10.1002/rob.21523
Teed Z, Deng J (2021) RAFT: recurrent all-pairs field transforms for optical flow (extended abstract) . In: Proceedings of the thirtieth international joint conference on artificial intelligence. International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2021/662
Toth T, Pusztai Z, Hajder L (2020) Automatic LiDAR-camera calibration of extrinsic parameters using a spherical target. In: 2020 IEEE international conference on robotics and automation (ICRA). IEEE. https://doi.org/10.1109/icra40945.2020.9197316
Ullman S (1979) The interpretation of structure from motion. Proc R Soc Lond B 203(1153):405–426. https://doi.org/10.7551/mitpress/3877.003.0009
Unnikrishnan R, Hebert M (2005) Fast extrinsic calibration of a laser rangefinder to a camera. Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-09
Vaida A-S, Nedevschi S (2019) Automatic extrinsic calibration of LIDAR and monocular camera images. In: 2019 IEEE 15th international conference on intelligent computer communication and processing (ICCP). IEEE. https://doi.org/10.1109/iccp48234.2019.8959801
Vel'as M, Španěl M, Materna Z, Herout A (2014) Calibration of rgb camera with velodyne lidar
Vo A-V, Truong-Hong L, Laefer DF, Bertolotto M (2015) Octree-based region growing for point cloud segmentation. ISPRS J Photogramm Remote Sens 104:88–100. https://doi.org/10.1016/j.isprsjprs.2015.01.011
von Gioi RG, Jakubowicz J, Morel J-M, Randall G (2012) LSD: a line segment detector. Image Process On Line 2:35–55. https://doi.org/10.5201/ipol.2012.gjmr-lsd
Vora S, Lang AH, Helou B, Beijbom O (2020) PointPainting: sequential fusion for 3d object detection. In: 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr42600.2020.00466
Wang R, Ferrie FP, Macfarlane J (2012) Automatic registration of mobile LiDAR and spherical panoramas. In: 2012 IEEE Computer Society conference on computer vision and pattern recognition workshops. IEEE. https://doi.org/10.1109/cvprw.2012.6238912
Wang L, Xiao Z, Zhao D, Wu T, Dai B (2018) Automatic extrinsic calibration of monocular camera and LIDAR in natural scenes. In: 2018 IEEE international conference on information and automation (ICIA). IEEE. https://doi.org/10.1109/icinfa.2018.8812555
Wang Z, Wu Y, Niu Q (2020a) Multi-sensor fusion in automated driving: a survey. IEEE Access 8:2847–2868. https://doi.org/10.1109/access.2019.2962554
Wang W, Nobuhara S, Nakamura R, Sakurada K (2020b) Soic: semantic online initialization and calibration for lidar and camera. arXiv preprint arXiv:2003.04260
Wang Y, Li J, Sun Y, Shi M (2021) A survey of extrinsic calibration of lidar and camera. In: International conference on autonomous unmanned systems. Springer, pp 933–944
Willis A, Sui Y (2009) An algebraic model for fast corner detection. In: 2009 IEEE 12th International Conference on Computer Vision. IEEE. https://doi.org/10.1109/iccv.2009.5459443
Xiao Z, Li H, Zhou D, Dai Y, Dai B (2017) Accurate extrinsic calibration between monocular camera and sparse 3d lidar points without markers. In: IEEE intelligent vehicles symposium (IV). IEEE. https://doi.org/10.1109/ivs.2017.7995755
Xu B, Jiang W, Shan J, Zhang J, Li L (2015) Investigation on the weighted RANSAC approaches for building roof plane segmentation from LiDAR point clouds. Remote Sens 8(1):5. https://doi.org/10.3390/rs8010005
Xu H, Lan G, Wu S, Hao Q (2019) Online intelligent calibration of cameras and LiDARs for autonomous driving systems. In: 2019 IEEE intelligent transportation systems conference (ITSC). IEEE. https://doi.org/10.1109/itsc.2019.8916872
Yaopeng L, Xiaojun G, Shaojing S, Bei S (2021) Review of a 3d lidar combined with single vision calibration. In: 2021 IEEE international conference on data science and computer application (ICDSCA). IEEE. https://doi.org/10.1109/icdsca53499.2021.9649726
Ye C, Pan H, Gao H (2022) Keypoint-based LiDAR-camera online calibration with robust geometric network. IEEE Trans Instrum Meas 71:1–11. https://doi.org/10.1109/tim.2021.3129882
Yoo J-S, Kim D-H, Kim G-W (2018) Improved lidar-camera calibration using marker detection based on 3d plane extraction. J Electr Eng Technol 13(6):2530–2544

Yu C, Wang J, Peng C, Gao C, Yu G, Sang N (2018) BiSeNet: bilateral segmentation network for real-time semantic segmentation. In: Computer vision – ECCV 2018. Springer, pp 334–349. https://doi.org/10.1007/978-3-030-01261-8_20

Yu H, Zhen W, Yang W, Scherer S (2020) Line-based 2-d-3-d registration and camera localization in structured environments. IEEE Trans Instrum Meas 69(11):8962–8972. https://doi.org/10.1109/tim.2020.2999137

Yuan K, Guo Z, Wang ZJ (2020) RGGNet: tolerance aware LiDAR-camera online calibration with geometric deep learning and generative model. IEEE Robot Autom Lett 5(4):6956–6963. https://doi.org/10.1109/lra.2020.3026958

Yuan C, Liu X, Hong X, Zhang F (2021) Pixel-level extrinsic self calibration of high resolution LiDAR and camera in targetless environments. IEEE Robot Autom Lett 6(4):7517–7524. https://doi.org/10.1109/lra.2021.3098923

Zhang Q, Pless R (2004) Extrinsic calibration of a camera and laser range finder (improves camera calibration) . In: 2004 IEEE/RSJ international conference on intelligent robots and systems (IROS) (IEEE Cat. No.04CH37566). IEEE. https://doi.org/10.1109/iros.2004.1389752

Zhang J, Singh S (2014) LOAM: Lidar odometry and mapping in real-time. In: Robotics: science and systems X. Robotics: Science and Systems Foundation. https://doi.org/10.15607/rss.2014.x.007

Zhang X, Zhang A, Meng X (2015) Automatic fusion of hyperspectral images and laser scans using feature points. J Sens 1–9:2015. https://doi.org/10.1155/2015/415361

Zhang W, Zhou H, Sun S, Wang Z, Shi J, Loy CC (2019) Robust multi-modality multi-object tracking. In: 2019 IEEE/CVF international conference on computer vision (ICCV). IEEE. https://doi.org/10.1109/iccv.2019.00245

Zhang X, Zhu S, Guo S, Li J, Liu H (2021) Line-based automatic extrinsic calibration of LiDAR and camera. In: 2021 IEEE international conference on robotics and automation (ICRA). IEEE. https://doi.org/10.1109/icra48506.2021.9561216

Zhao Y, Wang Y, Tsai Y (2016) 2d-image to 3d-range registration in urban environments via scene categorization and combination of similarity measurements. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE. https://doi.org/10.1109/icra.2016.7487332

Zhao G, Hu J, You S, Kuo CCJ (2021) CalibDNN: multimodal sensor calibration for perception using deep neural networks. In: Grewe LL, Blasch EP, Kadar I (eds) Signal processing, sensor/information fusion, and target recognition XXX. SPIE. https://doi.org/10.1117/12.2587994

Zhou L, Deng Z (2012) A new algorithm for computing the projection matrix between a LIDAR and a camera based on line correspondences. In: 2012 IV international congress on ultra modern telecommunications and control systems. IEEE. https://doi.org/10.1109/icumt.2012.6459706

Zhou Y, Qi H, Ma Y (2019) End-to-end wireframe parsing. In: 2019 IEEE/CVF international conference on computer vision (ICCV). IEEE. https://doi.org/10.1109/iccv.2019.00105

Zhu N, Jia Y, Ji S (2018) Registration of panoramic/fish-eye image sequence and LiDAR points using skyline features. Sensors 18(5):1651. https://doi.org/10.3390/s18051651

Zhu Y, Sapra K, Reda FA, Shih KJ, Newsam S, Tao A, Catanzaro B (2019) Improving semantic segmentation via video propagation and label relaxation. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2019.00906

Zhu Y, Li C, Zhang Y (2020) Online camera-LiDAR calibration with sensor semantic information. In: 2020 IEEE international conference on robotics and automation (ICRA). IEEE. https://doi.org/10.1109/icra40945.2020.9196627

Zuniga-Noel D, Ruiz-Sarmiento J-R, Gomez-Ojeda R, Gonzalez-Jimenez J (2019) Automatic multi-sensor extrinsic calibration for mobile robots. IEEE Robot Autom Lett 4(3):2862–2869. https://doi.org/10.1109/lra.2019.2922618

Zuo X, Geneva P, Lee W, Liu Y, Huang G (2019) LIC-fusion: LiDAR-Inertial-Camera Odometry. In: 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE. https://doi.org/10.1109/iros40897.2019.8967746