IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 September 2023, accepted 29 September 2023, date of publication 5 October 2023,
date of current version 11 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3322229

TOPICAL REVIEW

External Extrinsic Calibration of Multi-Modal Imaging Sensors: A Review

ZHIEN LIU[1,2], ZHENWEI CHEN[1,2], XIAOXU WEI[1,2], WAN CHEN[1,2], AND YONGSHENG WANG[3], (Member, IEEE)

[1]School of Automotive Engineering, Wuhan University of Technology, Wuhan 430063, China
[2]Hubei Key Laboratory of Advanced Technology for Automotive Components, Wuhan University of Technology, Wuhan 430070, China
[3]School of Information Engineering, Wuhan University of Technology, Wuhan 430063, China

Corresponding author: Xiaoxu Wei (wxx2014@whut.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 52175111.

ABSTRACT With the rapid development of autonomous driving, robotics, and intelligent transportation, multi-sensor-based environment sensing technology for intelligent vehicles has become a popular research direction. In order to better fuse the data acquired by multi-sensors, accurate external parameter calibration becomes one of the critical issues. According to the method of external parameter calibration, this paper first introduces the offline calibration technology based on target and targetless methods. However, once these two methods change the relative position between the camera and the LiDAR, it can only be returned to the field to re-calibrate. The computational complexity is high, which makes it necessary to use the online calibration directly. Hence, this paper follows up with the introduction of online calibration technology based on deep learning. Unlike previous methods that need to extract features from calibration boards or environments, various types of networks can directly learn the mapping relationship between images and point clouds, From the calibration results, the average error of translation and rotation of traditional methods can reach 0.34cm and 0.45°, the average error of using deep learning networks such as LCCNet, which is the most widely used in existing networks and has good calibration effect, can reach 0.297cm and 0.017°. Compared with the traditional method, the accuracy of online calibration technology is respectively improved by 12.6% and 96.2%, which shows the results of online calibration technology are better than the traditional offline method, and there are some recently proposed methods incorporate an attention mechanism and use an optimization algorithm instead of a loss function to refine the outer parameters. From the review, learning the relative relationships between sensors through neural networks works best, and the process is relatively free of human intervention. Contrary to the existing reviews, this paper provides a general structure of calibration methods universally used in various environments and compares various methods based on this general structure.

INDEX TERMS Multi-sensors, external parameter calibration, offline calibration, online calibration.

I. INTRODUCTION

The perception system in autonomous driving is one of the critical technologies for realizing self-driving capabilities. It allows vehicles to perceive the surrounding environment, make subsequent decisions, and plan driving paths by analyzing and understanding the environment. Self-driving cars rely on advanced perception systems to obtain accurate and comprehensive information about the surrounding environment. The research and development of perception systems are crucial for advancing autonomous driving technology. By continuously improving perception systems' accuracy, stability, and adaptability, we can achieve safer and more efficient intelligent driving cars, bringing significant potential benefits to society. Sensors, as the most reliable data sources in perception systems, enable real-time understanding of road conditions, obstacle positions, and motion states through data fusion from different sensors, such as sensor-level and

The associate editor coordinating the review of this manuscript and approving it for publication was Xuebo Zhang.

feature-level fusion. This allows intelligent driving decisions, forming reliable environmental estimations that exceed the sum of individual sensor data [1].

In recent years, with the rapid development of sensor technology, data fusion of multimodal sensors has been applied in various fields such as fault detection [2], [3], remote sensing technology [4], [5], robotics [6], Simultaneous Localization and Mapping (SLAM) [7], and autonomous driving [8], achieving significant results. However, in the process of realizing these techniques, in order to get more accurate fusion data, reducing the error of the calibration parameters is one of the essential links. Multisensor calibration can be categorized into intrinsic and extrinsic [9]. Intrinsic calibration determines the internal mapping relationships of sensors, such as camera focal length, principal point coordinates, pixel spacing, etc [10]. Extrinsic calibration defines the relative position and orientation between multiple sensors. Most intrinsic parameters are provided by manufacturers or calculated, so the calibration primarily focuses on the external parameters. By accurately calibrating sensors position and attitude relationships, data from different sources can be transformed into a common reference system required for early-stage sensor fusion. This enables accurate transformation of perception data, such as conversion between the world coordinate system and the camera coordinate system, improvement in object detection and tracking accuracy, and enhancement of environmental perception and map construction reliability. Therefore, the accuracy and stability of extrinsic calibration are crucial for successfully applying autonomous driving technology.

Inspired by the tremendous success of data fusion, various types of sensors have also witnessed rapid development in the past few decades, primarily used for detection, segmentation, recognition, and other purposes [11]. According to the data source [12], External sensors can collect data about the external environment in which the smart car is located, such as on-board cameras, millimeter wave radar, LiDAR, etc. Cameras can provide rich visual information including color, texture and shape through light reflection, such as monocular cameras, binocular cameras, depth cameras, etc [13], [14]. On the other hand, LiDAR utilizes laser pulses to measure the time or phase differences of the returning beams, providing high-precision distance and geometric information [15], with examples like HDL-64E and MEMS LiDAR. The combination of cameras and LiDAR has wide applications in various scenarios and fields [16], playing a critical role in perception tasks. Moreover, as advanced sensor configurations, they offer complementary information, enabling better data fusion and enhancing system decision-making and behavior.

Due to the use of multi-sensors and the proliferation of various methods for externally parameters calibration, it is becoming increasingly challenging to keep up with new advances in neural network updates. Moreover, collecting a comprehensive review focused explicitly on extrinsic calibration is difficult. Therefore, there is an urgent need to conduct a comprehensive review of existing work and discuss potential directions for future improvements, it would greatly benefit the community. In this paper, we focus on an overview of the methodology for external parameters calibration between cameras and LiDARs. To facilitate future research in different subtopics, we categorize them based on feature-based or learning-based approaches, including target-based extrinsic calibration, targetless extrinsic calibration, and deep learning-based online calibration. Target-based and targetless extrinsic calibration are collectively referred to as feature-based extrinsic calibration, it primarily involves calibration based on checkerboard patterns or unique environmental features. Deep learning-based online calibration [17] criteria focus on learning mapping relationships of outputs through neural network models.

The organization of this paper is as follows. Section II discusses the prerequisites and preparations for multisensor extrinsic calibration. Sections III, IV, and V form the core of this paper, where we summarize techniques related to feature-based calibration using calibration boards, environmental features, and various deep-learning models. We also briefly introduce the recently proposed ATOP methods, which are closely related to our topic and in line with the latest technological trends. In the final section, we provide conclusions, discuss future research directions, and address challenges. Throughout this survey, We analyze the methodology primarily through representative works (early, seminal, novel, or illuminating works) and strive to provide detailed coverage of various current techniques within the constraints of limited pages.

II. PREREQUISITES AND PREPARATIONS

This section first discusses the prerequisites for calibrating external multimodal imaging sensors, including data acquisition and processing, time synchronization, calibration algorithms and tools, and motion compensation. These conditions play a crucial role in achieving accurate calibration. For example, data must be collected starting at a unified time, and the accuracy of obtaining intrinsic parameters is crucial. Otherwise, there will be data clutter and redundancy. Even if an appropriate extrinsic calibration method is used later, it will yield results far from the ground truth. Furthermore, the principles of coordinate system transformation are explained, including the world, camera, and image coordinate systems. These are essential concepts in multimodal sensor calibration, ensuring that sensor data can be transformed into a unified coordinate system for alignment and registration.

A. PREREQUISITES

In the process of external parameter calibration, in order to make the external parameter calibration results more reliable and constantly close to the ground truth, the following prior conditions also determine the calibration results.

1) DATA ACQUISITION SYNCHRONIZATION

Feature comparison and matching is an important component in the calibration process and is needed to ensure that all sensors are synchronized in time. This means that all sensors collect data at the same time and that the timestamps of the data can be aligned with each other [18], then they can be matched and compared in the subsequent calibration process. There are several methods, such as hardware triggering, time synchronization protocols, and software interpolation.

Lixin et al. [19] proposed using hardware signals or triggers to synchronize the operation of data acquisition devices. Specifically, by sending a synchronization signal between the devices, it can be ensured that they start collecting data simultaneously. This method requires that the devices have hardware interfaces or synchronization signal lines between them, and that the devices are able to respond to external trigger signals. However, due to the different sampling frequencies of each sensor, there is still a delay in data transmission between sensors, and to solve such a problem, they describe a simple synchronization protocol that improves stability in various experimental situations.

They observed the desirability of time synchronization of devices using time synchronization protocols such as NTP, PTP, etc., they use networks or dedicated hardware devices to ensure that the clocks of individual devices remain synchronized. High time accuracy and stability can be achieved by connecting devices to a time server or by time synchronization over the Internet. However, this technique only manages to mitigate delays, and if the time server itself is in error, it may still result in delays in data transmission.

In addition to the methods mentioned above, Römer et al. [20] proposed another approach for devices lacking hardware synchronization capabilities. This method involves using time interpolation to achieve time synchronization. Specifically, it involves recording each device's sampling time and frequency information, using interpolation algorithms to infer the timestamps for each device, ultimately fitting them into a time curve. This method needs to consider sampling frequency differences and delays between devices to obtain more accurate time synchronization results. However, if the frame rate is unstable, more sophisticated schemes such as passive synchronization [21] are required.

In this section, by outlining the prerequisites of the calibration process and comparing various types of methods for synchronizing data, we found that these works initially explored the potential of applying network communication and clock synchronization algorithms to data acquisition. However, relatively little research has been done on time interpolation and passive synchronization. In future research, it is possible to explore how to combine the hardware synchronization function with time interpolation, and propose more effective clock synchronization algorithms.

2) ACCURATE INTRINSIC PARAMETERS OF CAMERA AND LIDAR

In the process of outputting the results of the external parameters, it can be seen from Equation (5), it is necessary to use accurate values of the internal parameters to carry out the geometric correction and coordinate conversion of the camera, so the accuracy of the external parameters calibration results is largely dependent on the accuracy of the internal parameters. The internal parameters are the internal parameters of the camera, including the focal length, principal point coordinates and aberration parameters, describe the imaging characteristics and geometric aberrations of the camera. According to Zhang [22], Accurate internal parameters can provide accurate image geometric information, so that the external parameter calibration can more accurately calculate the position and attitude of the camera and the relationship between the camera and the scene, e.g., an accurate internal parameters can eliminate image aberrations, correct the image coordinates and pixel scales, and thus improve the accuracy and reliability of the external parameters calibration.

3) MOTION DISTANCE COMPENSATION

In addition to the methods mentioned above, motion distance compensation and other techniques play a significant role in determining the accuracy of the extrinsic parameters. In motion distance compensation, commonly used methods involve utilizing the accelerometer and gyroscope data from an Inertial Measurement Unit (IMU) to estimate the object's attitude changes, and then correct the sensor data. Furthermore, iterative closest point (ICP) algorithms such as the Velocity Iterative Closest Point (VICP) algorithm proposed by Hong et al. [23], and the Approximate Nearest Neighbor (ANN) algorithm proposed by Liu et al. [24] can utilize visual or LiDAR data for feature matching and point cloud registration. In recent years, motion compensation has played a vital role in various fields, such as robot navigation, virtual reality, augmented reality, and motion analysis. Accurate motion compensation can improve subsequent data processing and application outcomes.

This section provides insights into the preparation of prerequisites for future calibrations, where the only way to obtain ideal data is to choose a suitable calibration method that satisfies the prerequisites as much as possible. In the future, more attention should be paid to the prerequisites section to consider how to ensure that the data are obtained more realistically and reliably.

B. FUNDAMENTALS OF EXTRINSIC CALIBRATION

An object in three-dimensional space puts light into the camera's sensor by reflecting or bouncing light, and the camera obtains an image of the object by saving the data information on the sensor. In the imaging process, the spatial transformation is divided into several different coordinate systems. The points in the three-dimensional space are

projected onto the image plane to form two-dimensional pixels, so the imaging process can be simplified as the conversion process from the world coordinate system to the camera coordinate system, from the camera coordinate system to the image coordinate system. From the image coordinate system to the pixel coordinate system [25], and the specific relationship between the conversion diagrams are shown in Figure 1.

First of all, to clarify the conversion relationship from the world coordinate system to the camera coordinate system, when the coordinate system is rotated by θ along the z-axis, only the values in the x-axis and y-axis direction will change due to the change of angle. Similarly, when the coordinate system is simultaneously rotated by ω along the x-axis, rotated by ψ along the y-axis, and rotated and translated along the z-axis, the rotation matrix (denoted as R) is as follows [26]:

$$\begin{aligned} R &= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & \sin\omega \\ 0 & -\sin\omega & \cos\omega \end{bmatrix} \\ &\quad \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & \sin\omega \\ 0 & -\sin\omega & \cos\omega \end{bmatrix} \\ &= \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \end{aligned} \tag{1}$$

The translation matrix is:

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{2}$$

Therefore, we can obtain the homogeneous coordinates for the transformation from the world coordinate system to the camera coordinate system as follows:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{3}$$

where R refers to the rotation matrix and T refers to the translation matrix, which represents the external parameter of the transformation between the world coordinate system and the camera coordinate system. Secondly, in order to get the principle of the camera space transformation relationship, Hartley and Zisserman [27] using the pinhole imaging technology and the similar triangle relationship, the relationship between the image coordinate system and the pixel coordinate system is:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4}$$

Finally, the transformation relationship between the world coordinate system and the pixel coordinate system is obtained by using the above Equation (3) and (4):

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{5}$$

The logic behind Equation (5) is not difficult to understand. Set

$$K_1 = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{6}$$

$$K_2 = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \tag{7}$$

where K_2 is the external parameter of the camera, which is also a parameter to be solved, closely related to the relative position of the camera and the lidar, and K_1 is the internal parameter of the camera, which is only related to the interior of the camera. Since this paper focuses on the external space conversion parameters between the lidar and the camera, it is assumed that there are accurate internal parameters of the camera.

This section outlines the relative relation of coordinate systems in space transformation. In the subsequent work, calibration methods suitable for various environments should be constantly selected to obtain the parameters in Equation (5), and finally, the external parameters should be smoothly returned.

III. TARGET-BASED CALIBRATION TECHNIQUES

A. CALIBRATION BOARD

So far, there have been many toolkits for calibrating LiDARs and cameras, including the Autoware [28], the Apollo [29], and the "LiDAR and Camera Calibration Toolbox" [30]. 3D LiDAR can emit and receive multiple laser beams, allowing for richer depth information, such as point normals [31]. 3D LIDAR and camera calibration involve a six-degree-of-freedom (DOF) problem with six unknowns. Each observation of the board provides three constraints. Since calibration over multiple observations can impact accuracy, researchers have recently begun to pay attention to how to obtain a more significant number of constraints in a single shot. Different studies have evaluated the effect of different calibration targets. A calibration board is an artifact used for sensor calibration in a calibration target. It is usually a flat or flat approximation of an object with a specific geometry, pattern, or feature on the surface. The *calibrationboard* is essential in the sensor calibration process. It is used as a reference object to measure the correspondence between the image or point cloud observed by the sensor and the actual geometry.

Checkerboard or hole-patterned checkerboards, and their combinations, are the most common designs for offline calibration boards [32]. Table 1 compares the characteristics of some typical calibration boards. As shown in Table 1, typical features such as checkerboards have accurately

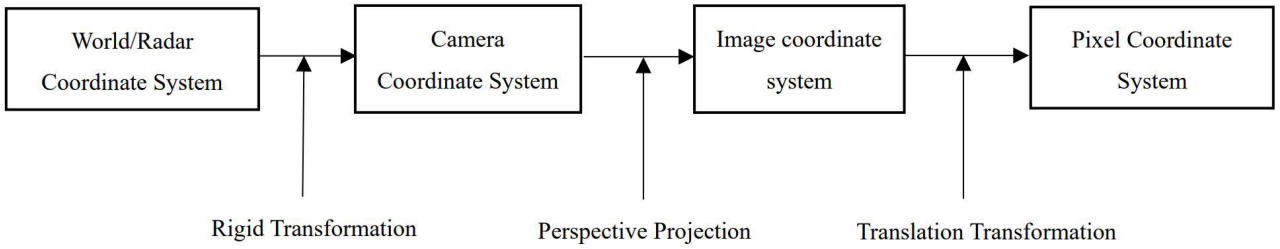

FIGURE 1. Coordinate system transformation relationship.

detectable corners and intersections, and they offer high flexibility [33], [34], [35]. Additionally, Zhao et al. [36] used a calibration board composed of a rectangular plane and ArUco markers for LiDAR-stereo camera calibration. Domhof et al. [37] proposed an object that both LiDAR and cameras can observe for calibration. Peršić et al. [38] constructed another calibration target for the joint calibration of LiDAR and cameras. The extrinsic parameters can be determined based on constraints from points and lines on these targets.

The table above summarizes different calibration targets and their characteristics. We found that most calibration targets, such as checkerboards and ArUco markers, are composed of black-and-white alternating, regionally distinct color blocks. Some calibration targets have prominent geometric features on the outer edges, greatly facilitating feature detection. But at present, some researchers in order to overcome the traditional chessboard under the strong ring light may produce uneven illumination on the calibration target, leading to the failure of ordinary checkerboard detection, An et al. [48] proposed using CharuCo as an alternative, which can still be used for detecting the remaining saddle points. In the future, we expect to see specific and novel target boards emerging in 2D extrinsic parameter calibration.

B. TWO-DIMENSIONAL TARGET CALIBRATION METHODS

Figure 2 shows the general process of target-based calibration, which can be used as the general structure of the off-line external parameter calibration process. Mishra et al. [49] proposed the most classical chessboard calibration method for two-dimensional targets, mainly consisting of feature extraction, feature matching, and regression of transformation matrices.

Firstly, point cloud data and RGB image features are extracted. Since calibration boards or targets exist in the scene, automatic detection and recognition of chessboard corners can be achieved in the image. The main methods for corner detection include Harris corner detection [50], SURF corner detection [51], SIFT corner detection [52], and other corner detection algorithms and corner descriptors. The principle of the detector is based on the Hessian matrix. Given a point in the image I, the Hessian matrix is defined as shown in Equation (8) [51]. $L_{xx}(x, \sigma)$ is the convolution of the Gaussian second derivative $\frac{d^2 g(\sigma)}{dx^2}$ on the point x of the image I. The same is true for $L_{xy}(x, \sigma)$ and $L_{yy}(x, \sigma)$. The position of corner points is determined by calculating the response function of each pixel. This response function measures the change of local pixel intensity by the sum of squares of the difference after a small translation of the gray value in the window around the pixel. It has the advantage of simplicity and efficiency and has certain invariance to the scale and rotation changes of the image. Therefore, it is

TABLE 1. Characteristics of typical calibration targets.

Calibration Target	Characteristics	Indexing literature
Checkerboard	Applicable to both 3D and 2D LiDAR.By combining 3D lines and planes,the minimum number of observations is reduced to a panoramic image. Improved using the Levenberg-Marquardt method.	Pandey et al. [39] Zhou et al. [40]
Planar targets with holes	A plane with circular or triangular holes achieves edge detection by detecting the depth differences between adjacent points on the same line.	Beltrán et al. [41] Yan et al. [42]
Multi-plane checkerboard	Multiple checkerboard patterns on different planes.The camera can observe the checkerboard from various angles and directions, which helps to compensate for occlusions and improves the robustness of the calibration process	Geiger et al. [43]
Polygonal Planar Board	The polygonal plane board consists of multiple sides and angles. Its diverse shapes make it suitable for different application scenarios and requirements. The geometric shape of the board is stable, and the relative positions between its sides and angles remain unchanged.	Park et al. [44]
ArUco Markers	The polygonal plane board consists of multiple sides and angles. Its diverse shapes make it suitable for different application scenarios and requirements. The geometric shape of the board is stable, and the relative positions between its sides and angles remain unchanged.	Povendhan et al. [45] Zamanakos et al. [46]
Circular grid	A circular grid is composed of a series of equidistantly arranged circular cells. These cells can be concentric circles,circles with equal radii, or other regular circular shapes. They can be observed by cameras or other vision sensors from different viewpoints and distances.	Domhof et al. [47]

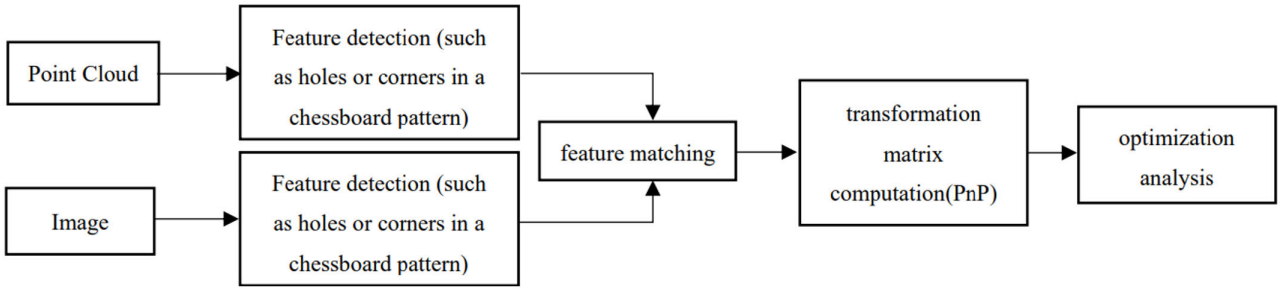

FIGURE 2. General process of target calibration techniques relationship.

widely used in feature extraction and target tracking.

$$H(x,\sigma)=\begin{bmatrix}L_{xx}(x,\sigma)\ L_{xy}(x,\sigma)\\ L_{xy}(x,\sigma)\ L_{yy}(x,\sigma)\end{bmatrix} \tag{8}$$

Secondly, for feature matching, as early as a decade ago, Zhao et al. [53] and Scaramuzza et al. [54] proposed a manual selection of 2D image features of the target board and corresponding 3D target points for matching. However, this method resulted in low accuracy and needed to be more time-consuming. After repeated experiments, some ideas of manual matching were acquired, Raguram et al. [55] proposed a method based on geometric constraints, and Nguyen et al. [56] proposed a method based on mutual information for feature matching. These methods mainly use geometric constraints or similarity between descriptors to filter and validate the consistency of matched point pairs. Compared to manual matching, these two methods achieved relatively more automation. Since the general process is similar for both target-present and target-absent cases, we will explain the ideas of feature matching in detail in the fourth section.

Finally, the transformation matrix calculation is widely applied, and the most accurate regression method is the PnP (Perspective-n-Point) computation method [57]. The PnP problem is a nonlinear optimization problem, and the objective is to estimate the camera's rotation and translation transformations using known correspondences between 3D points and their corresponding 2D image points (generally at least three pairs) [58]. Its mathematical model is shown in the following formula (9) [58], where p is the coordinates of the point in the pixel coordinate system, P^C is the coordinates of the point in the camera coordinate system, P^W is the coordinates of the point in the pixel coordinate system, ω is the depth of the point, K is the internal parameter matrix of the camera, R_{CW} and t_{CW}^C is the pose transformation from the world coordinate system to the camera coordinate system. Solution methods, such as P3P, involve using three pairs of known 3D coordinates in the world coordinate system, and the corresponding 2D coordinates of points projected onto the camera's normalized plane. The 3D coordinates of the three points in the camera coordinate system can be obtained by constructing equations. The problem is then transformed into a 3D-3D ICP (Iterative Closest Point) problem. Since solving the 3D-3D pose with matching information is straightforward, this method is effective. However, when there are more than 3 groups of matching points, the spatial information points cannot be fully utilized.

$$\omega p = KP^C = K(R_{CW}P^W + t_{CW}^C) \tag{9}$$

Like P3P, Lepetit et al. [59] proposed the EPnP algorithm, it utilizes more spatial point information and optimizes the camera pose iteratively to minimize the influence of noisy points compared to P3P. The main idea of both methods is to obtain the coordinates of the corresponding points in the camera coordinate system, converting the 3D-2D problem into a 3D-3D problem and then solving it using ICP.

After solving the PnP problem, the matched point pairs need to be further optimized to regress the specific external parameter values. The specific methods mainly include those based on linear equation-solving [60], [61], iterative optimization methods (such as the Levenberg-Marquardt algorithm) [62], and methods based on RANSAC [59]. Among them, the RANSAC method proposed by Jian and Vemuri [63] is currently the most popular approach. In addition to selecting the appropriate data and model, the selection process of the number of traversals is derived from the following Equation (10) [63], where p represents the probability that the RANSAC algorithm results are useful, and w is the probability that the data will be in the set of internal points, so the result of k iterations satisfies the Equation (11) [63]. Specifically, researchers project the feature points onto the model obtained by PnP and calculate their reprojection error. Feature points with reprojection errors below a certain threshold are considered inliers, while others are considered outliers. PnP replaces the complexity of manual matching, It is commonly used in applications such as camera pose estimation, camera calibration, and 3D object pose estimation.

$$k = \frac{log(1-p)}{log(1-w^n)} \tag{10}$$

$$1 - p = (1 - w^n)^k \tag{11}$$

The above Outlines the general calibration process of the ordinary checkerboard. Compared with the ordinary checkerboard, which only has the information of points and edges, the checkerboard with holes increases the feature of depth discontinuity. When the depth changes beyond

the threshold, it is more likely to be the edge. Because of its more distinctive characteristics, it has been widely used in recent years. For example, Beltrán et al. [41] and Vel'as et al. [64] proposed a chessboard pattern with circular holes. They perform plane segmentation, target detection, circle segmentation, and reference point estimation on the point cloud data, use the depth discontinuity detection method to obtain the hole contour, and perform ArUco mark detection and 3D pose estimation of the target on the image.

Figure 3 shows the schematic diagram of different stages of reference point estimation. By finding the vertical plane and setting the threshold, the non-plane points are removed for the first time, and the point cloud is applied to the edge point cloud to remove all points that are not plane points. Then 3D-2D projection, while searching for the corresponding circle, remove the inner point, and extract the rectangular feature of the circle. If there is no matching center point, this frame is abandoned. This method pioneered depth information for calibration plates, after which researchers used depth continuity or discontinuity to extract edges fully.

Apart from circular holes providing depth information, chessboard patterns with holes come in various shapes. For instance, Cai et al. [65] designed a novel checkerboard; specifically, it has local gradient depth information and central plane square corner information, select feature points, and obtain their corresponding coordinates in the LiDAR and camera pixel coordinate systems. The following Equation (12) [65] is the coordinate solution of feature point coordinates in the coordinate system of the calibration board, and then the coordinate solution of the coordinate system of the Lidar and camera, where x_{m1}, x_{m2}, x_{m3}, x_{m4} represents the coordinate of feature points, $|MM|$ represents the distance between two points, and L represents collinear conditions. Secondly, the calibration experiment is carried out, and the calibration results are verified by incremental verification and reprojection error comparison. In addition to the mentioned methods, there are other approaches for edge detection using triangle-shaped holes [66] or rectangular-shaped holes [47].

$$\begin{aligned}
&\sqrt{(x_{m1}-x_{m2})^2+(x_{m1}+x_{m2}-10L)^2}=|M_1M_2|\\
&\sqrt{(x_{m2}-x_{m3})^2+(-x_{m2}-x_{m3}+L)^2}=|M_2M_3|\\
&\sqrt{(x_{m3}-x_{m4})^2+(-x_{m3}+x_{m4}-24L)^2}=|M_3M_4|\\
&\frac{x_{m1}-x_{m2}}{x_{m1}+x_{m2}-10L}=\frac{x_{m1}-x_{m3}}{x_{m1}+x_{m3}+7L}
\end{aligned} \tag{12}$$

The above method is mainly for the experimental conditions with fewer planes and simple environment, and has certain limitations in the use conditions. To adapt to different calibration requirements and special scenes, multi-plane checkerboards can be placed at different angles and positions to obtain calibration data from multiple perspectives. The biggest advantage of multi-plane checkerboard is that even if some planes are blocked or have problems, it can still use the information of other planes for calibration. This work proves the benefits of multi-planes in harsh environments. As shown in Figure 4, the placement of the multi-plane checkerboard and the test results of its corners.

Geiger et al. [43] proposed using four filter kernels to exclude corner points. If any of the four filter kernels respond weakly, it indicates a lower likelihood of being a corner point. This method is crucial for removing non-chessboard pattern corners from the hypothesis space as much as possible. However, there are some problems with the layout of this method, and the checkerboard layout is relatively complex, especially in large-scale measurement systems, which may require more time and effort to accurately place the calibration board. Therefore, in future development, how to ensure the detection effect and lightweight the calibration board in the case of multi-plane placement has become the key point of future optimization.

These works prove that different 2D calibration plates can be applied in different environments to improve the performance of various vision tasks. They make full use of the information of corner points and edge points on the calibration board to extract features, match features by geometric or mutual information, and finally solve the PnP algorithm to return the external parameters. From the overview in this section, it can be concluded that although the two-dimensional target is the most traditional calibration method, it has some innovations in recent years, such as multi-plane polygon, etc., but its performance is still not as good as that of three-dimensional and non-target calibration methods discussed in Section IV. One of the main reasons is that since 2D calibration targets may be more susceptible to occlusion, lighting changes, and image distortion, we also expect better 2D architectures in the future.

C. THREE-DIMENSIONAL TARGET CALIBRATION METHODS

The sphere target is the most commonly used 3D target in 3D object calibration, which is suitable for low-resolution LiDAR data and can be detected from different angles. Tóth et al. [67] and Kümmerle et al. [68] proposed using spheres as calibration targets because the surface of a sphere can be accurately detected in point cloud data. In contrast, its contour can be precisely detected in camera images. The calibration problem is defined as shown in the following Equation (13) [68], and the transformation T is found to minimize the sum of the squares of the distances of all observations to P in the reference frame, where $T = T_1, T_2, \ldots\ldots, T_n$ defines the pose of the sensor in the reference frame, and $P = P_1, P_2, \ldots\ldots, P_m$ is the observation of m pairs of time synchronization. Figure 5 shows the relative position relationship between the camera based on sphere detection and the Lidar. The sphere's center point is calculated from the Lidar point cloud and the image respectively. The center point of the ellipse is determined by RANSAC and LSQ regression to the point cloud to obtain the center point of the sphere further. However, compared with the two-dimensional target board, the complexity of its

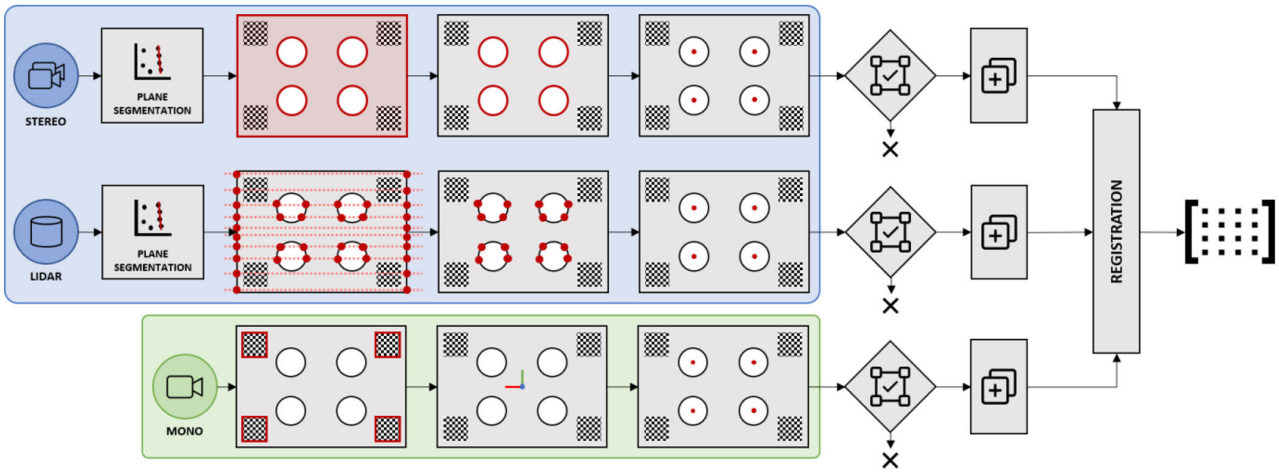

FIGURE 3. The detection process for reference point estimation relationship [41].

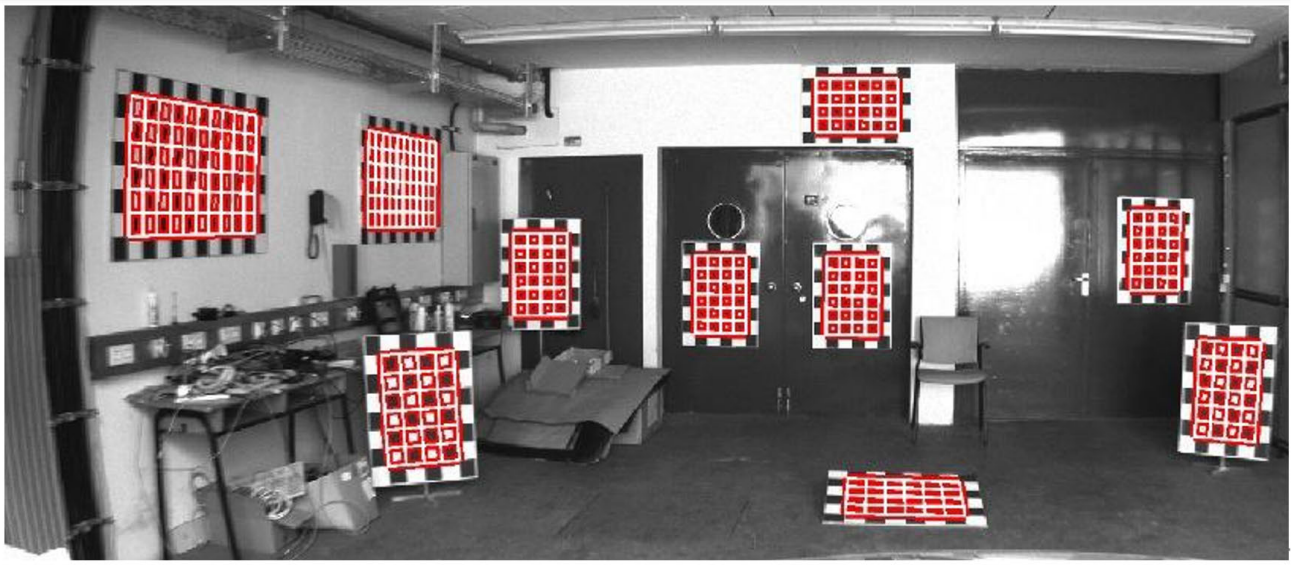

FIGURE 4. Corner detection for multiple plane chessboard patterns [43].

operation is also apparent.

$$\arg\min_{T} \sum_{i=1}^{m} dist(P_i, T)^2 \tag{13}$$

In addition to using 2D and 3D calibration boards separately, there are also improvements to combining 2D and 3D targets. These methods all utilize the common structure of targeted calibration methods as shown in Figure 2. In this section, by comparing different 2D and 3D targets and different calibration methods, we find that the traditional target-based off-line external parameter calibration technology still has certain applicability in improving accuracy nowadays, such as in relatively simple indoor environments. However, when the environment is slightly complex, there are some defects in wrong feature extraction and initial value limitation.

Similarly, researchers have learned from target-based calibration methods and can successfully extract features in the environment. Therefore, with the development of technology, in recent years, physical objects have been gradually changed to target features in the environment, which not only eliminates the difficulty of arranging scenes, but also simplifies the calibration method. This paper will be introduced in Section IV.

IV. TARGETLESS-BASED CALIBRATION TECHNIQUES

Some current studies learn from the experience of target-based work and find that the target plate has a relatively complex calculation defect, so they put forward the targetless calibration technology, the overall calibration process is similar to the target-based method, and the general structure of target-based calibration pointed out in Section III is also applicable in this section. However, different from the target-based calibration method in Section III, the targetless calibration method avoids the use of geometric components such as calibration plates but extracts features directly from

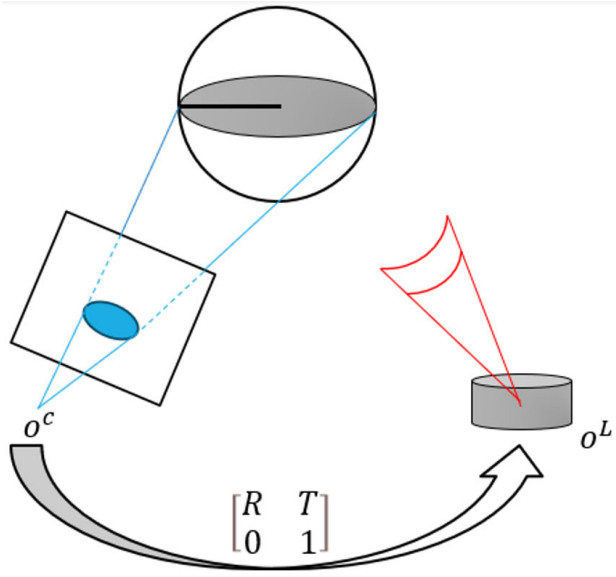

FIGURE 5. Estimation of relative attitude between camera and LiDAR device.

the natural scene [69], including geometric structures like edge lines, corners, and other characteristics [70]. It is worth noting that it does not depend on the real object in the scene, which makes the targetless calibration method more general and flexible, and can be applied to various scenarios and applications. Specifically, it is divided into information-based, feature-based, and self-motion-based methods [71].

Below, for each class method, we introduce the fundamentals of the method and further explore the differences between them by specifying multiple implementation options. At the same time, it is pointed out that the future development of targetless technology should pay more attention to these three methods, and it is expected that more reliable improvement schemes will appear in the future.

A. INFORMATION-BASED METHODS

External parameter calibration excelses in targetless-based methods, especially information-based methods, which estimate external parameters primarily by maximizing the similarity transformation between the liDAR sensor and the camera, where similarity is measured by various information measures. Specifically, the method based on information theory consists of three steps, the first is the 3D-2D projection of lidar points, It is the projection of three-dimensional point cloud data into the image. The second step is a statistical similarity measure, which measures the similarity between the 2D projected image and the camera image based on some features that share a similar distribution between the sensor data acquired by the LiDAR and the camera. Finally, the measure function is optimized, and the global optimal solution is obtained by using the optimization algorithm for the non-convex function of similarity.

There are various measures of similarity. For example, Pandey et al. [72] and Wang et al. [73] consider that highly reflective LiDAR data usually correspond to bright areas in the projected image. Thus, reflectivity can be chosen as the similarity index. Additionally, Taylor and Nieto [74] suggests using the coincidence of LiDAR data with normal vectors or gradient information in the image as similarity indicators. Different methods can choose different shared features. Below, we provide a detailed introduction to commonly used attribute pairs.

Reflectivity and grayscale intensity are the most widely used attribute pairs. Grayscale intensity represents the brightness values of an image, where LiDAR points with higher reflectivity correspond to higher pixel values in the image, and points with lower reflectivity correspond to lower pixel values in the image. Similarity measurement can be conducted by comparing the consistency between reflectivity and grayscale intensity. The measurement of both attributes mainly depends on the surface properties of objects [72]. In addition to reflectivity, other attribute pairs can also be compared. For example, Zhao et al. [75] proposed using reflectivity and hue, and Irie et al. [76] suggested using reflectivity and color. All of these can serve as indicators for the subsequent statistical similarity measurement.

Although the above attribute pairs with reflectivity are straightforward, they are relative values inferred from the image rather than directly measured physical quantities. This implies that they may be affected by environmental conditions and lighting changes. To address the issues related to reflectivity combinations, Jiang et al. [77] proposed using 3D and 2D semantic information as attribute pairs between 3D point clouds and 2D images. 3D semantic labels represent the semantic categories of the point cloud, while 2D semantic labels represent the semantic categories of pixels in the image. By associating 3D semantic labels with corresponding 2D semantic labels through semantic segmentation methods [78], [79], a correspondence relationship is established. Moreover, this correlation between 3D and 2D semantic labels can play a significant role in various computer vision tasks, such as scene understanding and object detection. It enables a better understanding and analysis of scenes and supports various visual applications.

According to relevant information, combining 3D-2D attribute pairs can provide more comprehensive information. In some methods currently used, it is shown that the combination of multiple features is conducive to improving the robustness of the algorithm to different environments [76], It is because even if one part of the attribute pairs in the combination is unstable under the influence of the environment, the other part can work with complementary information. This demonstrates the reliability of the combined properties. For example, suppose we choose to combine 3D-2D semantic information with other attributes. In that case, another set can also be selected as reflectivity, surface normals, gradient information, etc., to adapt and process the data in different environments. With technology development and in-depth research, the combined attribute pair will continue improving

and perfecting to provide more accurate, efficient, and reliable solutions.

After selecting the attribute pairs for LiDAR and camera, we find that larger metric values indicate better correspondence. Next, we need to use non-convex functions to statistically measure the similarity, which is challenging because it requires suitable functions to evaluate correspondence quality. The most common method currently is based on mutual information, proposed by Shannon [80], which measures the statistical dependency between two random variables on a given attribute pair. It can capture arbitrary relationships between variables, including non-linear relationships. The mathematical model of mutual information (MI) is defined as the following Equation (14) [80], where $H(X)$ and $H(Y)$ are the individual entropy of random variables X and Y, $H(X, Y)$ is the joint entropy of the two random variables, $p_X(x)$,$p_Y(y)$ and $p_{XY}(xy)$ represents the edge probability and joint probability respectively.

$$\begin{aligned} MI(X, Y) &= H(X) + H(Y) - H(X, Y) \\ H(X) &= -\sum_{x \in X} p_X(x) log p_X(x) \\ H(Y) &= -\sum_{y \in Y} p_Y(y) log p_Y(y) \\ H(X, Y) &= -\sum_{x \in X} \sum_{y \in Y} p_{XY}(x, y) log p_{XY}(x, y) \end{aligned} \tag{14}$$

In order to prevent the mutual information method from being affected by the total amount of information, Li et al. [82] proposed normalized information distance, alleviates the problem by standardizing the variables in it. The NMI model can be defined as the following Equation (15) [82], whose physical meaning is similar to that of MI, mainly because the variable values are different.However, besides mutual information, there are many other statistical methods in practical applications. For example, Guislain et al. [83] suggested using mutual information and distance of gradient histograms. These methods fully utilize the similarity or distance of matching points for measurement. After multiple scene tests, we found that while these methods effectively express the correlation between variables, they may still suffer from issues such as dimensional disaster and noise sensitivity.

$$NMI(X, Y) = \frac{H(X) + H(Y)}{H(X, Y)} \tag{15}$$

Finally, we must use optimization algorithms to solve the global optimal solution for non-convex functions. According to the existing research data, the particle swarm optimization (PSO) proposed by Shami et al. [84] is a heuristic optimization algorithm, which regards the solution space of the problem as the search space of the particle swarm. Each particle represents a potential solution and searches for the optimal solution by continuously adjusting its velocity and position. The core idea of the algorithm is to allow particles to update and adjust themselves based on individual and collective experiences.The following Equation (16) [84] is the core modeling part of the particle swarm optimization algorithm. The first part represents the trust of the particle to the previous state of its own motion; the second part represents the distance and direction between the current position of the particle and its own historical optimal position; and the third part represents the distance and direction between the current position of the particle and the historical optimal position of the group.

$$v_{id}^{k+1} = \omega v_{id}^{k} + c_1 r_1 (p_{id,pbest}^{k} - x_{id}^{k}) + c_2 r_2 (p_{d,gbest}^{k} - x_{id}^{k}) \tag{16}$$

Similarly, other optimization algorithms, such as the gradient descent method proposed by Ruder [85],determine the updating direction of parameters according to the gradient direction of the objective function. The gradient indicates the rate of change of the objective function at a particular point, pointing toward the fastest increase in the function value. Therefore, by iteratively adjusting the parameters in the direction of negative gradient, the global optimal solution can be gradually approached. The following Equation (17) [85] should be repeated in gradient descent until the loss function converges, where w represents the initial weight value, w_{i+1} represents the updated weight, and a represents the learning rate, which must be an appropriate value.

$$w_{i+1} = w_i - \alpha * \frac{dL}{dw_i} \tag{17}$$

Additionally, Newton's method can be used to solve the minimization problem of the objective function. It is an iterative method that uses the objective function's second derivative (Hessian matrix) to update parameters, thus gradually approaching the optimal solution. However, while Newton's method has the advantage of fast convergence, it may encounter numerical stability issues when performing matrix inversion operations. To address this limitation, subsequent papers such as Nocedal and Wright [86] proposed Conjugate Newton's method, and Kelley [87] suggested Quasi-Newton methods to ensure numerical stability. Table 2 summarizes some information-based calibration methods, provides an overview of these methods based on the corresponding LiDAR attributes, image attributes, information measurement, and optimization methods.

This section provides an overview of information-based targetless calibration methods, which infer the external attitude of the camera or sensor by analyzing features and relationships in the image or sensor data. By comparing the methods in the existing literature, we find that no special calibration board or target is needed, and there are many attribute pairs or combinations of attribute pairs to choose from, which is easy to calculate and has certain flexibility. However, it also has some limitations, such as accuracy depends on data quality, and properties such as reflectivity and gray intensity are more dependent on the environment.

Future research will depend on the progress of technology, changes in application requirements, and the efforts of

TABLE 2. Information-based targetless calibration techniques.

Methods	The selected attribute pairs	Information measurement methods	Optimization algorithm
Igelbrink et al. [88]	Reflectivity and Grayscale intensity	Standardize mutual information	Nelder-Mead downhill simplex method
Pascoe et al. [89]	Reflectivity and Visible light wavelengths	Normalize information distance	BFGS quasi-Newton
Pandey et al. [90]	Reflectivity and Grayscale intensity	Standardize mutual information	Barzilai-Borwein steepest descent method
Taylor et al. [69]	Gradient magnitude and orientation and Gradient magnitude and orientation	Gradient direction metric	Particle swarm arithmetic
Guislain et al. [83]	Reflectivity–Grayscale intensity and Surface normal—Grayscale intensity	Mutual information and distance in the direction of the histogram gradient	BOBYQA algorithm

researchers, it may be necessary to combine spatial information methods and select appropriate methods according to specific scenarios and needs.

B. FEATURE-BASED METHODS

It can be seen from Section A that the information-based method requires statistical attribute similarity. Unlike the method based on information theory, the feature-based goal-free calibration method directly extracts features from the environment for matching and external parameter estimation. Specifically, these features can be divided into three categories: Geometric features, semantic features, and motion features that need to be acquired online from the LiDAR point cloud and the surrounding environment of the camera image. The general process is the same as the targeted method, divided into feature extraction, feature matching, and transformation estimation. Feature extraction aims to automatically detect stable and unique features from point clouds and images representing typical semantic or geometric features in the environment (such as various types of vehicles, telephone poles, pedestrians, etc.). Feature matching aims to provide the correspondence of extracted features, and some feature descriptors are needed to express their spatial correspondence. Transformation estimation is the external transformation parameters of LiDAR and camera based on the corresponding feature matching relationship. Singular value decomposition is a widely used algorithm to derive external parameters.

Many existing studies have separated feature matching and external pose estimation, which typically requires using two or more algorithms to conclude. To reduce workload, combining both calculations is crucial. Several recent works propose integrating feature matching and external pose estimation into a single step. For instance, Li et al. [91] proposed using differential inertial measurement units to calculate the external pose, Zhu et al. [92] proposed simplifying the procedure by translating the calibration problem into an optimization problem for a novel calibration quality measure based on semantic features, which successfully and robustly aligned a pair of time-synchronized camera and LiDAR frames directly. In addition, many researchers are currently exploring other merging methods to save computing time and look forward to the emergence of better technologies in the future. Next, we will introduce the existing research methods in three steps.

In LiDAR camera calibration, we need a pair of feature detectors for point clouds and images. Currently, there are various categories of feature extractors. We will introduce various feature extraction pairs according to the above geometric, semantic, and motion features. For example, Willis and Sui [93] proposed to use the intersections of ground plane edge contours as features. One of the important reasons is that the intersections have the characteristics of all edges at the same time, and the intersections are more real and reliable in the matching process than the points on only one contour. However, in the process of selecting points and edges, features need to be transformed by translation and rotation, and distortion and other problems will occur. Therefore, it is extremely important to introduce feature operators. For example, the scale-invariant feature transform (SIFT) proposed by Lowe [94] is a popular operator, The following Equation (18) [94] is the process in which SIFT algorithm adopts Gaussian kernel function to filter when constructing scale space. It defines $L(x, y, \sigma)$ as the convolution operation between the original image $I(x, y)$ and a variable scale 2-dimensional Gaussian function $G(x, y, \sigma)$. (x, y) represents the pixel position of the image, which is the scale space factor. A smaller value means that the image is smoothed less and the corresponding scale is smaller, it can extract features of interest points in point clouds and images. Whether the original image or the projection map is scaled, rotated or translated, the characteristics of the points will not change, and it has strong stability. With the development of technology, Some researchers later improved and accelerated the original SIFT and proposed SURF, which was faster than SIFT.

$$\begin{aligned} G(x_i, y_i, \sigma) &= \frac{1}{2\pi\sigma} exp(-\frac{(x - x_i)^2 + (y - y_i)^2}{2\sigma^2}) \\ L(x, y, \sigma) &= G(x, y, \sigma) * I(x, y) \end{aligned} \tag{18}$$

In addition to points of interest, edges are another geometric feature widely used in LiDAR camera calibration. These edges in point clouds and images contain geometric information about the environment, and edge features are particularly important in such environments when point features show instability and are insufficient to describe the environmental features.

Using depth discontinuity to extract edge information in point cloud is the most common method used by researchers at present. The main principle is to set the depth difference threshold between adjacent points, and retain the adjacent point when the depth change of the adjacent point is greater than the threshold, to filter out all points below the threshold, to achieve the purpose of extracting edge information. This idea has been widely used in various methods of point cloud edge extraction. For example Ma et al. [95] successfully extracted edge contours from point clouds by setting depth thresholds, which is an effective approach in point cloud processing.

Similar to edge extraction of point cloud, most feature extraction of image is added with various image processing operators. For example, the Sobel operator proposed by Sobel et al. [96] uses the Sobel operator to detect the edge of the image by calculating the change of the gray level of the image and calculating the gradient value of the pixel. The operator mainly consists of the following Equation (19) [96] two matrices G_x and G_y, where the P matrix is the 3*3 matrix of the image. It is very sensitive to the change of the high frequency of the image, but the noise in the image will lead to false extraction. Therefore, it is necessary to conduct pre-processing to remove noise before using Sobel operator. In addition, there are Canny edge detection [97], LSD algorithm can achieve edge extraction. These two edge extraction methods can be directly operated on the image.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * P$$
$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * P \tag{19}$$

Canny edge detection is a multi-stage edge detection algorithm, mainly including noise suppression, gradient calculation, non-maximum suppression and other steps, The basic idea is to find the position with the strongest change in gray intensity in an image, which is called the gradient direction. The following Equation (20) is the boundary gradient and direction obtained according to the first derivative G_x and G_y. Compared with Sobel operator, Canny edge detection can weaken the noise edge, has strong robustness, and is less affected by noise.

$$Edge\_Gradient(G) = \sqrt{G_x^2 + G_y^2}$$
$$Angle(\theta) = tan^{-1}(\frac{G_x}{G_y}) \tag{20}$$

On the other hand, the LSD algorithm [98] detects the line segment by analyzing the change of pixel value in the image, uses the direction and amplitude of the gradient to identify the line segment, which mainly includes the steps of image preprocessing, gradient calculation, edge direction analysis, etc. The four pixels below the right of each pixel are used for calculation to reduce the dependence on gradient, and the linear direction and gradient changes are calculated by the following formula (21) [98]. Compared with traditional methods, the realization results show that the LSD algorithm can deal with line fracture and noise interference well.

$$g_x(x, y) = \frac{i(x+1,y)+i(x+1,y+1)-i(x,y)-i(x,y+1)}{2}$$
$$g_y(x, y) = \frac{i(x,y+1)+i(x+1,y+1)-i(x,y)-i(x+1,y)}{2}$$
$$\arctan\left(\frac{g_x(x,y)}{-g_y(x,y)}\right)$$
$$G(x, y) = \sqrt{g_x^2(x, y) + g_y^2(x, y)} \tag{21}$$

The above mainly outlines the extraction methods of points and edges, but in recent years, with the development of technology, there are also features extracted according to the trajectory of the same object, semantic attributes obtained by semantic segmentation, skyline poles, etc. For example, Peršić et al. [99] proposed an unrelated calibration trajectory association method that selected the trajectory in the same time sequence as the feature according to the time information. Liu et al. [100] employed the latest DNN method to obtain semantic information as features, Ma et al. [101] selected physical edges with apparent features, such as electric poles. These methods have been fully utilized in current research, and future research should focus more on how to aggregate multiple classes of features together to provide more convenience for subsequent steps.

The next step is feature matching. In order to establish the corresponding relationship between points in Lidar point cloud and pixels in image, the geometric constraint or similarity between descriptors mentioned in Section III calibration method with targets can also be used to match feature points in non-target environment. For example, the brute force matching method proposed by Li-Chee-Ming and Armenakis [102]. Specifically, for each feature point in the image, by calculating its Euclidean distance from all feature points in the point cloud, the closest feature point is found, It is the most similar feature point. However, such a method, on the one hand, may cause mismatching problems.

On the other hand, there may be no point corresponding problems, which is highly likely to cause interference. Therefore, it is extremely necessary to introduce the random sampling consistency random optimization algorithm (RANSAC) mentioned in Section III, It can eliminate mismatched feature point pairs. RANSAC removes extraneous points to get a more accurate mapping of feature points. In addition, the matching strategy based on semantic relation is also a commonly used matching method, aiming at matching features at the level of semantic information as much as possible. For example, Wang et al. [103] proposed that after semantic segmentation in the image, points reflecting vehicles in the point cloud can be matched with pixels with vehicle semantic labels. The method of singular value decomposition can be used for the regression of external parameter. Table 3 summarizes the various targetless

TABLE 3. Various targetless calibration methods.

Methods	Feature type	Feature extraction	Feature matching
Hsu et al. [104]	Edge	Depth discontinuity and Canny detector	Spatial geometrical relation
Zhang et al. [105]	Edge	Depth discontinuity and LSD	Spatial geometrical relation
Alba et al. [106]	Point	SIFT/SUFT and SIFT/SUFT	Descriptors similarity
Zhang et al. [107]	Point	SUFT and SUFT	Descriptors similarity
Hofmann et al. [108]	Semantic	Skyline and Skyline	Semantic relation
Wang et al. [103]	Semantic	3D semantic centroid and 2D semantic centroid	Semantic relation
Peršić et al. [99]	Motion	Object trajectory and Object trajectory	Trajectory relation

calibration methods introduced above, and classifies them according to different feature categories selected.

The feature-based matching method in this section introduces the targetless LiDAR and camera parameter calibration according to different feature types, feature extraction and matching methods. In addition, semantic segmentation can also be carried out only in the image, and only the semantic mask in the image can be constructed. By comparing different literatures, current research data and experimental results obtained, we find that this part of the method is similar to the part of the calibration method with targets, and the features are also selected for relationship matching.

C. MOTION-BASED METHODS

The self-motion-based calibration method mainly utilizes the motion of sensors installed on vehicles to infer the external parameters. Currently, the existing methods mainly involve matching correspondences between sensor trajectories. Specifically, it includes odometry techniques, such as visual odometry proposed by Ishikawa et al. [109] and Park et al. [110], LiDAR odometry, GNSS odometry, and inertial measurement units (IMUs), among others. Based on the usage of self-motion information between sensors and existing literature, self-motion-based methods can be mainly categorized into hand-eye calibration and 3D structure-based calibration.

1) HAND-EYE CALIBRATION METHOD

The hand-eye calibration is widely used in robot vision applications, where the robot's arm represents the "hand," and the camera mounted on the arm represents the "eye." The transformation principle of hand-eye calibration is illustrated in the following Figure 6.

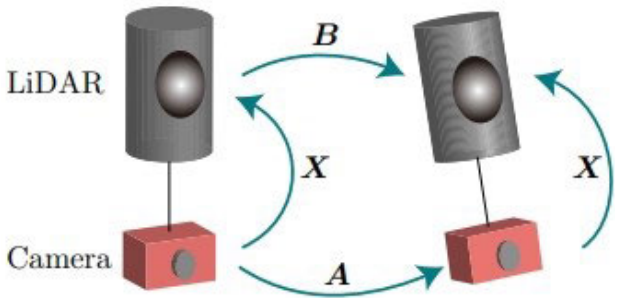

FIGURE 6. The transformation principle of hand-eye calibration [109].

Translating this robotic problem into a calibration problem between LiDAR and the camera, its mathematical expression is as follows:

$$AX = XB \tag{22}$$

Equation (5) directly represents the calculation formula for hand-eye calibration, where and represents the motions between the camera and the robotic arm. After transforming it into a calibration problem between LiDAR and the camera, they respectively represent the motions between the camera and the LiDAR is the matrix to be solved for. Specifically, the process of hand-eye calibration can be roughly divided into three stages: estimating the motion of each sensor, estimating the external parameters, and finally regressing the external parameter matrix.

$$\begin{aligned} T_C^i &= \begin{bmatrix} R_C^i & t_C^i \\ 0 & 1 \end{bmatrix} \\ T_L^i &= \begin{bmatrix} R_L^i & t_L^i \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{23}$$

It is first necessary to obtain the position transformation matrices T_C^i and T_L^i between the camera and the LIDAR neighboring frames and respectively, both can be denoted by the formula (23). For the LiDAR transform matrix, Shi et al. [111] and Taylor and Nieto [112] suggested that it can be obtained as accurately as possible by LiDAR odometry with the ICP algorithm (iterative nearest point), where the iterative nearest point algorithm proposed by Besl and McKay [113] is suitable for motion estimation of a point cloud, which relies on iteratively searching for the nearest points of the previous and previous two sets of point clouds, and then continuously minimizing the distance so that the two sets of point clouds can be linked by transforming the parameters, assume that the original point cloud is pi' and the target point cloud is pi, and set the residual as Equation (24) [113], subtract the transformed point cloud and the target point cloud from the original point cloud. The smaller the residual, the better the registration effect. Also, the odometry can estimate the motion state of the LiDAR [110]. For example, Zhang and Singh [114] proposes that LOAM can select a pair of point clouds and images from each trajectory, and use the pair to estimate the outer parameters, where two algorithms are used, one algorithm runs the odometry at a lower accuracy but uses a higher

frequency, and the other algorithm runs the odometry at an order of magnitude lower frequency but achieves an accurate matching and point cloud alignment.

$$J = \sum ||pi - (R * pi' + T)||^2 \tag{24}$$

Similarly to Lidar odometry, the transformation matrix for successive frames of the camera is obtained using the camera's visual odometry [115]. However, unlike the LIDAR odometry in terms of the evaluation scale, there is a possibility of image distortion and dimensional ambiguity for images that are only purely visually estimated, so some additional scales need to be added to the camera's motion estimation for the calculations, e.g., Inertial Measurement Units (IMUs) or GPS or by the dimensions of the known objects in the scene.

Secondly, the external parameter estimation, because the sensors work independently, can be calculated by Equation (22) to calculate the external parameter, Taylor and Nieto [116] proposed that hand-eye calibration, in general, can be chosen to return to the rotation matrix and translation matrix respectively, and then according to Equation (25) [116] will be obtained by substituting the rotation and translation matrices, as long as the matrix of R_T is obtained, then by the transformation matrix $X = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$ in Equation (22) can be returned to the external parameter.

$$R_C^i t_T - I t_T = R_T t_L^i - t_C^i \tag{25}$$

Many different expressions can return the outer parameters transformation matrix, for example, the rotation matrix (3*3) can be uniquely defined for 3D rotations. Park et al. [110] recently proposed to output the rotation matrix by finding correspondences from the time trajectory. However, it is independent of the order of the rotations, there are singularities and ambiguities in the parameterization of the Euler angles of the rotation matrix. For example, problems such as the gimbal lock problem and gimbal ambiguity can lead to complexity in the parameterization and resolution of the rotation matrix. In order to alleviate such problems of computational complexity, Taylor and Nieto [117] proposed the use of axis-angle representation of the rotation relation; specifically, it parameterizes the rotation by two parameters: a unit vector (i.e., the axis of rotation) pointing in the direction of the rotation, and an angle indicating the magnitude of the rotation along that axis. For example, if there is a vector v in 3D space, u is the unit vector in the same direction as the rotation axis, and θ is the Angle at which v passes around the right hand direction of u, then the rotated vector v' is represented by (26) [117]. The axis-angle representation simplifies the output process. Recently, it has also been suggested by researchers such as Xu et al. [118] that the rotation parameters can also be expressed in a Lie algebra form, which is applicable to optimization problems; specifically, it specifies the external parameters through a vector with 6 degrees of freedom (DoF) variables.

$$v' = (cos(\theta)v + (1 - cos(\theta))(u * v)u + sin(\theta)(u * v)) \tag{26}$$

TABLE 4. LiDAR-camera calibration methods based on hand-eye calibration.

Methods	Motion Estimation (LiDAR-Image)	Rotation Parameter Representation	Refinement Methods
Taylor and Nieto et al. [112]	ICP and SFM	Angle-axis	Edge Alignment
Taylor and Nieto et al. [117]	ICP and Visual odometry	Angle-axis	Color matching
Shi et al. [111]	LiDAR odometry and Visual odometry	Angle-axis	Intensity matching
Liao and Li [119]	ICP and Visual odometry	Quaternion	Edge Alignment
Xu et al. [118]	ICP and Visual odometry	Lie algebra	Depth matching and Edge alignment
Park et al. [110]	LiDAR odometry and Visual odometry	Rotation matrix	3D-2D point matching

However, by far the most used representation is the quaternion representation proposed by Liao and Liu [119], This unique and simple representation describes finite rotations in 3D space, divided into real and imaginary parts. w, x, y, and z are the four parts of the quaternion, which denote the three components of the real and imaginary parts, respectively. The following table summarizes the current hand-eye-based LiDAR-camera calibration methods according to different motion estimation strategies.

This section discusses targetless calibration methods based on hand-eye calibration. Different calibration methods use different motion estimation and optimization methods, and by comparing the various methods, we find that ICP and odometer methods are still used more often, and feature matching is mainly used in refining the parameters. The current hand-eye calibration methods have been evolving for several years, and it is expected that more accurate motion estimation and representation methods will emerge in the future.

2) 3D STRUCTURE-BASED CALIBRATION

Unlike the hand-eye calibration-based methods, the 3D structure-based methods do not rely on odometry but instead estimate the surrounding environment's 3D structure through image analysis. This is another type of motion-based LiDAR-camera calibration method, with structure from Motion (SFM) being one of the most commonly used techniques [120]. SFM is a technology that estimates the 3D structure of a scene from a sequence of 2D images. It has numerous applications in various fields, such as 3D modeling, augmented reality, and visual SLAM. Specifically, the 3D structure estimation based on SFM involves mounting the camera on a moving vehicle and capturing a series of images as the vehicle moves, resulting in a set of 3D point clouds.

This allows the LiDAR-camera calibration problem to be transformed into a registration task in the 3D domain. The next step involves converting the 2D image sequence into 3D point clouds and using the Iterative Closest Point (ICP) algorithm to align the SFM point cloud with the LiDAR point cloud, obtaining an initial estimation of the external parameters. In theory, 3D points will be perfectly projected to pixels, but due to noise interference, error thinning is required, as shown in the following formula (27) [120], where W_{ij} is 1 when camera i observes track j, and 0 when camera i observes track j otherwise, and $||q_{ij} - P(C_i, X_j)||$ represents the cumulative sum of projection errors of track j in camera i. For example Swart et al. [121] utilized this SFM approach to find the point of interest exact results by point cloud alignment followed by SIFT. In addition to this Moussa et al. [122] used beam block accurate external parameters after using 3D-3D alignment, so that the matching results are better after fine calibration with continuous refinement of parameters.

$$g(C, X) = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} ||q_{ij} - P(C_i, X_j)||^2 \tag{27}$$

In recent years, with technological advancements, many researchers have optimized subsequent operations after SFM registration. For example, Wang et al. [123] utilized continuous scene information from vehicle motion and converted it into 3D information to obtain initial external parameters. This method employs the SFM algorithm to compute 3D points from a sequence of 2D images, followed by ICP-based point cloud registration to estimate preliminary results. However, they have made improvements based on the continuation of the SFM method of alignment, specifically the projection of 3D LiDAR points into the 2D image plane through preliminary external parameters, the use of edge feature points and the strategy of combining other optimization methods, such as the use of a combination of points, edges, and semantic information repeated to improve the accuracy of the external parameters.

Although the SFM method can reconstruct 3D environments and obtain environmental information, it currently faces challenges. For instance, converting images into 3D point clouds may lead to sparsity in the point cloud, resulting in decreased matching rates and the error will increase if ICP algorithm is used again. Some researchers proposed upsampling when converting images into point clouds, which can be an effective solution. Additionally, to address this issue, Li et al. [124] proposed an automatic matching method for semantic features in point clouds and images, They refine the parameters by maximizing the overlap area between the two, so that even if the number of point clouds is small, the parameters can be iterated by the overlap area. Furthermore, Nagy et al. [125] also utilized semantic information during the point cloud registration stage.

After the overview in this section, we find that the method based on 3D structure estimation does not need precise prior information, and it can recover the 3D structure of the scene from a set of images without prior knowledge of the internal and external parameters of the camera or the specific information of the scene. However, there is also a problem of high computational complexity in large-scale scenarios. With the development of current research, although effective reconstruction methods such as SFM are proposed, they also need various measures to increase the number of point clouds. Therefore, in future work, We believe that the focus should be on how to obtain as much detailed location cloud information as possible and how to refine it to the best effect once the initial external parameters are obtained.

V. DEEP LEARNING-BASED ONLINE CALIBRATION TECHNIQUES

In Section III.IV, we provide an overview of offline calibration methods based on targeted and untargeted calibrations, the two traditional calibration methods usually need to be carried out in a laboratory environment, and the process cannot be separated from the manual calibration, which can only be returned to the field for recalibration in case of a change in the relative position between the camera and the Lidar, whereas the on-line calibration technique allows calibrations to be carried out in real-time scenarios to adapt to real-world variations.

In current research, many researchers, such as Xu et al. [118], employ neural network models to estimate camera or sensor parameters. These models can be trained with training data to learn the relationship between camera parameters and input images. The training data may be a dataset containing known parameters and corresponding images. Once the model is trained, it can be applied to images in a live scene and camera parameters can be obtained by inference. The end-to-end approach simplifies the online calibration process, and several research efforts are already working towards fully automated LiDAR-Camera calibration without any a priori information. For instance, Li and Lee [126] transformed the alignment problem into a classification task by first determining whether each point in the point cloud is in or out of range of the camera's image, and then passing these labeled points into a novel inverse camera projection solver to estimate the relative pose, but this method still requires an initial pose guess that can be projected.

Moreover, Feng et al. [127] proposed the 2D3D-MatchNet for matching low-level cross-modal features (such as SIFT and ISS). It should be noted that because the Field Of View and data modes of LiDAR and the camera are very different, the morphology of low-level features is also very different, so it is difficult to match the low-level features of the camera and LiDAR, and the features of cross-modal invariance tend to exist in the high-level features (such as structural or semantic information).

In order to better evaluate the calibration effect, online calibration techniques usually need data sets for training and verification. Datasets are collection that contain images of the calibrated object and corresponding labels or comments.

Online calibration is to identify and locate the object in the image by deep learning model, and then determine its position and attitude in the image. In order to evaluate the effectiveness of the external parameter calibration method, datasets containing multiple calibration objects are needed. After the external parameter is obtained, the error between the data set and the ground truth value can be observed experimentally to evaluate the calibration effect, and the reliability of the algorithm in different scenarios can be verified. Multimodal sensor data sets provide rich real-world scene data. Over the years, various datasets have been developed for different purposes. For example, the earliest dataset, ''Rawseeds'' [128], was used for detection, recognition [129], and vehicle positioning using a combination of GPS and IMU [130]. More recently, one of the most well-known datasets is the KITTI dataset proposed by Geiger et al. [131]. It is widely used for algorithm validation, providing 1392×512 pixel RGB images and depth map data from Velodyne HDL-64E lidar, including methods for calibrating LiDAR and cameras, such as the work by Zhang and Rajan [132].

Deep learning-based online calibration methods can be categorized into two approaches based on network regression: direct regression and error regression. In the direct regression approach, the network takes input data from point clouds, such as RGB images and depth maps. It aims to learn the direct mapping relationship between the input data and calibration parameters. The network learns to predict the calibration parameters based on the input data directly. The network is trained to regress and predict the errors between the estimated calibration parameters and the ground truth in the error regression approach. By optimizing the initial calibration parameters using these errors iteratively, the network aims to reduce the calibration errors and improve the accuracy of the calibration. In the following, we will provide a detailed introduction to the process of deep learning-based online calibration techniques.

A. REGRESSION NETWORK FOR EXTERNAL CALIBRATION PARAMETERS

The RegNet [140] is a representative example of a regression network for directly calibrating external parameters. It replaces the traditional three-step process and directly regresses the six degrees of freedom for calibration. The external parameters of the datasets are calculated using the method of formula (28) [140] as the ground truth value, which means that the given point x in the sensor coordinate system is converted to point y in the world coordinate system, and then the ground truth value H is obtained. The network architecture consists of a feature extraction part and a registration network part. RGB images and point cloud depth maps are used as input data. First, the NiN feature extraction network [141] is applied to extract features from the image and point cloud data with different channel numbers. Then, feature concat is performed on both data, followed by two fully connected layers to obtain the loss function.

$$y = Hx \tag{28}$$

Secondly, cycle the steps of ''3D-2D projection ->RegNet network processing -> Update external parameters'' to continuously obtain new external parameters and optimize the loss function. Finally, five kinds of networks are trained from large to small, representing five different deviation ranges. Comparing the calibration effects of five kinds of networks, it is found that the regression external parameters are constantly close to the ground truth value. In the course of training here, two challenging sequences were selected on the KITTI dataset for a total of 574 frames, while all other sequences were only used for training (14,863 frames). We randomly vary the error of the external parameter $\phi_{decalib}$, each frame during training generates a potentially infinite amount of training data on the dataset, so that the dataset can be used for iterative training. RegNet can conduct target-free calibration from scratch without manual intervention, and the calibration accuracy is higher than that of traditional methods. However, the network's performance is limited by the design and capability of the network structure. The feature extraction and matching network used is too simple, and the geometry of the point cloud is not considered. When the internal parameters of the camera change, it is necessary to fine-tune the trained model, which may not be able to capture the characteristics and relationships between complex sensors, and in this case, the accuracy is not high. Figure 7 is the network structure diagram of RegNet.

Similarly, building upon RegNet, Liu et al. proposed an online calibration method that integrates visual and depth sensors. They fused image and point cloud data into a single depth sensor and then calibrated it with the image.

CalibRCNN [143] calculates the external parameters between camera and Lidar by solving the relative position relationship between successive frames. The network takes as input images, incorrectly calibrated point cloud projection depth maps, and camera calibration matrices (camera internal parameters). After pre-processing KITTI-odometry's 00-06 sequence datasets, 90% of the frames in each sequence are taken as the training set and the remaining 10% as the test set. In addition, part of the Kitti-raw 0926 sequence is used, which has unfamiliar scenes and different internal parameters. First, feature extraction is carried out. For RGB image branches, the convolution layer of the pre-trained ResNet-18 network is used. For the branches of the depth map, a network similar to the structure of ResNet-18 is used. The convolution layer and LayerNorm layer can extract the deep features, aggregate the extracted features and input them into the LSTM layer, extract the time information between successive frames for sequential learning, analyze the translation and rotation information from the depth features output by LSTM, and predict the translation vector τ and rotation vector γ. CalibRCNN obtains the calibration parameters of the depth map by calculating the formula, and then replaces the formula to predict external parameters, to obtain the position matrix

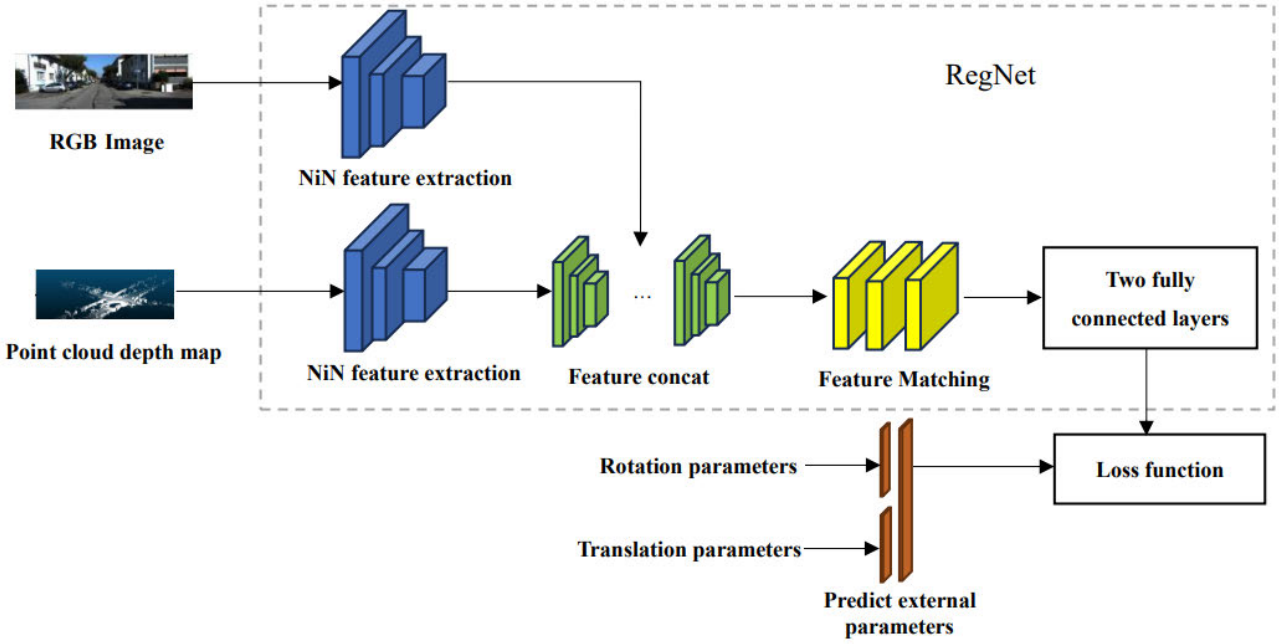

FIGURE 7. Network architecture diagram of RegNet.

transformation between two frames of the camera. Calibr-cnn introduces three loss functions, including luminosity, geometry, and regression, and continuously optimizes the camera error loss function to accurate external parameters. The following formula (29) [143] is the solution formula for the real external parameter $T_{\phi gt}$, where T_{velo} is the coordinate transformation parameter between point cloud frames and the pose transformation between T_{cam} cameras. The network combines CNN and LSTM through end-to-end training and optimization, CalibRCNN can effectively capture the visual features of targets, improve target detection accuracy, and adapt to different scenarios. However, CalibRCNN needs to constantly improve its generalization ability in the future. The network structure of CalibRCNN can be visualized as shown in Figure 8.

$$T_{velo} = T_{\phi gt} T_{cam} T_{\phi gt}^{-1} \tag{29}$$

All kinds of networks described above have been widely used in the field of online calibration, but they all need initial pose estimation to optimize iteratively in the calibration process. Since 2023, Sun et al. [144] have proposed an adaptive lidar camera calibration method, ATOP, that first converts point cloud data into depth map information, then uses a cross-modal object matching network (CMON) consisting of two parallel embedded branches for feature extraction. The setting of the attention mechanism mainly focuses on the overlapping area of the field of view between the liDAR and the camera, thus generating the corresponding relationship of the 3D-2D image. Through the corresponding relationship, two cascaded particle swarm optimization (PSO) based optimization algorithms, namely Point-PSO and Pose PSO, are used for the attitude initialization and refinement of the optimization stage. To train CMON, a cross-modal target matching dataset was created, which was collected from the platform acquisition data and part of the KITTI mileage datasets (sequence 0 to 8), with a total of 2892 labeled data, of which 1513 were used for training CMON, 379 for validation, and 1000 for testing. The tag data for sequence 03 in KITTI is all allocated to the test set. Figure 9 shows the overall framework of ATOP method. Compared with other methods, the most significant advantage of this method is that it does not need to estimate the initial pose and does not rely on specific calibration targets, which has great reference value for future research on perceptual systems. However, as far as the current research status is concerned, there are still time-consuming and laborious, and highly subjective problems in the training process of collecting annotated data.

B. REGRESSION NETWORK FOR ERRORS

CalibNet [145] is a network that regresses external calibration parameters instead of calibration parameters, primarily using raw data from the KITTI dataset, specifically RGB images and velodyne point cloud data to prepare the dataset, 2,609 driving sequences were used for training, as they consist of a large number of sequences with good scene variation while capturing a wide range of input biases. The training data was amplified by random sampling in the range of ±10o rotation and ±0.2 m translation of any axis, through a neural network that takes as input the image, the depth map of the point cloud projection that is not correctly calibrated, and the camera calibration matrix (Camera Intrinsic Parameters). Feature extraction is performed first; for the image branch, the convolutional layer of a pre-trained ResNet-18 network is used [146], and for the depth map branch, we use a similar

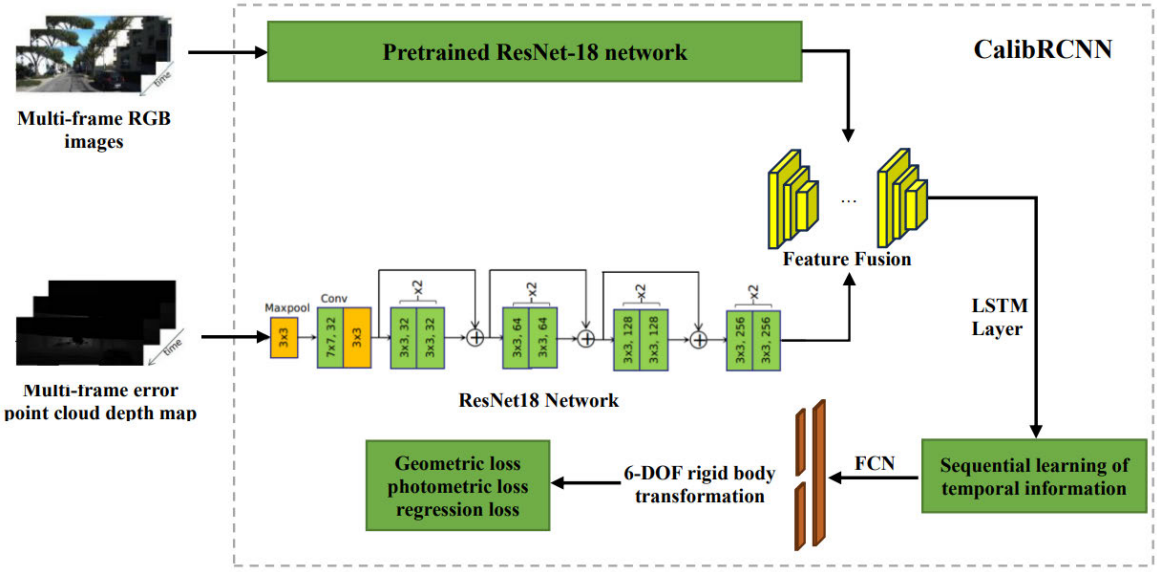

FIGURE 8. Network architecture diagram of CalibRCNN.

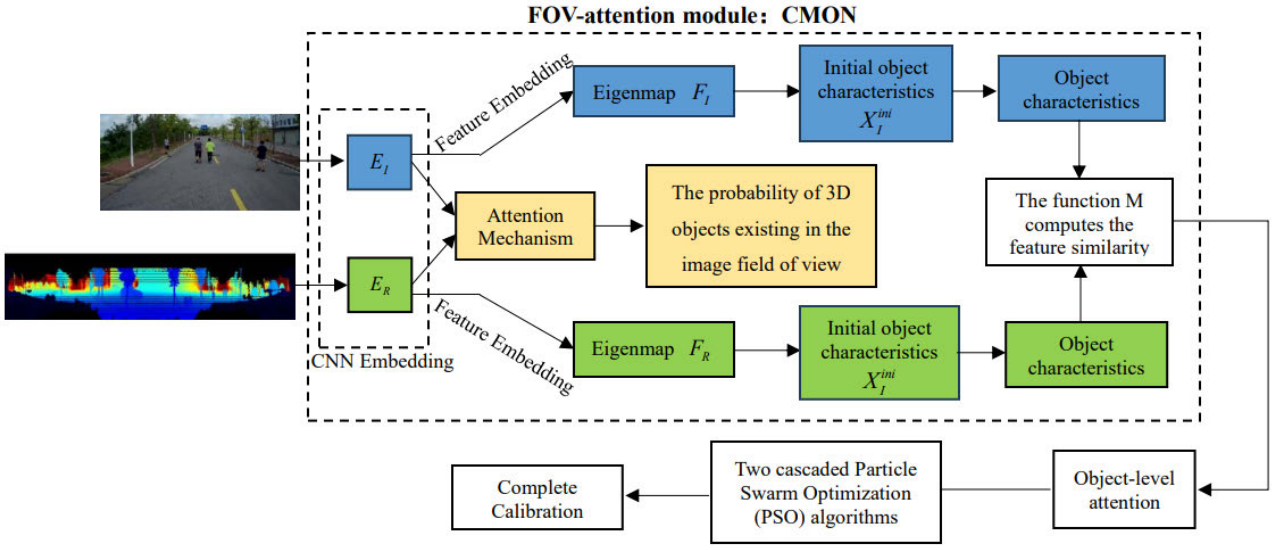

FIGURE 9. Network architecture diagram of ATOP.

architecture to the RGB branch but halve the number of filters at each stage as the features of the depth map, It need to be learned from scratch. The outputs of the two branches are feature spliced along the channel dimensions, followed by global feature aggregation through a series of additional total convolutional layers, and the final outputs are obtained as the corresponding rotation and translation vectors, which are used to solve the external calibration parameters between the camera and the Lidar by minimizing the introduced distance loss from the point cloud, as well as the photometric loss function, to obtain the best possible range of error. The initial calibration parameters are continuously calibrated to maximize the image's and the point cloud's geometric and photometric consistency. Although CalibNet can solve the point cloud structure problem, unique training methods must be designed to achieve calibration accuracy during the training process. Figure 10 shows the network structure of CalibNet.

LCCNet [147] improves the network structure based on CalibNet. Instead of directly regressing the Lidar and camera external parameters, it regresses the uncalibrated deviation between the projection and the initial calibration to the ground truth. The results of the algorithm were observed primarily using the odometer branch of the KITTI dataset, which provides calibration parameters between sensors, trained and validated using sequences 01 to 20 (39,011 frames) and tested using sequence 00 (4541 frames). The test data set is spatially independent from the training data set, except for a very small subset sequence (about 200 frames), so it can be assumed that the test scenario is not in the

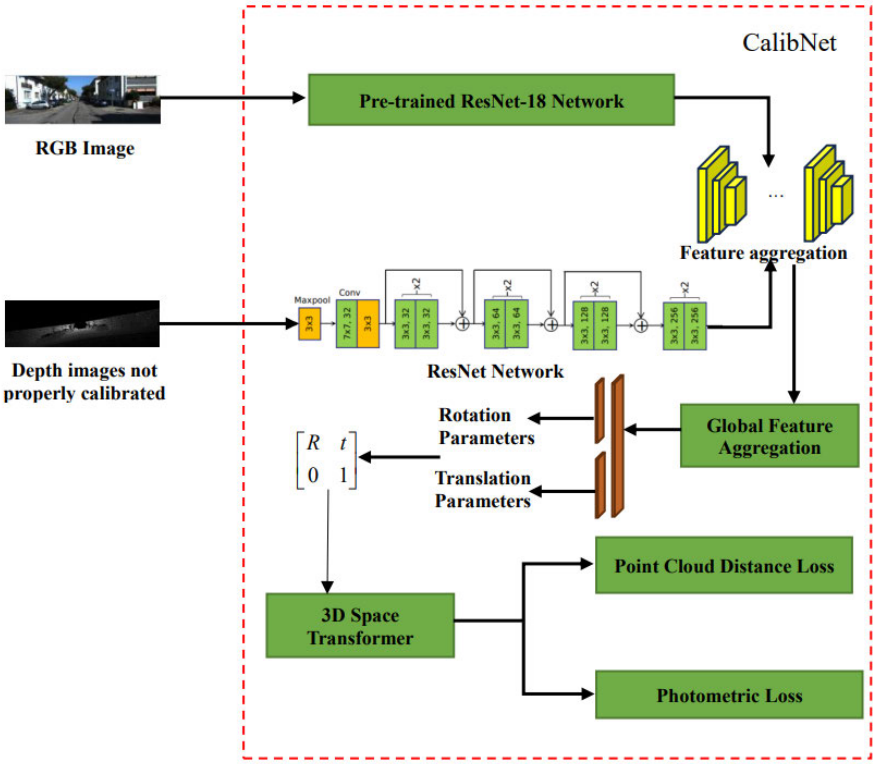

FIGURE 10. Network architecture diagram of CalibNet.

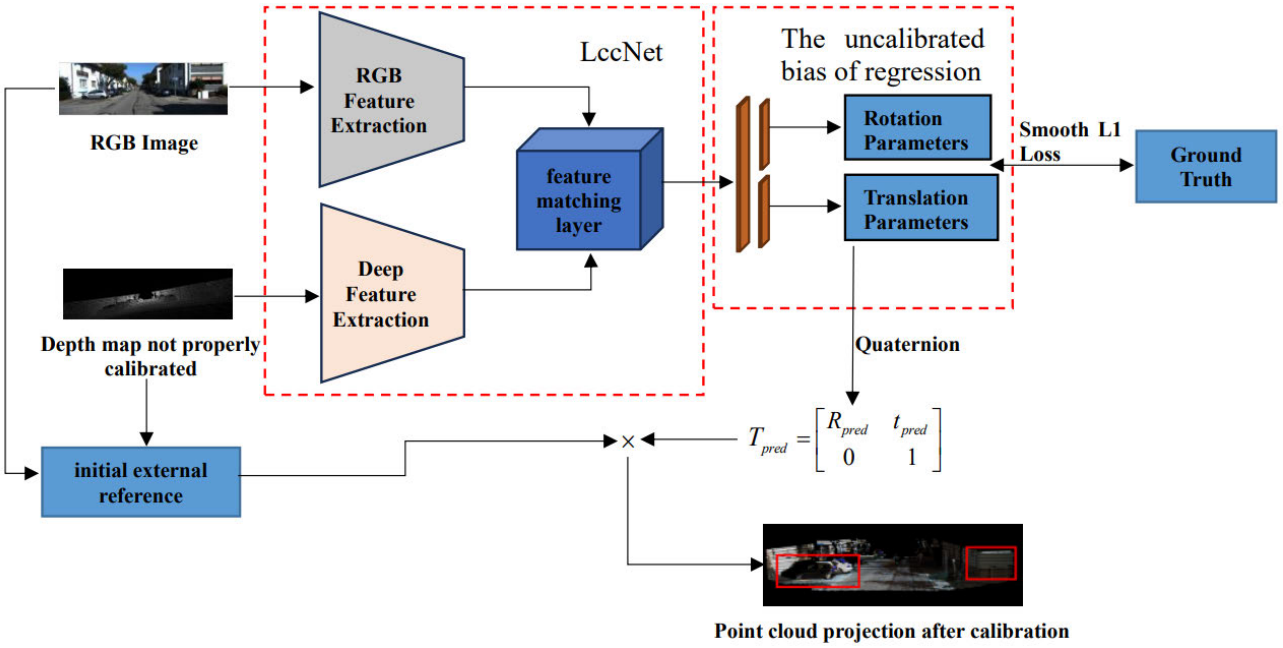

FIGURE 11. Network architecture diagram of LCCNet.

training data. The network takes the RGB image and the depth map formed by the projection of the incorrectly calibrated point cloud as the input to the network, and extracts the features of the image and the point cloud depth map through a feature extraction network respectively, and then introduces a feature matching layer in the optical flow detection network FlowNet [148], which correlates and matches the image features and the point cloud depth map features by calculating the similarity between the feature map vectors, and finally utilizes the feature global aggregation network to regress the external parameter error calibration parameters between the LIDAR and the camera, and continuously calibrate the

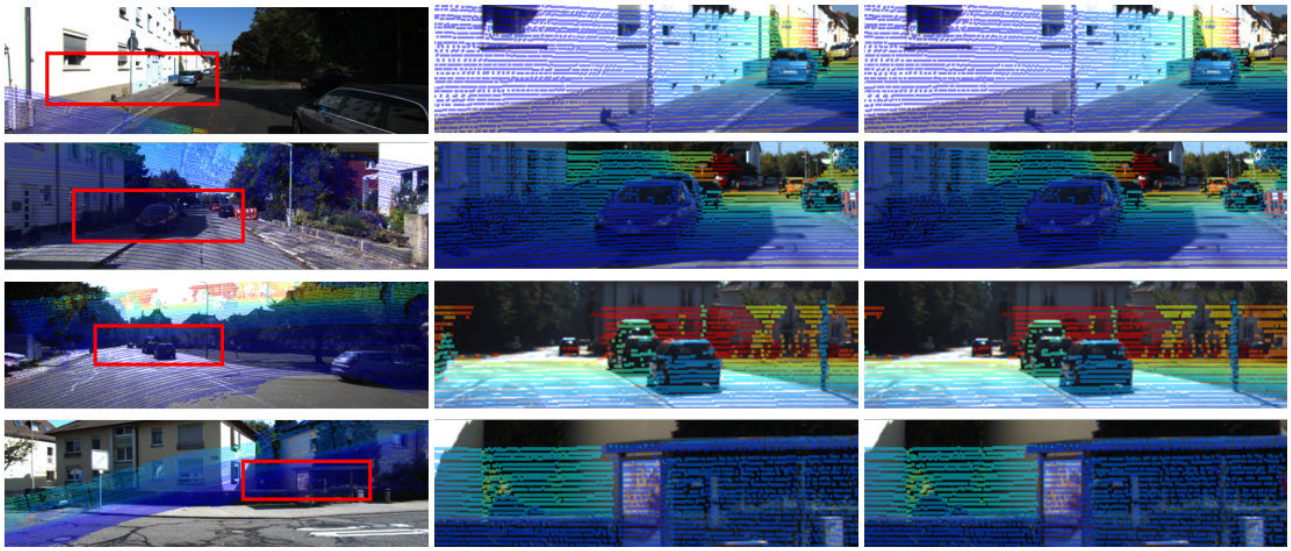

FIGURE 12. Visual calibration results for LCCNet.

TABLE 5. Comparison of online calibration network structures based on deep learning.

	RegNet	CalibNet	LccNet	CalibRCNN	ATOP
Input	RGB image and error calibration depth map	RGB image and error calibration depth map	RGB image and error calibration depth map	RGB image and error calibration depth map	RGB image and depth map
Output	External reference six degrees of freedom	Errors from ground truth	Errors from ground truth	External reference six degrees of freedom	External reference six degrees of freedom
Feature Extraction (RGB Images)	NiN module extraction	Pre-trained ResNet-18 network	Pre-trained ResNet-18 network	Pre-trained ResNet-18 network	Parallel embedded branches
Feature Extraction (Depth Map)	NiN module extraction	Similar Pre-trained ResNet-18 network	Pre-trained ResNet-18 network	Similar Pre-trained ResNet-18 network	Parallel embedded branches
Test Objects	Continuous calibration at the same frame	Continuous calibration at the same frame	Continuous calibration at the same frame	Continuous Frame Calibration	Continuous calibration at the same frame
Dataset	KITTI	KITTI	KITTI	KITTI	KITTI
Type of loss function	Geometric loss function	Point cloud distance loss and photometric loss	Point cloud distance loss and smoothed L1 loss	Photometric loss, geometric loss and regression loss	Point-PSO, Pose-PSO to refine the parameters

TABLE 6. Calibration results for different networks.

Method	Mis-calibrated Range	Translation absolute Error (cm)				Rotation absolute Error (°)			
		mean	X	Y	Z	mean	Roll	Pitch	Yaw
RegNet	[-1.5m, 1.5m]/[-20° , 20°]	6	7	7	4	0.28	0.24	0.25	0.36
CalibNet	[-0.2m, 0.2m]/[-10° , 10°]	4.34	4.2	1.6	7.22	0.41	0.18	0.9	0.15
LccNet	[-1.5m, 1.5m]/[-20° , 20°]	0.297	0.262	0.271	0.357	0.017	0.02	0.012	0.019
CalibRCNN	[-0.25m, 0.25m]/[-10° , 10°]	0.093	0.062	0.043	0.054	0.805	0.199	0.64	0.446
ATOP	/	2.56	1.21	2.83	3.64	0.029	0.06	0.005	0.02

initial external parameter through the calibration parameters, to improve the external parameter calibration accuracy. In the calibration process, unlike other networks, the loss function introduces the point cloud distance loss as an additional self-supervised signal in addition to the smoothed L1 loss. LCCNet adopts an end-to-end training method, which does

not require too much manual intervention, and it is robust enough to adapt to the real-time environment and also meets the requirement of higher accuracy so that there is no need for fine-tuning the model when the camera's internal parameter is changed. Figure 11 shows the network structure of LCCNet.

Various studies have evaluated the effectiveness of different calibration networks. Table 5 compares the characteristics of some typical networks for online calibration based on deep learning. From Table 5, the typical networks' input part is basically the same. At the same time, the output can be error regression with external parameter regression, and also summarizes the advantages and disadvantages of the various networks and the different approaches used in the feature extraction and matching process.

This section introduces deep learning-based online calibration techniques through networks that regress external calibration parameters and networks that regress errors, respectively. As the hottest method in the current development of multi-sensor calibration direction, it mainly uses, for example, pre-trained Resnet-18 as well as Resnet-18 networks to perform feature extraction on images and point clouds, respectively, and the initial external parameters through loss functions or optimization algorithms for continuously optimized by loss functions or optimization algorithms. At the same time, in the process of using the data set, it is found that in addition to the KITTI data set commonly used in online calibration networks, other existing data sets are also applicable, such as BDD [133], ApolloScape [134], ONCE [136], MVSEC [137], Argoverse 2 [138], Rope3D [139], etc, and others are suitable for outdoor sensor calibration, and it is expected that various algorithms can also use other data sets for verification and effect comparison in the future.

We compare the calibration effects of different methods in Table 6. Through the calibration data of various types of networks on the same dataset of KITTI, we find that LCCNet is the neural network with the best calibration effect at present, and the ATOP network, which was just proposed this year, has added the attention mechanism, and its calibration effect is also noteworthy. In future development, the attention mechanism and the optimization algorithm is a module that can be borrowed. Compared with the iterative complexity of other networks, the network made so far is better at achieving lightweight, and its fastest response time can be up to an average of 0.073 s. We look forward to the emergence of better calibration networks in the future.

VI. CONCLUSION AND OUTLOOK

Multi-sensor exo-reference calibration techniques have become a hot topic in computer vision due to their competitive performance and great potential in deep learning. Many methods have been proposed in recent years to discover and summarize various external parameter calibration methods. These methods perform excellently on various vision tasks such as fault detection, remote sensing, robotics, localization and 3D map reconstruction, and unmanned driving. However, the potential of external reference calibration has yet to be fully explored, and there are still some challenges to be addressed. This section summarizes the approaches discussed throughout the paper and presents current challenges, providing insights into future perspectives.

A. CONCLUSION

This review introduces various methods for external parameter calibration of modal sensors such as cameras and LIDARs, mainly including targeted calibration techniques, targetless calibration techniques, and online calibration techniques based on deep learning, and contrary to other reviews, we propose a typical process structure for external parameter calibration for each method. Improve accuracy, such as the simpler indoor environments, are still applicable. However, when the environment is slightly more complex, there are some defects of erroneous feature extraction and the limitation of the initial value, mainly due to the susceptibility of the calibration plate to changes in illumination. In order to alleviate this problem and evolve the technique of owning targets, we introduce the fourth section of the targetless calibration technique, specifically through the three categories of methods based on information, features, and motion. By comparing the methods, we find that the general process is similar to the third section. However, it eliminates the trouble of arranging the environment, and the current calibration using odometers and inertial measurement units is more effective. In addition, we also review the latest methods for online calibration and introduce popular point cloud datum datasets and the performance of these methods on 3D visual calibration tasks. Currently, LCCNet is the best performer in both response time and error of six values from the ground truth. However, methods incorporating an attention mechanism can be considered in the future to help improve performance. This review is limited to these three methods, which cover most of the primary methods for external parameter calibration, and we believe that the deep learning-based method is the essential method to be emphasized at present. Its process is relatively free of human intervention.

B. CHALLENGES AND FUTURE PROSPECTS

Although the methods for external parameter calibration have shown good performance on several tasks (including classification, part, and semantic segmentation), some areas still need more attention. Extensions for more significant scenes are rarely exploited, as most current works rely on slicing large scenes into smaller parts. At the time of this review, most of the works are for one-frame scenes, and only CalibRCNN employs continuous frame calibration, so future works exploring could focus more on deep learning of large-scale 3D scenes.

In order to advance the development of camera and LiDAR external parameter calibration techniques, we propose potential directions for future research. One direction is effectiveness and efficiency. The goal is to develop efficient

deep learning networks, i.e., networks with high performance and low resource costs. Performance determines whether the model can be applied to real-world applications, while resource cost affects device deployment. Effectiveness is usually associated with efficiency, so finding a better balance between them is an exciting research topic.

With large-scale data training, deep learning networks can achieve advanced performance on benchmark dataset tests. Whether it is possible that neural networks need big data rather than a uniform one-frame feature generalization bias also echoes the large-scale scenario problem we posed in the first paragraph of this subsection. Finally, a question is left for reflection: can deep learning-based network models achieve satisfactory results under lightened manipulation and with large-scale data training?

REFERENCES

[1] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: A survey," *IEEE Access*, vol. 8, pp. 2847–2868, 2020.

[2] L. Song and R. Yan, "Bearing fault diagnosis based on cluster-contraction stage-wise Orthogonal-Matching-Pursuit," *Measurement*, vol. 140, pp. 240–253, Jul. 2019.

[3] V. Ankarao, V. Sowmya, and K. P. Soman, "Multi-sensor data fusion using NIHS transform and decomposition algorithms," *Multimedia Tools Appl.*, vol. 77, no. 23, pp. 30381–30402, Dec. 2018.

[4] L. Luo, X. Wang, H. Guo, R. Lasaponara, X. Zong, N. Masini, G. Wang, P. Shi, H. Khatteli, F. Chen, S. Tariq, J. Shao, N. Bachagha, R. Yang, and Y. Yao, "Airborne and spaceborne remote sensing for archaeological and cultural heritage applications: A review of the century (1907–2017)," *Remote Sens. Environ.*, vol. 232, Oct. 2019, Art. no. 111280.

[5] S. Siachalou, G. Mallinis, and M. Tsakiri-Strati, "A hidden Markov models approach for crop classification: Linking crop phenology to time series of multi-sensor remote sensing data," *Remote Sens.*, vol. 7, no. 4, pp. 3633–3650, Mar. 2015.

[6] X. Mao, W. Li, C. Lei, J. Jin, F. Duan, and S. Chen, "A brain-robot interaction system by fusing human and machine intelligence," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 3, pp. 533–542, Mar. 2019.

[7] S. Saponara, "Sensing and connection systems for assisted and autonomous driving and unmanned vehicles," *Sensors*, vol. 18, no. 7, p. 1999, Jun. 2018.

[8] X. Jia, Z. Hu, and H. Guan, "A new multi-sensor platform for adaptive driving assistance system (ADAS)," in *Proc. 9th World Congr. Intell. Control Autom.*, Jun. 2011, pp. 1224–1230.

[9] J. Kümmerle and T. Kühner, "Unified intrinsic and extrinsic camera and LiDAR calibration under uncertainties," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 6028–6034.

[10] Y. Lu, W. Zhong, and Y. Li, "Calibration of multi-sensor fusion for autonomous vehicle system," *Int. J. Vehicle Des.*, vol. 91, nos. 1–3, pp. 248–262, May 2023.

[11] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11618–11628.

[12] H. A. Ignatious and M. Khan, "An overview of sensors in autonomous vehicles," *Proc. Comput. Sci.*, vol. 198, pp. 736–741, Dec. 2021.

[13] D. Li, G. Wen, and S. Qiu, "Cross-ratio-based line scan camera calibration using a planar pattern," *Opt. Eng.*, vol. 55, no. 1, Jan. 2016, Art. no. 014104.

[14] S. Donné, H. Luong, S. Dhondt, N. Wuyts, D. Inzé, B. Goossens, and W. Philips, "Robust plane-based calibration for linear cameras," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 36–40.

[15] R. Roriz, J. Cabral, and T. Gomes, "Automotive LiDAR technology: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6282–6297, Jul. 2022.

[16] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, Mar. 2021.

[17] Z. Qiu, J. Martínez-Sánchez, P. Arias-Sánchez, and R. Rashdi, "External multi-modal imaging sensor calibration for sensor fusion: A review," *Inf. Fusion*, vol. 97, Sep. 2023, Art. no. 101806.

[18] J.-O. Nilsson and P. Händel, "Time synchronization and temporal ordering of asynchronous sensor measurements of a multi-sensor navigation system," in *Proc. IEEE/ION Position, Location Navigat. Symp.*, May 2010, pp. 897–902.

[19] W. Lixin, S. Wei, and L. Chao, "Implementation of high speed real time data acquisition and transfer system," in *Proc. 4th IEEE Conf. Ind. Electron. Appl.*, May 2009, pp. 382–386.

[20] K. Römer, P. Blum, and L. Meier, "Time synchronization and calibration in wireless sensor networks," in *Handbook of Sensor Networks: Algorithms and Architectures*. Hoboken, NJ, USA: Wiley, 2005, pp. 199–237.

[21] E. Olson, "A passive solution to the sensor synchronization problem," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 1059–1064.

[22] Y.-J. Zhang, "Camera calibration," in *3-D Computer Vision: Principles, Algorithms and Applications*. Cham, Switzerland: Springer, 2023, pp. 76–77.

[23] S. Hong, H. Ko, and J. Kim, "VICP: Velocity updating iterative closest point algorithm," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 1893–1898.

[24] T. Liu, A. Moore, K. Yang, and A. Gray, "An investigation of practical approximate nearest neighbor algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 17, 2004, pp. 1–8.

[25] J. Schultz and K. Lynch, "Robotics, vision and control: Fundamental algorithms in MATLAB," *IEEE Control Syst. Mag.*, vol. 40, no. 1, pp. 52–54, Feb. 2020.

[26] P. An, T. Ma, K. Yu, B. Fang, J. Zhang, W. Fu, and J. Ma, "Geometric calibration for LiDAR-camera system fusing 3D-2D and 3D-3D point correspondences," *Opt. Exp.*, vol. 28, no. 2, pp. 2122–2141, 2020.

[27] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[28] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi, "Autoware on board: Enabling autonomous vehicles with embedded systems," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2018, pp. 287–296.

[29] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, "Baidu Apollo EM motion planner," 2018, *arXiv:1807.08048*.

[30] A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna, "LiDAR-camera calibration using 3D-3D point correspondences," 2017, *arXiv:1705.09785*.

[31] L. Zhou and Z. Deng, "Extrinsic calibration of a camera and a LiDAR based on decoupling the rotation from the translation," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 642–648.

[32] Y. Lyu, L. Bai, M. Elhousni, and X. Huang, "An interactive LiDAR to camera calibration," in *Proc. IEEE High Perform. Extreme Comput. Conf. (HPEC)*, Sep. 2019, pp. 1–6.

[33] R. Yang, S. Cheng, and Y. Chen, "Flexible and accurate implementation of a binocular structured light system," *Opt. Lasers Eng.*, vol. 46, no. 5, pp. 373–379, May 2008.

[34] M. Rufli, D. Scaramuzza, and R. Siegwart, "Automatic detection of checkerboards on blurred and distorted images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 3121–3126.

[35] S.-H. Lee, S.-K. Lee, and J.-S. Choi, "Correction of radial distortion using a planar checkerboard pattern and its image," *IEEE Trans. Consum. Electron.*, vol. 55, no. 1, pp. 27–33, Feb. 2009.

[36] X. Zhao, W. Chen, Z. Liu, X. Ma, L. Kong, X. Wu, H. Yue, and X. Yan, "LiDAR-ToF-binocular depth fusion using gradient priors," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Aug. 2020, pp. 2024–2029.

[37] J. Domhof, J. F. P. Kooij, and D. M. Gavrila, "An extrinsic calibration tool for radar, camera and LiDAR," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8107–8113.

[38] J. Peršić, I. Marković, and I. Petrović, "Extrinsic 6DoF calibration of a radar-LiDAR-camera system enhanced by radar cross section estimates evaluation," *Robot. Auto. Syst.*, vol. 114, pp. 217–230, Apr. 2019.

[39] G. Pandey, J. Mcbride, S. Savarese, and R. Eustice, "Extrinsic calibration of a 3D laser scanner and an omnidirectional camera," *IFAC Proc. Volumes*, vol. 43, no. 16, pp. 336–341, 2010.

[40] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3D LiDAR using line and plane correspondences," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 5562–5569.

[41] J. Beltrán, C. Guindel, A. de la Escalera, and F. García, "Automatic extrinsic calibration method for LiDAR and camera sensor setups," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17677–17689, Oct. 2022.

[42] G. Yan, F. He, C. Shi, P. Wei, X. Cai, and Y. Li, "Joint camera intrinsic and LiDAR-camera extrinsic calibration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 11446–11452.

[43] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 3936–3943.

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

[45] A. P. Povendhan, L. Yi, A. A. Hayat, A. V. Le, K. L. J. Kai, B. Ramalingam, and M. R. Elara, "Multi-sensor fusion incorporating adaptive transformation for reconfigurable pavement sweeping robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 300–306.

[46] G. Zamanakos, L. Tsochatzidis, A. Amanatiadis, and I. Pratikakis, "A cooperative LiDAR-camera scheme for extrinsic calibration," in *Proc. IEEE 14th Image, Video, Multidimensional Signal Process. Workshop (IVMSP)*, Jun. 2022, pp. 1–5.

[47] J. Domhof, J. F. P. Kooij, and D. M. Gavrila, "A joint extrinsic calibration tool for radar, camera and LiDAR," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 3, pp. 571–582, Sep. 2021.

[48] G. An, S. Lee, M.-W. Seo, K. Yun, W.-S. Cheong, and S.-J. Kang, "Charuco board-based omnidirectional camera calibration method," *Electronics*, vol. 7, no. 12, p. 421, Dec. 2018.

[49] S. Mishra, G. Pandey, and S. Saripalli, "Extrinsic calibration of a 3D-LiDAR and a camera," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1765–1770.

[50] Q. Sun, "An improved Harris corner detection algorithm," in *Proc. 7th Int. Conf. Commun., Signal Process., Syst.* Cham, Switzerland: Springer, 2020, pp. 105–110.

[51] J. Ou, P. Huang, J. Zhou, Y. Zhao, and L. Lin, "Automatic extrinsic calibration of 3D LiDAR and multi-cameras based on graph optimization," *Sensors*, vol. 22, no. 6, p. 2221, Mar. 2022.

[52] W. Xiong, W. Tian, Z. Yang, X. Niu, and X. Nie, "Improved fast corner-detection method," *J. Eng.*, vol. 2019, no. 19, pp. 5493–5497, Oct. 2019.

[53] H. Zhao, Y. Chen, and R. Shibasaki, "An efficient extrinsic calibration of a multiple laser scanners and cameras' sensor system on a mobile platform," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2007, pp. 422–427.

[54] D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2007, pp. 4164–4169.

[55] R. Raguram and J. M. M. F. Pollefeys, "A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus," in *Proc. 10th Eur. Conf. Comput. Vis.*, Marseille, France: Springer, Oct. 2008, pp. 500–513.

[56] A. D. Nguyen, T. M. Nguyen, and M. Yoo, "Improvement to LiDAR-camera extrinsic calibration by using 3D-3D correspondences," *Optik*, vol. 259, Jun. 2022, Art. no. 168917.

[57] S. Pan and X. Wang, "A survey on perspective-n-point problem," in *Proc. 40th Chin. Control Conf. (CCC)*, Jul. 2021, pp. 2396–2401.

[58] F. Youyang, W. Qing, Y. Yuan, and Y. Chao, "Robust improvement solution to perspective-n-point problem," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 6, 2019, Art. no. 1729881419885700.

[59] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate $o(n)$ solution to the pnp problem," *Int. J. Comput. Vis.*, vol. 81, pp. 155–166, Jul. 2008.

[60] A. Vakhitov, J. Funke, and F. Moreno-Noguer, "Accurate and linear time pose estimation from points and lines," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Sep. 2016, pp. 583–599.

[61] Y. Zhang, Y. Zhang, B. Hu, Y. Yin, W. Chen, X. Liu, and Q. Yu, "An efficient and accurate solution to camera pose estimation problem from point and line correspondences based on null space analysis," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 3762–3769.

[62] R. Sheffer and A. Wiesel, "PnP-Net: A hybrid Perspective-n-Point network," 2020, *arXiv:2003.04626*.

[63] B. Jian and B. C. Vemuri, "A robust algorithm for point set registration using mixture of Gaussians," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Oct. 2005, pp. 1246–1251.

[64] M. Velas, M. Španěl, Z. Materna, and A. Herout, "Calibration of RGB camera with Velodyne LiDAR," in *Proc. WSCG Commun. Papers*, vol. 2014. Union Agency, 2014, pp. 135–144.

[65] H. Cai, W. Pang, X. Chen, Y. Wang, and H. Liang, "A novel calibration board and experiments for 3D LiDAR and camera calibration," *Sensors*, vol. 20, no. 4, p. 1130, Feb. 2020.

[66] J.-E. Ha, "Extrinsic calibration of a camera and laser range finder using a new calibration structure of a plane with a triangular hole," *Int. J. Control, Autom. Syst.*, vol. 10, no. 6, pp. 1240–1244, Dec. 2012.

[67] T. Tóth, Z. Pusztai, and L. Hajder, "Automatic LiDAR-camera calibration of extrinsic parameters using a spherical target," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 8580–8586.

[68] J. Kümmerle, T. Kühner, and M. Lauer, "Automatic calibration of multiple cameras and depth sensors with a spherical target," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–8.

[69] Z. Taylor and J. Nieto, "Automatic calibration of LiDAR and camera images using normalized mutual information," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2013.

[70] Z. Bai, G. Jiang, and A. Xu, "LiDAR-camera calibration using line correspondences," *Sensors*, vol. 20, no. 21, p. 6319, Nov. 2020.

[71] X. Li, Y. Xiao, B. Wang, H. Ren, Y. Zhang, and J. Ji, "Automatic targetless LiDAR-camera calibration: A survey," *Artif. Intell. Rev.*, vol. 56, pp. 9949–9987, Nov. 2022.

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

[73] R. Wang, F. P. Ferrie, and J. Macfarlane, "Automatic registration of mobile LiDAR and spherical panoramas," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 33–40.

[74] Z. Taylor and J. Nieto, "A mutual information approach to automatic calibration of camera and LiDAR in natural environments," in *Proc. Australas. Conf. Robot. Automat.*, Dec. 2012, pp. 3–5.

[75] Y. Zhao, Y. Wang, and Y. Tsai, "2D-image to 3D-range registration in urban environments via scene categorization and combination of similarity measurements," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1866–1872.

[76] K. Irie, M. Sugiyama, and M. Tomono, "Target-less camera-LiDAR extrinsic calibration using a bagged dependence estimator," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2016, pp. 1340–1347.

[77] P. Jiang, P. Osteen, and S. Saripalli, "SemCal: Semantic LiDAR-camera calibration using neural mutual information estimator," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Sep. 2021, pp. 1–7.

[78] G. B. R. Boyle, B. P. D. Koracin, P. R. Ä. Nefian, G. M. V. Pascucci, J. Z. J. Molineros, and H. T. T. Malzbender, *Advances in Visual Computing*. Berlin, Germany: Springer, 2012.

[79] J. Lorandel, J.-C. Prévotet, and M. Hélard, "Fast power and performance evaluation of FPGA-based wireless communication systems," *IEEE Access*, vol. 4, pp. 2005–2018, 2016.

[80] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[81] C. E. Shannon, "A mathematical theory of communication," in *Mobile Computing and Communications Review*, vol. 5, no. 1. New York, NY, USA, 2001, pp. 3–55.

[82] M. Li, X. Chen, X. Li, B. Ma, and P. M. B. Vitányi, "The similarity metric," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3250–3264, Dec. 2004.

[83] M. Guislain, J. Digne, R. Chaine, and G. Monnier, "Fine scale image registration in large-scale urban LiDAR point sets," *Comput. Vis. Image Understand.*, vol. 157, pp. 90–102, Apr. 2017.

[84] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle swarm optimization: A comprehensive survey," *IEEE Access*, vol. 10, pp. 10031–10061, 2022.

[85] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.

[86] J. Nocedal and S. J. Wright, "Conjugate gradient methods," in *Numerical Optimization*. New York, NY, USA: Springer, 2006, pp. 101–134.

[87] C. T. Kelley, *Iterative Methods for Optimization*. Philadelphia, PA, USA: SIAM, 1999.

[88] F. Igelbrink, T. Wiemann, S. Pütz, and J. Hertzberg, "Markerless ad-hoc calibration of a hyperspectral camera and a 3D laser scanner," in *Proc. 15th Int. Conf. Intell. Auton. Syst.* Springer, 2019, pp. 748–759.

[89] G. Pascoe, W. Maddern, and P. Newman, "Direct visual localisation and calibration for road vehicles in changing city environments," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 98–105.

[90] G. Pandey, J. McBride, S. Savarese, and R. Eustice, "Automatic targetless extrinsic calibration of a 3D LiDAR and camera by maximizing mutual information," in *Proc. AAAI Conf. Artif. Intell.*, vol. 26, no. 1, 2012, pp. 2053–2059.

[91] T. Li, J. Fang, Y. Zhong, D. Wang, and J. Xue, "Online high-accurate calibration of RGB+3D-LiDAR for autonomous driving," in *Proc. 9th Int. Conf. Image Graph.*, Shanghai, China: Springer, Dec. 2017, pp. 254–263.

[92] Y. Zhu, C. Li, and Y. Zhang, "Online camera-LiDAR calibration with sensor semantic information," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4970–4976.

[93] A. Willis and Y. Sui, "An algebraic model for fast corner detection," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2296–2302.

[94] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 1150–1157.

[95] H. Ma, K. Liu, J. Liu, H. Qiu, D. Xu, Z. Wang, X. Gong, and S. Yang, "Simple and efficient registration of 3D point cloud and image data for an indoor mobile mapping system," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 38, no. 4, pp. 579–586, 2021.

[96] I. Sobel and G. Feldman, "A 3 × 3 isotropic gradient operator for image processing," presented at the Stanford Artif. Intell. Project (SAIL), Stanford, CA, USA, 1968, pp. 271–272.

[97] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[98] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A line segment detector," *Image Process. Line*, vol. 2, pp. 35–55, Mar. 2012.

[99] J. Peršić, L. Petrović, I. Marković, and I. Petrović, "Online multi-sensor calibration based on moving object tracking," *Adv. Robot.*, vol. 35, nos. 3–4, pp. 130–140, Feb. 2021.

[100] X. Liu, Z. Deng, and Y. Yang, "Recent progress in semantic image segmentation," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 1089–1106, Aug. 2019.

[101] T. Ma, Z. Liu, G. Yan, and Y. Li, "CRLF: Automatic calibration and refinement based on line feature for LiDAR and camera in road scenes," 2021, *arXiv:2103.04558*.

[102] J. Li-Chee-Ming and C. Armenakis, "Fusion of optical and terrestrial laser scanner data," in *Proc. Can. Geomatics Conf. Symp. Commission I, ISPRS Converg. Geomatics-Shaping Canada's Competitive Landscape*, 2010, pp. 1–6.

[103] W. Wang, S. Nobuhara, R. Nakamura, and K. Sakurada, "SOIC: Semantic online initialization and calibration for LiDAR and camera," 2020, *arXiv:2003.04260*.

[104] C.-M. Hsu, H.-T. Wang, A. Tsai, and C.-Y. Lee, "Online recalibration of a camera and LiDAR system," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 4053–4058.

[105] X. Zhang, S. Zhu, S. Guo, J. Li, and H. Liu, "Line-based automatic extrinsic calibration of LiDAR and camera," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 9347–9353.

[106] M. Alba, L. Barazzetti, M. Scaioni, and F. Remondino, "Automatic registration of multiple laser scans using panoramic RGB and intensity images," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 38, pp. 49–54, Sep. 2012.

[107] X. Zhang, A. Zhang, and X. Meng, "Automatic fusion of hyperspectral images and laser scans using feature points," *J. Sensors*, vol. 2015, Jul. 2015, Art. no. 415361.

[108] S. Hofmann, D. Eggert, and C. Brenner, "Skyline matching based camera orientation from images and mobile mapping point clouds," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 5, pp. 181–188, May 2014.

[109] R. Ishikawa, T. Oishi, and K. Ikeuchi, "LiDAR and camera calibration using motions estimated by sensor fusion odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 7342–7349.

[110] C. Park, P. Moghadam, S. Kim, S. Sridharan, and C. Fookes, "Spatiotemporal camera-LiDAR calibration: A targetless and structureless approach," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1556–1563, Apr. 2020.

[111] C. Shi, K. Huang, Q. Yu, J. Xiao, H. Lu, and C. Xie, "Extrinsic calibration and odometry for camera-LiDAR systems," *IEEE Access*, vol. 7, pp. 120106–120116, 2019.

[112] Z. Taylor and J. Nieto, "Parameterless automatic extrinsic calibration of vehicle mounted LiDAR-camera systems," in *Proc. Int. Conf. Robot. Automat., Long Term Autonomy Workshop*, Oct. 2014, pp. 3–6.

[113] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," *Proc. SPIE*, vol. 1611, pp. 586–606, Apr. 1992.

[114] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot., Sci. Syst. Conf.*, vol. 2. Berkeley, CA, USA: Univ. of California, 2014, p. 9.

[115] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1433–1450, Oct. 2021.

[116] Z. Taylor and J. Nieto, "Motion-based calibration of multimodal sensor arrays," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 4843–4850.

[117] Z. Taylor and J. Nieto, "Motion-based calibration of multimodal sensor extrinsics and timing offset estimation," *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1215–1229, Oct. 2016.

[118] H. Xu, G. Lan, S. Wu, and Q. Hao, "Online intelligent calibration of cameras and LiDARs for autonomous driving systems," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3913–3920.

[119] Q. Liao and M. Liu, "Extrinsic calibration of 3D range finder and camera without auxiliary object or human intervention," in *Proc. IEEE Int. Conf. Real-Time Comput. Robot. (RCAR)*, Aug. 2019, pp. 42–47.

[120] J. Iglhaut, C. Cabo, S. Puliti, L. Piermattei, J. O'Connor, and J. Rosette, "Structure from motion photogrammetry in forestry: A review," *Current Forestry Rep.*, vol. 5, no. 3, pp. 155–168, Sep. 2019.

[121] A. Swart, J. Broere, R. Veltkamp, and R. Tan, "Refined non-rigid registration of a panoramic image sequence to a LiDAR point cloud," in *Proc. ISPRS Conf. Photogramm. Image Anal.* Berlin, Germany: Springer, Oct. 2011, pp. 73–84.

[122] W. Moussa, M. Abdel-Wahab, and D. Fritsch, "Automatic fusion of digital images and laser scanner data for heritage preservation," in *Proc. 4th Int. Conf. Prog. Cultural Heritage Preservation*. Limassol, Cyprus: Springer, Oct./Nov. 2012, pp. 76–85.

[123] L. Wang, Z. Xiao, D. Zhao, T. Wu, and B. Dai, "Automatic extrinsic calibration of monocular camera and LiDAR in natural scenes," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, Aug. 2018, pp. 997–1002.

[124] J. Li, B. Yang, C. Chen, R. Huang, Z. Dong, and W. Xiao, "Automatic registration of panoramic image sequence and mobile laser scanning data using semantic features," *ISPRS J. Photogramm. Remote Sens.*, vol. 136, pp. 41–57, Feb. 2018.

[125] B. Nagy, L. Kovács, and C. Benedek, "SFM and semantic information based online targetless camera-LiDAR self-calibration," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 1317–1321.

[126] J. Li and G. Hee Lee, "DeepI2P: Image-to-point cloud registration via deep classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15955–15964.

[127] M. Feng, S. Hu, M. H. Ang, and G. H. Lee, "2D3D-MatchNet: Learning to match keypoints across 2D image and 3D point cloud," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 4790–4796.

[128] A. Bonarini, W. Burgard, G. Fontana, M. Matteucci, D. G. Sorrenti, and J. D. Tardos, "Rawseeds: Robotics advancement through Web-publishing of sensorial and elaborated extensive data sets," in *Proc. IROS*. vol. 6, 2006, p. 93.

[129] V. Mohanty, S. Agrawal, S. Datta, A. Ghosh, V. D. Sharma, and D. Chakravarty, "DeepVO: A deep learning approach for monocular visual odometry," 2016, *arXiv:1611.06069*.

[130] S. Zhao, Y. Chen, and J. A. Farrell, "High-precision vehicle navigation in urban environments using an MEM's IMU and single-frequency GPS receiver," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2854–2867, Oct. 2016.

[131] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[132] B. Zhang and R. T. Rajan, "Multi-FEAT: Multi-feature edge alignment for targetless camera-LiDAR calibration," 2022, *arXiv:2207.07228*.

[133] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2633–2642.

[134] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The ApolloScape open dataset for autonomous driving and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2702–2719, Oct. 2020.
[135] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, S. Chennupati, M. Uricar, S. Milz, M. Simon, K. Amende, C. Witt, H. Rashed, S. Nayak, S. Mansoor, P. Varley, X. Perrotton, D. Odea, and P. Pérez, "WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9307–9317.
[136] J. Mao, M. Niu, C. Jiang, H. Liang, J. Chen, X. Liang, Y. Li, C. Ye, W. Zhang, Z. Li, J. Yu, H. Xu, and C. Xu, "One million scenes for autonomous driving: ONCE dataset," 2021, *arXiv:2106.11037*.
[137] Y. Zhao, X. Liang, X. Fan, Y. Wang, M. Yang, and F. Zhou, "MVSec: Multi-perspective and deductive visual analytics on heterogeneous network security data," *J. Visualizat.*, vol. 17, no. 3, pp. 181–196, Aug. 2014.
[138] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. Kaesemodel Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," 2023, *arXiv:2301.00493*.
[139] X. Ye, M. Shu, H. Li, Y. Shi, Y. Li, G. Wang, X. Tan, and E. Ding, "Rope3D: The roadside perception dataset for autonomous driving and monocular 3D object detection task," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 21309–21318.
[140] N. Schneider, F. Piewak, C. Stiller, and U. Franke, "RegNet: Multimodal sensor registration using deep neural networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1803–1810.
[141] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.
[142] H. Liu, Y. Liu, X. Gu, Y. Wu, F. Qu, and L. Huang, "A deep-learning based multi-modality sensor calibration method for USV," in *Proc. IEEE 4th Int. Conf. Multimedia Big Data (BigMM)*, Sep. 2018, pp. 1–5.
[143] J. Shi, Z. Zhu, J. Zhang, R. Liu, Z. Wang, S. Chen, and H. Liu, "CalibRCNN: Calibrating camera and LiDAR by recurrent convolutional neural network and geometric constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10197–10202.
[144] Y. Sun, J. Li, Y. Wang, X. Xu, X. Yang, and Z. Sun, "ATOP: An attention-to-optimization approach for automatic LiDAR-camera calibration via cross-modal object matching," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 1, pp. 696–708, Jan. 2023.
[145] G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna, "CalibNet: Geometrically supervised extrinsic calibration using 3D spatial transformer networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1110–1117.
[146] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[147] X. Lv, B. Wang, Z. Dou, D. Ye, and S. Wang, "LCCNet: LiDAR and camera self-calibration using cost volume network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 2888–2895.
[148] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

ZHIEN LIU received the B.S. degree in mechanical engineering from the Wuhan University of Engineering, in 1999, and the M.S. and Ph.D. degrees in power machinery and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2003 and 2007, respectively.

He is currently an Associate Professor with the School of Automotive Engineering, Wuhan University of Technology. His current research interests include signal processing and adaptive filtering.

ZHENWEI CHEN received the B.E. degree in vehicle engineering from the Ningbo University of Technology, Ningbo, China, in 2022. She is currently pursuing the master's degree with the School of Automotive Engineering, Wuhan University of Technology, Wuhan, China.

Her research interests include autonomous vehicles and perception.

XIAOXU WEI received the B.S. degree in automation and the M.S. degree in control science and engineering from the Wuhan University of Technology, Wuhan, China, in 2011 and 2014, respectively, where she is currently pursuing the Ph.D. degree in automotive electronics with the School of Automotive Engineering.

Her research interests include robotic machining and machine learning.

WAN CHEN received the B.S. and Ph.D. degrees in automotive engineering from the Wuhan University of Science and Technology, China, in 2016 and 2022, respectively.

From August 2021 to August 2022, she was a sponsored Researcher with the School of Engineering, University of Birmingham, U.K. She is currently a Postdoctoral Researcher with the School of Automotive Engineering, Wuhan University of Technology. Her research interests include signal processing and adaptive filtering.

YONGSHENG WANG (Member, IEEE) received the B.E. degree in automation from the Wuhan University of Technology, Wuhan, China, in 2011, and the M.S. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology, Wuhan, in 2014.

He is currently a Lecturer with the School of Information Engineering, Wuhan University of Technology. His research interests include autonomous vehicles and perception.