Robotica (2026), 1–21
doi:10.1017/S0263574726104093

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Design and evaluation of deep learning-based arbitrary multiple camera simultaneous localization and mapping system

Kang Li[1], Xuan An[1], Dewei Han[1], Jian Xu[1], Junjie Li[1] and Zhaoyuan Ma[1,2]

[1]School of Automation and Intelligent Manufacturing, Southern University of Science and Technology, China
[2]University College, Korea University, Republic of Korea
Corresponding author: Zhaoyuan Ma; Email: 12150147@mail.sustech.edu.cn

Received: 5 November 2025 UTC; **Revised:** 29 July 2026 UTC; **Accepted:** 1 October 2026 UTC

Keywords: SLAM; arbitrary multi-camera; deep learning; scale recovery

Abstract

Visual simultaneous localization and mapping (SLAM) systems are extensively explored due to their low cost. To tackle more challenging environments, multi-camera SLAM systems with a wider field of view (FoV) have been proposed. However, most of the existing multi-camera SLAM systems still rely on hand-crafted features. Given the remarkable advancements of deep learning techniques in the field of feature extraction, the integration of convolutional neural networks into arbitrary multi-camera SLAM systems represents a research topic worthy of in-depth exploration. In this paper, a Deep learning-based Arbitrary Multi-Camera SLAM system (DAMC-SLAM) is proposed. Specifically, this system leverages convolutional neural networks (CNNs) to perform online extraction of local features and global descriptors, while generating corresponding masks to filter out dynamic objects in the scene. Furthermore, DAMC-SLAM is designed with strong adaptability, as it can be flexibly compatible with an arbitrary number of cameras, regardless of whether there exist overlapping FoVs between the deployed cameras. Six self-collected sequences under various environmental conditions, together with the EuRoC MAV benchmark, were used to evaluate the performance of DAMC-SLAM. Experimental results demonstrate that our system possesses remarkable robustness and superior performance. Compared with other methods, it achieves consistent and successful tracking performance across all self-collected sequences, while concurrently yielding the minimum absolute trajectory error and scale error over these sequences. It also demonstrates competitive performance on the EuRoC MAV benchmark, achieving the lowest average ATE among the evaluated methods.

1. Introduction

In embodied-intelligence applications, a robot must estimate its position while sensing an unfamiliar environment. To address this challenge, visual simultaneous localization and mapping (SLAM) is presented [1–3]. Visual SLAM, which uses cameras as sensors, has received sustained attention because cameras provide rich environmental information at low cost and with low energy consumption [4]. This interest has led to various SLAM systems, such as MonoSLAM [5], PTAM [6], LSD-SLAM [7], and DSO [8], which utilize monocular camera, as well as ORB-SLAM2 [9], which can accommodate various types of cameras. LSD-SLAM and DSO employ direct methods that estimate camera pose by directly minimizing photometric error [10]. However, their assumption of grayscale invariance is difficult to satisfy in practical environments [4]. Feature-based approaches instead recover camera poses and scene structure from extracted feature points, then refine the estimates by minimizing reprojection error [11]. They are generally more robust, but still face challenges in dense vegetation, dynamic scenes, and rapidly changing illumination. Multi-camera systems offer a wider FoV and redundant observations, helping to alleviate these difficulties and improve SLAM robustness. Their cameras need not be centrally mounted; their placement can be adapted to the physical constraints of a mobile platform [12]. This flexibility in

© The Author(s), 2026. Published by Cambridge University Press.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

installation unlocks a realm of innovative applications. For instance, these systems can be effortlessly integrated into diverse devices, such as cars, drones, and robots, and positioned in various orientations to suit specific operational requirements. Consequently, multi-camera SLAM has attracted considerable attention.

Currently, most multi-camera SLAM systems are grounded in traditional feature extraction approaches, such as ORB, SIFT, and FAST. Meanwhile, feature extraction methods based on deep learning have achieved significant development [13]. Experiments have demonstrated that the matching scores and repeatability of the features extracted by deep learning methods exceed those of traditional methods [14]. In addition, the high performance of feature extraction methods enabled by deep learning has already been validated in monocular and stereo SLAM systems [15, 16]. In this paper, we propose a Deep learning-based Arbitrary Multi-Camera SLAM system (DAMC-SLAM). This system completely abandons traditional feature extraction methods and instead utilizes deep learning techniques to extract local features as well as the corresponding global descriptors of images. Furthermore, through simple modifications to the configuration file, this system can be adapted to any number of cameras with nonoverlapping FoV or camera pairs with overlapping FoV. When configured to accommodate any number of camera pairs with overlapping FoV, the system can make full use of the information in the co-visible regions between cameras to recover the scale of the environment. We do not make assumptions about the placement of the cameras, which significantly enhances the flexibility of the system. The main contributions of this paper are as follows:

1. Propose an SLAM system that leverages deep learning to replace traditional features and is capable of adapting to arbitrary multi-camera configurations, including any number of cameras, any number of overlapping FoVs, and arbitrary camera arrangements.
2. Construct a multi-camera data acquisition platform and collect six sequences under various challenging conditions in the real world.
3. Evaluate the accuracy, robustness, and metric scale recovery performance of DAMC-SLAM on the self-collected sequences, assess the impact of different multi-camera configurations, and validate its generalization on the EuRoC MAV benchmark.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 introduces the overview of DAMC-SLAM system. Section 4 elaborates on the key modules of the DAMC-SLAM system. Section 5 presents our experimental results on sequences collected by a multi-camera rig in challenging environments and EuRoC benchmark. Section 6 offers our conclusions based on the findings.

2. Related work

In this section, we discuss the existing related work in the field of visual SLAM from two aspects: camera configurations and feature extraction methods.

2.1. Camera configurations

A considerable corpus of research has been accumulated within the realm of visual SLAM utilizing monocular, stereo, or RGB-D cameras. Notable examples include MonoSLAM [5], PTAM [6], LSD-SLAM [7], DSO [8], ORB-SLAM [17], ORB-SLAM2 [9], ORB-SLAM3 [18], and stereo LSD-SLAM [19]. The SLAM systems that utilize the aforementioned camera configurations have a relatively narrow FoV. This characteristic leads to frequent tracking failures under extreme conditions, including nighttime scenarios, low-texture scenes, dynamic occlusions, dense vegetation, and rapid rotations. To achieve a wider FoV, researchers have begun to adopt monocular panoramic cameras as data input sources. Wang et al. proposed PAL-SLAM, which uses a catadioptric panoramic camera [20]. Caruso et al. extended the LSD-SLAM to a generic panoramic camera model [21] and presented an example of using fisheye

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

cameras in SLAM. PAN-SLAM, proposed by Ji et al., introduced a new panoramic camera model for a multiple fisheye camera rig.

The requirement that all cameras in monocular panoramic SLAM systems must be centrally arranged limits the installation flexibility and cannot fully utilize the overlapping FoV between cameras to recover scale information. Therefore, multi-camera SLAM system has attracted extensive research interest. Castle et al. utilized multiple cameras in their SLAM system, but each camera operated independently as a single camera [22]. MC-PTAM independently tracks each camera and optimizes the pose of the entire system by using the measurements from all cameras [23]. In MultiCo-SLAM [24], the stereo information in the overlapping FoV between adjacent cameras was used to initialize the map with the real environmental scale. However, the stereo information is no longer used after map initialization. ROVO [25] uses four cameras equipped with fisheye lenses as two pairs of stereo facing forward and backward to maximize the stability and accuracy of the pose estimation. Li et al. [26] enhanced the initialization stability of multi-camera SLAM systems under conditions of limited overlapping FoV and inaccurate extrinsic calibrations. In ref. [27], an adaptive initialization scheme was proposed for arbitrary multi-camera SLAM systems. The adoption of arbitrary multi-camera rigs enhances the flexibility and stability of SLAM systems. However, current multi-camera SLAM systems still rely on traditional feature extraction approaches, which leads to insufficient robustness in challenging scenarios.

2.2. Feature extraction methods

In feature-based methods, the extraction of robust features is critical to the stability of visual SLAM systems. Early visual SLAM systems usually relied on hand-crafted features. Tardif et al. [28] used SIFT to extract features from the omnidirectional camera. Kaess and Dellaert used Harris corner detector to identity features from a Multi-camera Rig [12]. The FAST feature was employed in PTAM [6], while ORB-SLAM [17] utilized ORB features, which are a combination of FAST keypoints and BRIEF descriptors. These hand-crafted features are computationally efficient and perform well in many scenarios. To date, a multitude of SLAM systems, with multi-camera SLAM systems in particular, remain reliant on ORB features as their dominant feature extraction method, such as those proposed in ref. [29], MultiCol-SLAM [24], and MCSLAM [30]. However, the hand-crafted features often struggle in challenging environments [15].

With the advancement of deep learning, such techniques have increasingly been incorporated into SLAM systems, for example, in neural visual SLAM [31] and semantic visual SLAM for dynamic environments [32]. Meanwhile, learned features have emerged as a promising alternative to hand-crafted features. Deep learning-based feature extraction methods, such as LIFT [13], SuperPoint [14], and HF-Net [33], have demonstrated superior performance in terms of robustness and repeatability. In light of the aforementioned advancements, researchers have integrated deep learning paradigms into SLAM systems. Zhang et al. employed the SuperPoint network as the feature extraction module to detect features from monocular or stereo cameras; however, they still adopted a vocabulary-based approach for loop closure detection [16]. Similarly, works that adopt deep learning approaches to extract features from monocular or stereo inputs also encompasses GCN-SLAM [34], LIFT-SLAM [35], HFNet-SLAM [15], and DX-SLAM [36].

In the context of multi-camera SLAM systems, the choice of feature extraction method is even more critical due to the increased complexity of the system. Traditional methods like ORB may not fully exploit the potential of multi-camera rigs, while deep learning-based methods can provide richer and more discriminative features, enhancing the overall system performance.

3. System overview

In ORB-SLAM2, the adopted framework partitions the entire system into three threads operating in parallel, which effectively enhances the system efficiency. Accordingly, the overall framework of our

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

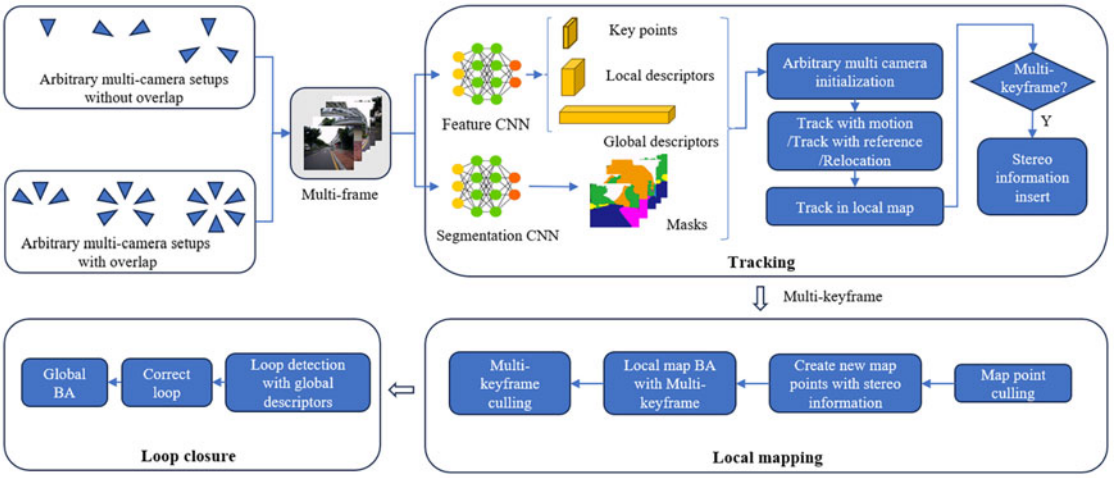


Figure 1. *The overview of DAMC-SLAM. The system is divided into three modules: tracking, local mapping, and loop closure. It uses CNNs to extract features and can accommodate configuration schemes such as monocular camera or arbitrary multi-camera (whether with overlapping FoV or not).*

system, drawing on this design concept, is divided into three modules: tracking, local mapping, and loop closure. Each module has been specifically designed for arbitrary multi-camera configurations to fully leverage the information collected by such devices. The framwork of our DAMC-SLAM is shown in Figure 1.

DAMC-SLAM accommodates arbitrary camera configurations, including monocular cameras and multi-camera systems with either non-overlapping or overlapping FoVs. Dedicated multi-camera and multi-frame structures manage camera architectures and multi-camera data, respectively. The tracking module performs CNN-based image preprocessing, hybrid initialization, multi-camera tracking, multi-camera relocalization, and multi-keyframe selection. Once a multi-frame is selected as a multi-keyframe, stereo information is inserted into it. The local mapping module creates map points in a hybrid manner based on new multi-keyframes, optimizes map points and multi-camera poses through multi-camera observations, and removes redundant map points and multi-keyframes. The loop closure module performs multi-camera loop detection and correction. It replaces the traditional Bag-of-Words method with CNN-generated global descriptors to search for candidate multi-keyframes, which contributes to a more concise code structure.

4. Modules of DAMC-SLAM

4.1. Management of arbitrary multi-camera

In the proposed DAMC-SLAM system, a multi-camera data structure is employed for camera management. Specifically, the intrinsic parameters, extrinsic parameters of each individual camera, as well as the overlapping regions among different cameras are loaded by reading the configuration file. Notably, both the number of cameras and the number of overlapping regions support flexible configuration as required. We define the body coordinate system to associate the data from each camera. Figure 2 shows the transformation relationship between coordinate systems. The process of projecting a 3D point onto the image plane of camera i can be modeled as

$$\lambda \mathbf{P}_i^{'} = K_i(\mathbf{T}_{ib}\mathbf{T}_{bw}\mathbf{P}_w). \tag{1}$$

In this equation, $\mathbf{P}_w$ is a 3D point represented in world coordinate system, K_i is the project function of camera i, and $\mathbf{P}_i^{'}$ is the corresponding 2D point in pixel coordinate system of camera i. The transformation matrix $\mathbf{T}_{ib}$ can be obtained by pre-calibration. $\mathbf{T}_{bw}$ is the multi-camera pose that needs to be solved in the

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

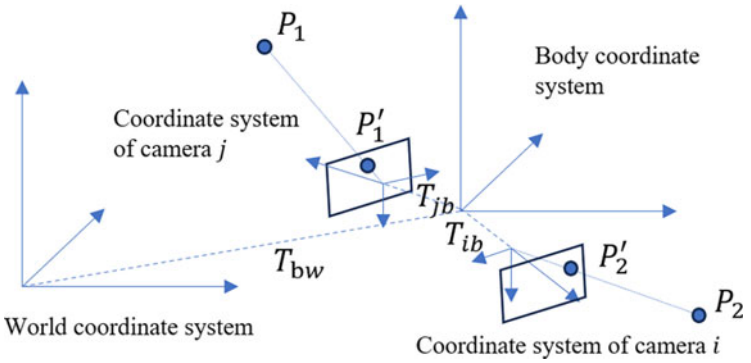


Figure 2. *Transformation relationship between coordinate systems.* $\mathbf{T_{bw}}$ *represents the transformation matrix from world coordinate system to body coordinate system.* $\mathbf{T_{ib}}$ *and* $\mathbf{T_{jb}}$ *represent the transformation matrices from the body coordinate system to the coordinate systems of camera i and j, respectively.* $\mathbf{P_1}$ *and* $\mathbf{P_2}$ *are 3D points, and* $\mathbf{P_1'}$ *and* $\mathbf{P_2'}$ *are corresponding 2D points in pixel coordinate systems.*

SLAM system. DAMC-SLAM supports both fisheye and pinhole camera models. For fisheye cameras, $K_i(\cdot)$ is instantiated using the radially symmetric model proposed by Kannala and Brandt [37] instead of the full model. This approach helps to reduce computational overhead and enhance system efficiency. The Kannala–Brandt projection function can be written as

$$\begin{aligned} r &= \theta + k_1\theta^3 + k_2\theta^5 + k_3\theta^7 + k_4\theta^9, \\ u &= f_x r\cos(\phi) + c_x, \\ v &= f_y r\sin(\phi) + c_y. \end{aligned} \tag{2}$$

In Eq. (2), θ and ϕ are, respectively, the zenith and horizontal angle of the incoming ray. k_i where $i = 1, 2, 3, 4$ and f_x, f_y, c_x, c_y are the intrinsic parameters and can be obtained through pre-calibration.

For pinhole cameras, DAMC-SLAM employs a radial–tangential distortion model. Given the normalized coordinates (x, y), the distorted coordinates and pixel coordinates are computed as

$$\begin{aligned} r^2 &= x^2 + y^2, \\ x_d &= x(1 + k_1r^2 + k_2r^4) + 2p_1xy + p_2(r^2 + 2x^2), \\ y_d &= y(1 + k_1r^2 + k_2r^4) + p_1(r^2 + 2y^2) + 2p_2xy, \\ u &= f_x x_d + c_x, \quad v = f_y y_d + c_y. \end{aligned} \tag{3}$$

Here, k_1 and k_2 are radial distortion coefficients, while p_1 and p_2 are tangential distortion coefficients. For each camera, the corresponding projection and back-projection functions are configured from its calibration parameters; therefore, the remaining modules can operate with either camera model without modification.

4.2. Tracking

4.2.1. CNN-based images preprocessing

The tracking module begins by processing images captured by multiple cameras. Unlike other multi-camera SLAM systems, DAMC-SLAM replaces hand-crafted features with two convolutional neural networks: a feature CNN and a segmentation CNN, as illustrated in Figure 1. HF-Net is used as the feature CNN to extract local feature points, descriptors, and a global image descriptor simultaneously. After threshold-based filtering, the points are ranked by score, and specified top points are kept to avoid interference from unstable features. FastSeg [38] is used as the segmentation CNN to generate masks for all images, facilitating the removal of potential dynamic objects during subsequent matching. Both networks are deployed using TensorRT, and images are processed in batches to improve inference speed.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

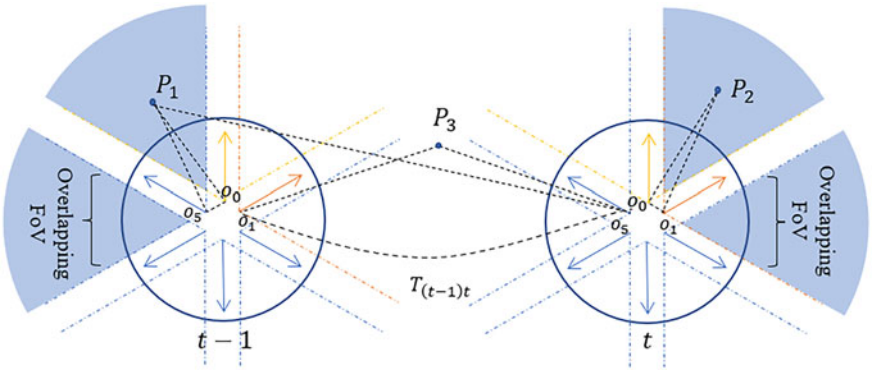


Figure 3. *Illustration of new map point creation. o represents the centers of camera. The arrows indicate the orientation of cameras, while the dashed lines on both sides represent the cameras' FoV.* $\mathbf{P}$ *represents new map points that need to be created.* $\mathbf{P}_1$ *and* $\mathbf{P}_2$ *are created in one multiframe, respectively, and* $\mathbf{P}_3$ *is triangulated through two different multi-frames.* $\mathbf{T}_{(t-1)t}$ *is the transformation matrix from the multi-camera system's body coordinate at time t to that at time* $t-1$.

Given the inherent scale invariance of CNN features, we do not use an image pyramid, thereby further improving the execution efficiency of the module. The resulting data are managed using a multi-frame structure.

4.2.2. Hybrid initialization

In an arbitrary multi-camera system, the creation of map points can be categorized into two types. As illustrated in Figure 3, for map points located within the overlapping FoV of multi-camera system (e.g., $\mathbf{P}_1$ and $\mathbf{P}_2$), their 3D coordinates can be reconstructed via stereo triangulation. For map points that do not lie within such overlapping regions (e.g., $\mathbf{P}_3$), initialization can be achieved using structure from motion (SfM) method based on consecutive observations across two multi-frames.

To accommodate arbitrary camera configurations, including monocular cameras, multi-camera with non-overlapping FoV, and multi-camera with overlapping FoV, a hybrid initialization method is proposed in this paper. The pipeline of the hybrid initialization is illustrated in Figure 4.

When the system receives the initial multi-frame, the hybrid initialization module first performs stereo initialization. To improve efficiency, KNN matching is limited to features in the predefined overlapping regions specified in the configuration file. After a matched feature pair passes the epipolar constraint check, triangulation is performed according to $s_i \mathbf{x}_j^{\wedge} \mathbf{T}_{ji} \mathbf{x}_i = 0$, where s_i is the depth in camera i. $\mathbf{x}_i$ and $\mathbf{x}_j$ are the coordinates of the same 3D map point on the unit image planes of cameras i and j, respectively, obtained by applying the corresponding camera back-projection functions $K^{-1}(\cdot)$. The transformation $\mathbf{T}_{ji}$ is calculated as $T_{ji} = T_{jb} T_{ib}^{-1}$ from the pre-calibrated extrinsic parameters. Thus, s_i is the only unknown and can be solved directly. Map points that pass the depth and reprojection-error checks in both cameras are successfully triangulated. To accelerate the triangulation speed, the computations for different overlapping regions are performed in separate threads. For each triangulated map point, we store its depth and parallax angle for subsequent map point creation in the local mapping module. If more than 200 map points are triangulated, they are inserted into the map. The first multi-frame is designated as a multi-keyframe, and its pose is set to the world coordinate system. Due to the use of stereo information, the real scale of the environment was introduced into the map.

Stereo initialization often fails when cameras have no overlapping FoVs or when too few feature points are available in the overlapping regions. This can occur with a limited number of cameras, small overlapping FoVs, or low-texture regions. In such cases, the hybrid initialization module matches features between two multi-frames and estimates the motion of each camera using SfM. After successful

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

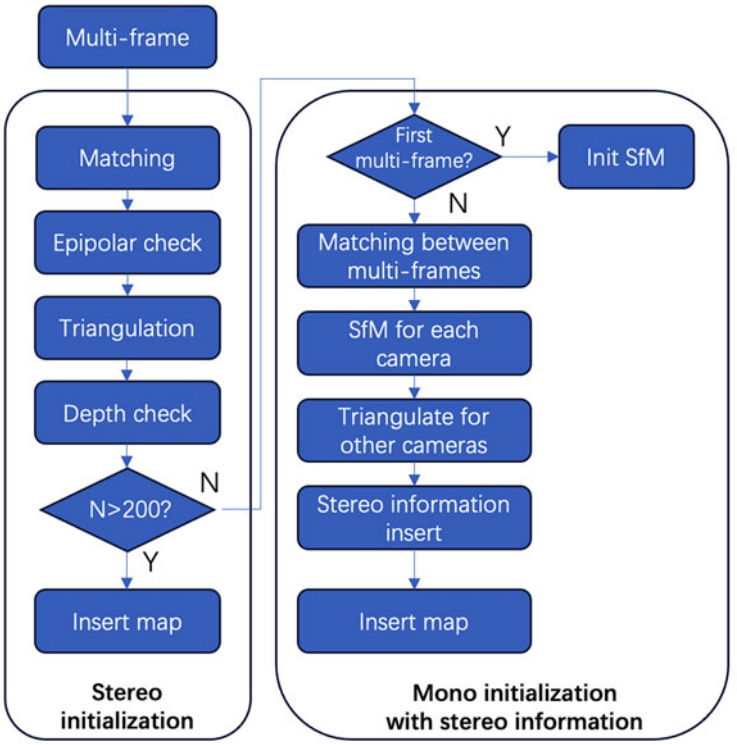


Figure 4. *Pipeline of hybrid initialization.*

motion estimation, triangulation is applied not only to features in the current camera but also to features from the other cameras. This significantly increases the number of map points generated during initialization and improves system stability. Finally, triple-observation map points are identified from the two multi-frames and added to the map. A triple-observation map point is defined as a point that is observed twice within one multi-frame and at least once in another multi-frame, such as $\mathbf{P}_1$ in Figure 3. Through this mechanism, scale information is inserted into the map. The underlying principle is that two observations within one multi-frame enable the determination of its true coordinates with scale information; meanwhile, the observation in another multi-frame allows for the estimation of the actual motion distance of the multi-camera system.

4.2.3. Pose prediction and tracking in local map

We employ a constant-velocity motion model to predict the pose of the current multi-frame based on the pose of the previous multi-frame. Leveraging the predicted pose, the points in the map are projected onto the current multi-frame in accordance with Eq. (1), followed by a search for matching feature points within the adjacent region. In a multi-camera system, due to the distinct orientations of individual cameras, the displacement distances of feature points across different directions in the image vary significantly. Specifically, feature points where the connection line between the map points and the body coordinate system is parallel to the motion direction exhibit the minimum displacement distance, whereas those with the connection line perpendicular to the motion direction demonstrate the maximum displacement distance. Based on this observation, we assign a unique search distance to each map point individually, which is determined by the angle between the connection line and the motion direction. The calculation of this search distance follows the formula:

$$r = \left(2 - \left|cos(\overrightarrow{bp}, \overrightarrow{v})\right|\right) r_{pre}, \tag{4}$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

where $\overrightarrow{bp}$ denotes the vector from the body coordinate system of the multi-camera system to a map point, $\overrightarrow{v}$ represents the velocity vector, and r_{pre} stands for the preset search distance.

At the final stage of feature matching, feature points associated with potential dynamic objects are filtered using the segmentation masks to reduce their interference with pose optimization. Not all such feature points are removed. When the semantic mask associated with a potential dynamic label occupies only a small proportion of the image, it contains relatively few feature points and has little effect on pose optimization. In addition, stationary objects, such as parked vehicles, can provide useful features for tracking. In contrast, a large semantic mask associated with a moving object can substantially degrade pose optimization or even cause tracking failure. We therefore compute the area of the semantic mask associated with each potential dynamic label relative to the whole image. Only those feature points associated with masks whose area exceeds a predefined threshold are removed. Based on empirical observations, this threshold is set to one-sixth of the image area. This selective strategy reduces the influence of undesirable matches while retaining useful features.

DAMC-SLAM optimizes the predicted pose of the multi-camera system based on the matching results using the bundle adjustment method. The optimal pose solution for the multi-camera system is expressed as follows:

$$\mathbf{T}_{bw}^{*} = \underset{\mathbf{T}_{bw}}{argmin} \sum_{i \in \mathbf{C}} \sum_{j \in \mathbf{M}_i} \rho\left(\|\mathbf{m}_i^j - K_i(\mathbf{T}_{ib}\mathbf{T}_{bw}\mathbf{P}_w)\|_{\sum_{(i,j)}}\right), \tag{5}$$

where $\mathbf{C}$ denotes the set of camera indices, and $\mathbf{M}_i$ denotes the set of measurement indices for camera i. $\mathbf{m}_i^j$ is the j-th measurement of a map point in camera i, and $\mathbf{P}_w$ is the corresponding 3D map point in the world coordinate system. $\mathbf{T}_{ib}$ is the transformation from the body coordinate system to camera i, and $K_i(\cdot)$ is the projection function of camera i. $\mathbf{T}_{bw}$ is the transformation from the world coordinate system to the body coordinate system which needs to be optimized. $\rho(\cdot)$ is the robust Huber function. Because DAMC-SLAM does not use an image pyramid, the CNN feature scores are normalized and used as the information matrix $\sum_{(i,j)}$.

The process of tracking in local map is analogous to that of pose prediction. The key distinction lies in that, besides matching map points derived from the previous multi-frame, it further retrieves additional map points from the local map–this supplementary step is intended to enhance the accuracy of pose optimization.

4.2.4. Multi-camera relocalization

Relocalization algorithms estimate the current pose by searching all map points when tracking fails. Traditional visual SLAM systems leverage the BoW to search for candidate keyframes across the entire map. However, a large body of research has demonstrated that CNN-based methods outperform the BoW method [15]. In our approach, we employ a CNN to generate a 4,096-dimensional vector for each camera frame, which serves as the global descriptor to retrieve all multi-keyframes within the map. To accommodate variations in the number of cameras in an arbitrary multi-camera system, we calculate the distance between the current multi-frame and each multi-keyframe in the map using the formula

$$distance = \frac{\sum_{i \in \mathbf{C}} cos(\mathbf{v}_{fi}, \mathbf{v}_{kfi})}{|\mathbf{C}|}. \tag{6}$$

where $\mathbf{v}_{fi}$ and $\mathbf{v}_{kfi}$ denote the global descriptors generated by the camera i in the current multi-frame and multi-keyframe, respectively. After acquiring the candidate multi-keyframes, the system is required to match the map points contained in the candidate multi-keyframes with the features in the current multi-frame, and then calculate the pose of the current multi-camera system by leveraging the matching results. In Mono SLAM system, the EPnP algorithm was applied to calculate the initial pose in a RANSAC loop. However, this algorithm was designed for a single camera and can not work with our DAMC-SLAM system. To address the multi-camera relocalization problem, inspired by the work of Kneip et al. [39], we extended EPnP by incorporating the extrinsic parameters of cameras.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

Each map point in the world coordinate system can be represented by a weighted sum of 4 control points shared by all map points. The relationship also holds in the body coordinate system and can be written as

$$\mathbf{P}_i^b = \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_j^b = \sum_{j=1}^{4} \alpha_{ij} \begin{bmatrix} x_j^b \\ y_j^b \\ z_j^b \end{bmatrix}, \tag{7}$$

where P_i^b is the i-th map point's body coordinates and the weights α_{ij} can be calculated in the world coordinate system. With the transformation matrix from the body coordinate system to the camera coordinate system, we can get

$$\lambda_{ik} \begin{bmatrix} \mathbf{p}_{ik}^f \\ 1 \end{bmatrix} = \mathbf{R}^{kb} \mathbf{P}_i^b + \mathbf{t}^{kb}, \tag{8}$$

where $\mathbf{p}_{ik}^f = [x_{ik}^f, y_{ik}^f]^T$ is the coordinates of the corresponding map point on the unit image plane of the camera. Let $\mathbf{P}_i^b = [x_i^b, y_i^b, z_i^b]^T$. The last row of Eq. (8) implies that $\lambda_{ik} = R_{31}^{kb} x_i^b + R_{32}^{kb} y_i^b + R_{33}^{kb} z_i^b + t_3^{kb}$. Substituting this formula into the first two lines of Eq. (8) and combining with Eq. (7) yields two linear equations for each map point:

$$\begin{aligned} &\sum_{j=1}^{4} \alpha_{ij} \left(R_{31}^{kb} x_{ik}^f - R_{11}^{kb}\right) x_j^b + \alpha_{ij} \left(R_{32}^{kb} x_{ik}^f - R_{12}^{kb}\right) y_j^b + \alpha_{ij} \left(R_{33}^{kb} x_{ik}^f - R_{13}^{kb}\right) z_j^b = t_1^{kb} - x_{ik}^f t_3^{kb}, \\ &\sum_{j=1}^{4} \alpha_{ij} \left(R_{31}^{kb} y_{ik}^f - R_{21}^{kb}\right) x_j^b + \alpha_{ij} \left(R_{32}^{kb} y_{ik}^f - R_{22}^{kb}\right) y_j^b + \alpha_{ij} \left(R_{33}^{kb} y_{ik}^f - R_{23}^{kb}\right) z_j^b = t_2^{kb} - y_{ik}^f t_3^{kb}. \end{aligned} \tag{9}$$

Thus, by combining these for all n map points, we construct a linear system of the form

$$\mathbf{M}\mathbf{x} = \mathbf{b} \text{ or } \mathbf{M}^T \mathbf{M}\mathbf{x} = \mathbf{M}^T \mathbf{b}, \tag{10}$$

where $\mathbf{M}$ is a $2n \times 12$ matrix, $\mathbf{x} = [(\mathbf{c}_1^b)^T, (\mathbf{c}_2^b)^T, (\mathbf{c}_3^b)^T, (\mathbf{c}_4^b)^T]$ is the 12-vector to be solved, and $\mathbf{b}$ is a $2n$ vector. Note that matrix $\mathbf{M}$ contains no camera projection parameters; therefore, the equation applies to any camera projection model, including that in Eq. (2). Its solution consists of homogeneous and particular components. The homogeneous component is solved as in EPnP, whereas the particular component is obtained by QR decomposition. This determines the coordinates of the control points corresponding to the map points in the body coordinate system. The initial pose matrix $[\mathbf{R}|\mathbf{t}]$ of the multi-camera system is estimated using ICP and then refined by bundle adjustment.

4.2.5. Multi-keyframe decision and stereo information insertion

The multi-keyframe selection mechanism follows ORB-SLAM2. When the number of map points observed by the current multi-frame falls below 90% of that observed by the reference multi-keyframe, the current multi-frame is selected as a new multi-keyframe. Unlike ORB-SLAM2, DAMC-SLAM then inserts stereo information into the multi-keyframe. This information is used by the local mapping module to create new map points. The insertion procedure follows the stereo initialization procedure described above. Rather than immediately inserting triangulated map points and their associated data into the map, the system stores these data temporarily in the multi-keyframe. This enables the local mapping module to select the most suitable data for map-point creation and reduce interference.

4.3. Local mapping

Local mapping leverages local information to optimize camera poses and map points. With the aim of adapting to arbitrary camera configurations, we mainly adjusted the creation of new map points and the local bundle adjustment. While the implementation of other modules draws on ORB-SLAM2,

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

corresponding modifications have been made to adapt to the arbitrary multi-camera configuration, and detailed elaboration thereof is omitted herein.

During map-point creation, we use the co-visibility graph to identify the 10 multi-keyframes most strongly connected to the current multi-keyframe. We then search these multi-keyframes for matching feature points. To reduce computational complexity, global descriptors first perform image-to-image matching between multi-keyframes. Feature matching is subsequently restricted to matched image pairs. This two-step strategy reduces the complexity of feature matching from $O(nm)$ to $O(m)$, where n is the number of images in a multi-keyframe and m is the number of feature points in each image. After that, some feature points are matched only across multi-keyframes, whereas others are also triangulated using stereo information in the current multi-keyframe. For the former, depth is recovered from inter-keyframe matches when the parallax is sufficient. For the latter, depth is recovered either from inter-keyframe matches or from stereo information which is completed after multi-keyframe decision. The method with the larger parallax angle is selected. Depth and reprojection-error checks remove invalid results. Triple-observation map points are inserted continuously during this process, helping to recover map scale and mitigate scale drift. Figure 3 illustrates the procedure. Finally, bundle adjustment optimizes the local map as follows:

$$\mathbf{T}_{bw}^{*}, \mathbf{P}_{w}^{*} = \underset{\mathbf{T}_{bw}, \mathbf{P}_{w}}{argmin} \sum_{l \in \mathbf{L}} \sum_{i \in \mathbf{C}} \sum_{j \in \mathbf{M}_i} \rho\left(\| \mathbf{m}_{l,i}^{j} - K_{l,i}(\mathbf{T}_{l,ib} \mathbf{T}_{bw} \mathbf{P}_{w}) \|_{\sum_{(l,i,j)}} \right). \tag{11}$$

where $\mathbf{L}$ denotes the set of local multi-keyframes indicates.

4.4. Loop closure

Loop correction plays a crucial role in mitigating cumulative errors in SLAM systems. Consistent with the multi-camera relocalization module, the BoW method is not employed in this step either; instead, global descriptors are utilized to search for similar multi-keyframes, which serve as candidate loop closure multi-keyframes. Once a loop is detected, the transform matrix Sim(3) is estimated, and the essential graph is reconstructed. Subsequently, a global optimization is performed on all multi-keyframes and map points. This ensures more accurate and reliable optimization results within our SLAM system.

5. Experiments

We designed and constructed a multi-camera rig to evaluate DAMC-SLAM. The rig integrates six fisheye cameras distributed uniformly around a central axis. Adjacent cameras are separated by 60°, and each camera has an FoV of 120° × 180°. This arrangement provides uniform coverage of the surrounding environment and captures sufficient stereo information. Different camera combinations can be selected to construct the required multi-camera configurations. For the experiments, the forward-facing camera is designated as Camera 0, and the remaining cameras are numbered clockwise. Figure 5a shows the rig, and Figure 5b shows its structural schematic.

We collected data using the multi-camera rig, and the reference trajectories were collected by an integrated navigation system (INS) with horizontal positioning accuracy of ±1 cm + 1 ppm and pitch/roll accuracy of 0.1°. It was used as ground truth for the quantitative assessment of SLAM accuracy. All data were collected with a frame rate of 20 Hz. A computer was used for data processing and recording. We mounted all equipments on a mobile platform, as depicted to Figure 6. Our experiments were conducted on a desktop computer equipped with a 3.6 GHz Intel CPU and 48 GB of RAM. In all tables, bold values indicate the best results for each metric and sequence.

5.1. Datasets

Self-collected datasets: We collected six sequences in different environments for evaluation (Table I). In Sequence 1, the multi-camera rig moves away from buildings and passes through dense vegetation.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

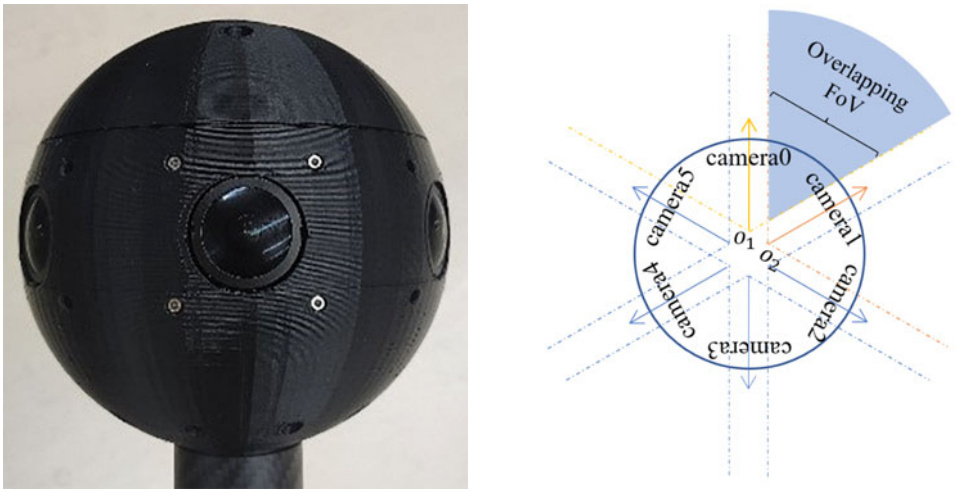


Figure 5. *The multi-camera rig with* 360° × 180° *FoV (a) and the structural schematic diagram (b).*

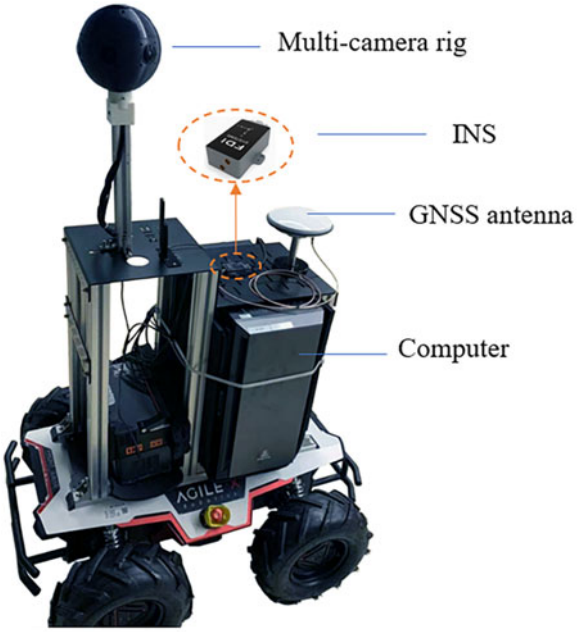


Figure 6. *The mobile platform used for collecting the datasets.*

The sparse, repetitive visual features in this environment challenge visual SLAM. Sequence 2 was collected in an ordinary outdoor environment without deliberately introducing particular difficulties. Since visual SLAM systems accumulate drift over time, loop closure is important for error correction. Sequence 3 was therefore designed to evaluate loop closure: the rig moves around several buildings and returns to a previously observed area. We also collected sequences under more challenging outdoor conditions. Sequence 4 follows a route used by campus buses while buses are passing, producing dynamic-object occlusions. This setting poses significant challenges for purely visual SLAM systems due to the reduction in available static information and the increase in dynamic feature points within the

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

Table I. *Details of our collected datasets.*

Sequences	Length(m)	Image count	Time	Environment
Sequence 1	626.192	10, 238	Day	Outdoors, dense vegetation
Sequence 2	227.829	4, 822	Day	Outdoors
Sequence 3	854.706	12, 922	Day	Outdoors, loop closure
Sequence 4	103.242	1612	Day	Outdoors, dynamic occlusion
Sequence 5	267.568	3, 926	Night	Outdoors, varying light
Sequence 6	–	703	Day	Indoors



(a) (b)

Figure 7. *The environments of sequences and the images from sequences. (a) The environments of sequence 1,2,3,4, and 5. Sequence 5 was collected at night. Sequence 4 includes dynamic object occlusions, and the direction of sequence 2 is opposite to that of sequence 1. (b) The images from sequences.*

camera FoV. Sequence 5 was acquired at night along a road with alternating illuminated and dark sections to evaluate performance under low light and changing illumination. All outdoor data were collected on campus, where pedestrian and vehicle traffic is generally sparse. Figure 7a shows the environments of the five outdoor sequences. For qualitative indoor evaluation, Sequence 6 was collected in a room. Figure 7b presents representative images from each sequence. Except for Sequence 6, all sequences pose substantial challenges to visual SLAM; even the relatively simple sequence 2 exhibits issues such as motion blur.

EuRoC MAV dataset: To complement the self-collected sequences, we also evaluate DAMC-SLAM on the public EuRoC MAV dataset. It comprises 11 indoor sequences acquired in three environments: a machine hall and two Vicon rooms. The sequences are labeled Easy, Medium, or Difficult and encompass diverse motion patterns, illumination conditions, and scene textures. This benchmark therefore provides an additional evaluation of the generalization of DAMC-SLAM beyond our own data.

5.2. Localization accuracy

For comparison, we evaluated DAMC-SLAM alongside ORB-SLAM3 [18], HFNet-SLAM [15], and MultiCo-SLAM [24]. ORB-SLAM3 was tested in monocular and stereo modes. In monocular mode, the forward-facing Camera 0 was used for Sequences 1, 2, 3, and 5. For Sequence 4, Camera 5 was used because dynamic objects are concentrated in its FoV. In stereo mode, Cameras 0 and 5 were used for all sequences. HFNet-SLAM used the same setup as ORB-SLAM3 in monocular mode. MultiCo-SLAM and DAMC-SLAM used all available cameras with overlapping FoVs. The number of extracted features from each image was set to 1,000 for all systems. We evaluated all multi-frame trajectories rather than only multi-keyframe trajectories. Before evaluation, trajectories were aligned with the ground truth

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

Table II. *Comparison of localization accuracy (RMS ATE in m), scale error, and tracking success rate on self-collected*[*].*

Frameworks		**Seq.**[1] **1**	**Seq. 2**	**Seq. 3**	**Seq. 4**	**Seq. 5**
ORB-SLAM3 mono	ATE	4.363	0.585	2.504	5.600	–
	Scale error	2.667	2.870	9.254	16.781	–
	TSR	10/10	9/10	8/10	10/10	0/10
HFNet-SLAM	ATE	7.763	0.495	6.843	11.004	1.056
	Scale error	1.918	1.998	2.531	16.669	14.707
	TSR	10/10	2/10	6/10	9/10	10/10
MultiCo-SLAM	ATE	–	55.799	–	9.442	–
	Scale error	–	0.694	–	2.678	–
	TSR	0/10	4/10	0/10	2/10	0/10
DAMC-SLAM(ours)	ATE	**2.650**	**0.400**	**1.319**	**0.174**	**0.406**
	Scale error	**0.181**	**0.044**	**0.199**	**0.087**	**0.473**
	TSR	**10/10**	**10/10**	**10/10**	**10/10**	**10/10**

[*]All displayed results are derived from the average of track success trials.
[1]"Seq." stands for "Sequence".

using Sim(3). We then calculated the absolute trajectory error (ATE). The scale is the Sim(3) alignment parameter s, and scale error is defined as $|s - 1|$. The tracking success rate (TSR) is the number of successful trials divided by the total number of trials. All reported results are the averages of 10 runs. A trial is considered successful when fewer than 20 frames have failed tracking.

Table II presents the results for the self-collected dataset. ORB-SLAM3 in stereo mode fails to track all sequences successfully because its two cameras must maintain sufficient overlapping FoVs. Its performance metrics are therefore not reported in table. ORB-SLAM3 in monocular mode tracks most sequences successfully, but its performance degrades substantially under dynamic occlusions and illumination changes. In particular, nighttime illumination changes make initialization difficult. HFNet-SLAM exhibits poorer overall ATE performance and unstable tracking on Sequences 2 and 3. However, its learning-based HFNet feature extractor provides strong robustness on Sequence 5 under varying illumination. MultiCo-SLAM also struggles to track all sequences. Although it achieves partial success on Sequences 2 and 4, its ATE remains unsatisfactory. DAMC-SLAM achieves the lowest average ATE and scale error across the evaluated sequences. It successfully tracks all trajectories, including those affected by dynamic-object occlusions and nighttime illumination changes. These results indicate strong robustness under challenging conditions. This result can be primarily attributed to the integration of multi-camera systems and CNNs. Specifically, CNN-based feature points exhibit higher stability, and the extracted local descriptors demonstrate stronger robustness against disturbances induced by variations in viewing angles, lighting conditions, and other confounding factors. In addition, the multi-camera system provides richer environmental information. When tracking fails, the multi-camera relocalization module enables efficient recovery. Together, these characteristics improve significantly the overall accuracy and robustness of the system. DAMC-SLAM also generates maps with scale closest to the ground truth because it effectively uses stereo information. Figure 8 compares the trajectories of the evaluated systems with the ground truth for Sequences 2, 4, and 5. Neither the monocular nor stereo mode of ORB-SLAM3 completed Sequence 5 successfully; their results are therefore omitted from Figure 8(c).

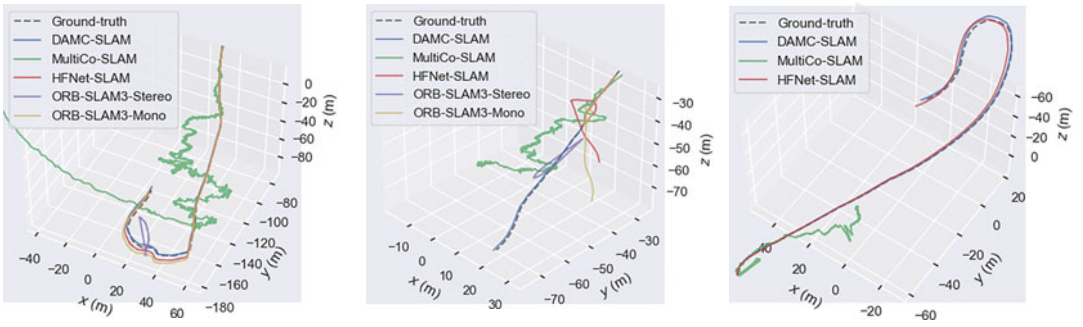
To complement the self-collected sequences and contextualize the performance of DAMC-SLAM on an established benchmark, we conducted experiments on the EuRoC dataset and compared it with ORB-SLAM3 and HFNet-SLAM. All methods used the left camera of the dataset as input, and each sequence was evaluated 10 times with the mean result reported.

Table III reports the RMSE results on the EuRoC MAV dataset. Although DAMC-SLAM is designed for arbitrary multi-camera configurations rather than specifically for monocular operation and does not employ an image pyramid, it achieves performance comparable to the two monocular baselines.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

Table III. *Comparison of localization accuracy (RMS ATE in M) on EuRoC.*

Frameworks	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203	Avg.
ORB-SLAM3	0.022	0.024	0.029	0.281	0.158	0.034	0.035	0.194	0.276	0.024	0.337	0.129
HFNet-SLAM	0.015	0.021	0.043	0.060	0.825	0.032	0.023	0.023	0.019	0.013	0.097	0.106
Ours	0.022	0.023	0.044	0.107	0.062	0.034	0.187	0.030	0.025	0.017	0.401	**0.087**



(a) Sequence 2 (b) Sequence 4 (c) Sequence 5

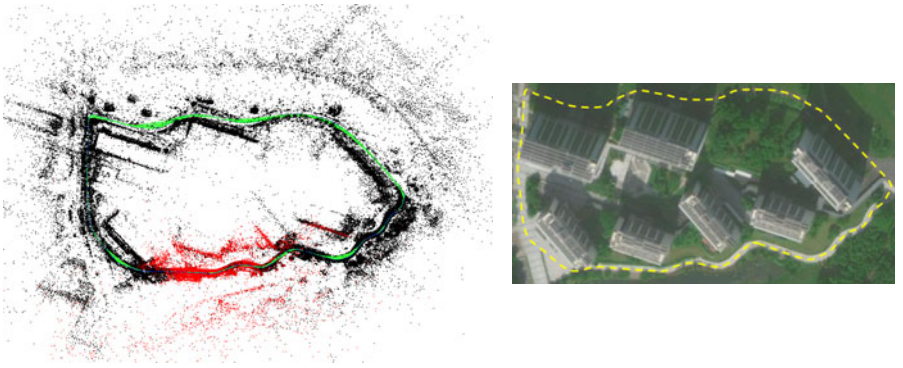
Figure 8. *Comparison of trajectories with the ground truth for sequences 2,4, and 5. Due to the failure of ORB-SLAM3 to operate successfully on sequence 5, it is not presented in (c).*

DAMC-SLAM matches or outperforms ORB-SLAM3 on 8 of the 11 sequences, while the monocular baselines perform better on some individual sequences. Notably, DAMC-SLAM obtains the lowest average RMSE of 0.087 m, compared with 0.106 m for HFNet-SLAM and 0.129 m for ORB-SLAM3. These results indicate that the proposed system retains competitive accuracy even when evaluated with a single-camera input, while its multi-camera design does not compromise its overall generalization on EuRoC.

5.3. Loop closure

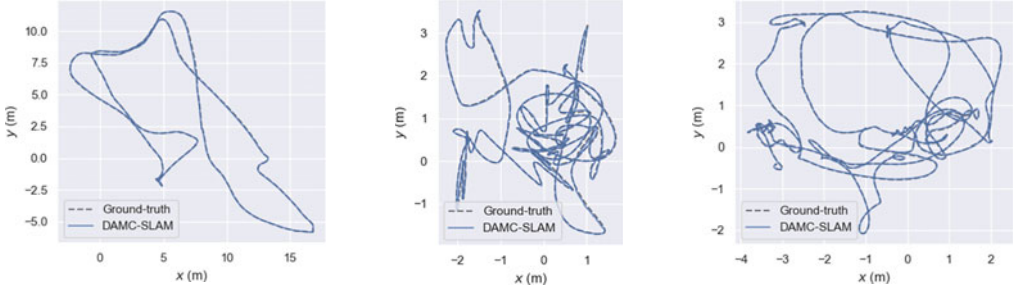
Loop closure is essential for mitigating cumulative errors in SLAM. Sequence 3 contains a loop and is used to evaluate this capability; the quantitative results are reported in Table II. MultiCo-SLAM failed in all 10 trials. Although HFNet-SLAM adopts HFNet, a learning-based feature extractor, it succeeded in only 6 of 10 trials. ORB-SLAM3 in monocular mode succeeded in 8 trials, whereas DAMC-SLAM achieved a 100% success rate. Note that, our DAMC-SLAM does not employ an image pyramid for feature extraction. The achievement of comparable performance can be attributed to two key factors: first, the adoption of CNN-extracted global descriptors for loop closure detection, and second, the rich information provided by the multi-camera system. With respect to the ATE metric, our DAMC-SLAM system achieved an error of 1.319 m, representing an approximate 47% improvement compared to ORB-SLAM3 (monocular mode), which exhibited an ATE of 2.504 m. Compared with HFNet-SLAM, which also adopts the learning-based feature extraction method, it achieves an improvement of 81%. Regarding the scale error, our method reduced it to 0.199. ORB-SLAM3 in monocular mode cannot recover metric scale, and its stereo mode failed because the cameras must maintain sufficient overlapping FoVs. Figure 9 shows maps created by DAMC-SLAM, which can be compared qualitatively with the satellite image. The EuRoC experiments also demonstrate loop-closure capability of DAMC-SLAM under a monocular configuration. Figure 10 shows trajectories of selected EuRoC sequences containing loop closures. In summary, our proposed method not only achieves effective loop closure detection but also leverages loop closure effectively and adequately to correct maps and mitigate cumulative errors.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093



(a) The map created by DAMC-SLAM (b) The satellite image of sequence 3

Figure 9. *Map qualitative assessment. The dashed line in (b) represents the ground truth trajectory of sequence 3.*



(a) MH05 (b) V103 (c) V202

Figure 10. *Loop closure results on the EuRoC dataset.*

5.4. Effect of camera configurations

To evaluate the impact of different camera configurations on the DAMC-SLAM system, experiments were conducted under both overlapping and nonoverlapping FoVs scenarios with varying camera numbers. The test sequences selected included sequence 2 (collected from normal environmental conditions) as well as sequence 4 and sequence 5 (collected from extreme conditions). Each experimental group was repeated 10 times to ensure statistical reliability.

Table IV shows the effect of camera configuration under nonoverlapping FoVs. Because dynamic objects in Sequence 4 appear in the FoV of Camera 5, cameras were added incrementally from Camera 5 to improve comparability. All configurations completed Sequence 2 successfully. In Sequences 4 and 5, however, the single-camera configuration did not provide stable tracking. It succeeded in only 1 of 10 trials on Sequence 4 and 3 of 10 trials on Sequence 5. Notably, when the number of cameras was increased to three, a 100% success rate was attained across all test sequences. The ATE also improved gradually as the number of cameras increased. Scale error remained unstable because nonoverlapping configurations provide no stereo information to recover metric scale. For deployments with purely nonoverlapping FoVs, future work could fuse IMU measurements, wheel odometry, or other metric sensors to provide an additional scale constraint.

We also evaluated DAMC-SLAM under two configurations with overlapping FoVs: Cameras (0, 1, 5) and all cameras. Adjacent cameras were configured with overlapping FoVs to provide stereo information. Table V presents the results. The system achieved a 100% tracking success rate for every configuration and test sequence, demonstrating its robustness. In terms of the ATE metric, the system performance gradually improved with the increase in the number of cameras, and the optimal performance was

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use , available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

Table IV. *Evaluation of different camera configurations with nonoverlapping FoVs.*

Camera configuration		Sequence 2	Sequence 4	Sequence 5
DAMC-SLAM-5	ATE	1.161	11.318	1.705
	Scale error	17.897	33.620	29.997
	TSR	10/10	1/10	3/10
DAMC-SLAM-1,5	ATE	0.824	0.247	0.387
	Scale error	0.082	3.531	1.119
	TSR	10/10	6/10	10/10
DAMC-SLAM-1,3,5	ATE	**0.695**	**0.150**	**0.334**
	Scale error	3.953	2.697	0.901
	TSR	**10/10**	**10/10**	**10/10**

Table V. *Evaluation of different camera configurations with overlapping FoVs.*

Camera configuration		Sequence 2	Sequence 4	Sequence 5
DAMC-SLAM-0,1,5	ATE	0.735	0.331	0.469
	Scale error	0.231	0.261	0.516
	TSR	10/10	10/10	10/10
DAMC-SLAM-All	ATE	**0.400**	**0.174**	**0.406**
	Scale error	**0.044**	**0.087**	**0.473**
	TSR	10/10	10/10	10/10

achieved when all cameras were employed. Owing to the configured overlapping FoVs between cameras, the system was provided with available stereo information. It can be observed from the scale error metric that the scale error decreased progressively as the number of overlapping FoV increased. This phenomenon confirms that the system can effectively utilize stereo information to recover metric scale.

Overall, DAMC-SLAM can operate with arbitrary multi-camera configurations. This capability results from optimizations throughout the system. The hybrid initialization module provides reliable initialization, whereas the tracking, local mapping, and loop closure modules exploit multi-camera information while filtering interference. As a result, the system exhibits a tendency where its performance metrics gradually improve with the increase of input information, rather than diverging due to the introduction of interference.

5.5. Filtering and retention of potential dynamic features

In the motion-model tracking module, map points are projected using the pose predicted by the constant-velocity model for feature matching. When only a few feature points lie on dynamic objects, their interference with pose estimation is limited. The resulting accurate pose estimate further reduces matches to dynamic features in subsequent frames. Figure 11 compares results without and with potential dynamic feature filtering. In the first row, panels (a)–(c) illustrate the increasing influence of dynamic objects as their image coverage increases. Panels (a) and (b) show that the system is inherently robust to small dynamic objects. In contrast, the large dynamic object in panel (c), which occupies more than one-sixth of the image, still affects the system. The second row shows the results after filtering. Panels (d) and (e) show that the threshold-based dynamic-feature filtering strategy remains robust to small dynamic objects, whereas panel (f) demonstrates effective suppression of the interference caused by the large dynamic object.

Figure 12 further illustrates the filtering and retention of potential dynamic features under different dynamic-object distributions. Panels (b)–(e) show that the strategy effectively retains features on stationary objects classified as potentially dynamic across different spatial distributions. It can also retain

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093



Figure 11. *Illustration of potential dynamic features filtering. The first row shows feature points before filtering, and the second row shows feature points after filtering. Green features serve as the features for successful tracking.*

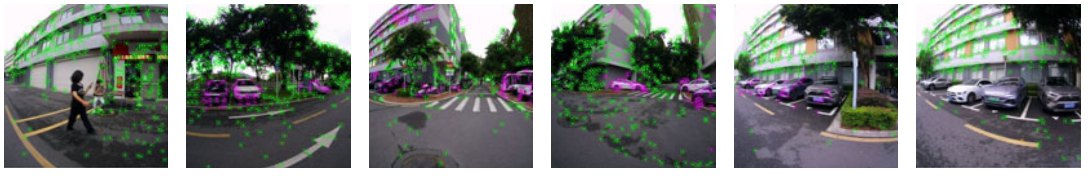


Figure 12. *Examples of filtering and retaining potential dynamic features under different dynamic-object distributions. Pink points indicate successfully tracked features on stationary objects classified as potentially dynamic, whereas green points indicate successfully tracked features on static background regions.*

such features when the segmentation model produces erroneous labels, as demonstrated by the building features in panel (e). However, as shown in panel (f), features are also filtered when the semantic masks associated with stationary objects classified as potentially dynamic occupy a large portion of the image. This limitation warrants further improvement, but it does not affect pose estimation when sufficient other features are available for pose optimization.

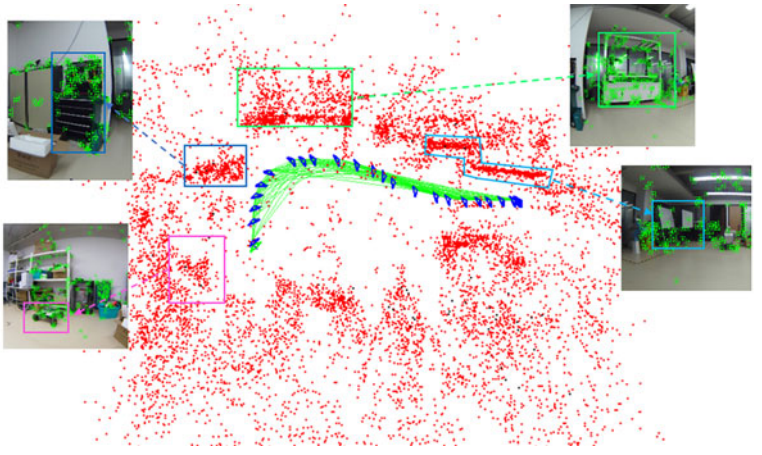
5.6. Run time performance

Table VI presents the average processing time per multi-frame for different camera configurations. The experiments used Sequence 2 and included the Camera (0, 1) configuration to measure runtime with the

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use , available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

Table VI. *The average time consumed per multiframe at different camera configurations.*

	Camera configurations	Time(s)
Non-overlapping	DAMC-SLAM-5	0.059
	DAMC-SLAM-1,5	0.112
	DAMC-SLAM-1,3,5	0.167
Overlapping	DAMC-SLAM-0,1	0.077
	DAMC-SLAM-0,1,5	0.117
	DAMC-SLAM-All	0.199



(a) Map created indoors by DAMC-SLAM



(b) One multi-frame from DAMC-SLAM

Figure 13. *Indoor performance on sequence 6.*

minimum stereo information. Under nonoverlapping FoVs, DAMC-SLAM achieves a maximum speed of 17 fps, which decreases to 6 fps with three cameras. Under overlapping FoVs, the maximum speed is 13 fps and decreases to 5 fps when all six cameras are used. Considering that the system needs to handle a $360^\circ \times 180^\circ$ stereo FoV, this processing speed is acceptable. The main bottlenecks are CNN-based preprocessing and local-map tracking. Under the DAMC-SLAM-All configuration, frame processing and local-map tracking require approximately 85 and 92 ms, respectively, accounting for approximately 89% of the total processing time reported in Table VI. Within local-map tracking, local-point matching requires approximately 67 ms and is the dominant cost, whereas pose-only bundle adjustment requires approximately 17 ms. Future improvements could be achieved by deploying a lighter or quantized CNN and by reducing the local-point search workload through adaptive selection of keyframes and map points.

5.7. Indoor performance

DAMC-SLAM also operates indoors. Sequence 6 was collected in a room using the multi-camera rig. Figure 13(b) shows one multi-frame, and Figure 13(a) shows the resulting map. The map appears

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use , available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

cluttered because the system extracts features from the ceiling and floor. We therefore show a map produced shortly after initialization for clarity. Prominent objects from which map points are extracted are labeled in Figure 13(a), and their locations are shown in Figure 13(b). The result demonstrates that DAMC-SLAM captures indoor spatial features and generates an accurate trajectory and map.

6. Conclusion

To enhance robustness under extreme conditions, this paper proposes DAMC-SLAM, an adaptive system for arbitrary multi-camera rigs. The system uses a CNN to extract feature points, their descriptors, and global descriptors for input images. A second CNN generates potential dynamic object masks online. A threshold-based mechanism then reduces the adverse effects of large-scale dynamic objects. The proposed hybrid initialization method enables map initialization for arbitrary multi-camera configurations, irrespective of whether their FoVs overlap. DAMC-SLAM also includes a dedicated multi-camera relocalization approach that fully exploits multi-camera information to re-establish poses after tracking failure. In local mapping, the system continuously integrates triple-observation map points (endowed with stereo information) into the map. This strategy recovers metric scale and mitigates scale drift. For loop closure detection, global descriptors of multi-keyframes replace the traditional BoW method when selecting candidate multi-keyframes.

We evaluated DAMC-SLAM on six sequences captured with a multi-camera rig in challenging environments, including dense vegetation, low light, illumination changes, and dynamic-object occlusions. Compared with ORB-SLAM3, HFNet-SLAM, and MultiCo-SLAM, DAMC-SLAM achieves the highest tracking success rate and the lowest ATE on this dataset. When configured with overlapping FoVs, it also recovers metric scale from stereo information and reduces scale drift. The study of different camera configurations further shows that performance generally improves as the number of cameras and overlapping FoVs increases. Not only outdoors, but the system is also effective in indoor environments. In addition, experiments on the EuRoC MAV benchmark further demonstrate competitive performance: despite not using an image pyramid, DAMC-SLAM attains the lowest average RMSE among the compared methods. Owing to its flexible adaptability, this system can be seamlessly integrated into a diverse range of mobile robot platforms, thereby providing the robots with high-precision localization and mapping functionalities.

Beyond localization and mapping, DAMC-SLAM can provide metric pose estimates and a reconstructed map to obstacle-perception and motion-planning modules in a closed-loop navigation framework. These modules can use the outputs to generate collision-free trajectories, while subsequent multi-camera observations update the robot state. The wide FoV, robustness to occlusion, and relocalization capability help maintain state estimation when individual views are degraded. Integrating DAMC-SLAM with obstacle-avoidance planning and tracking-control methods, such as those in refs. [40, 41], is a promising direction for future autonomous deployment.

Author contributions. Conceptualization, **Kang Li;** methodology**, Kang Li;** software, **Kang Li and Xuan An;** validation, **Kang Li, Xuan An and Dewei Han;** formal analysis, **Kang Li and Dewei Han;** investigation, **Xuan An;** resources, **Zhaoyuan Ma;** data curation, **Junjie Li and Jian Xu;** writing—original draft preparation, **Kang Li;** writing—review and editing, **Dewei Han, Junjie Li and Jian Xu;** visualization, **Kang Li and Xuan An;** supervision, **Zhaoyuan Ma;** project administration, **Zhaoyuan Ma;** funding acquisition, **Zhaoyuan Ma**. All authors have read and agreed to the published version of the manuscript.

Financial support. This research was funded by the 2022 Shenzhen 'Pengcheng Peacock Plan' for the Special Talent A-class funding, i.e., Ma Zhaoyuan Research Start-up Fund, grant number Y01966114.

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

References

[1] J. Boal, A. Sánchez-Miralles and A. Arranz, "Topological simultaneous localization and mapping: A survey," *Robotica* **32**(5), 803–821 (2014).

[2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.* **13** (2), 99–110 (2006).

[3] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.* **13**(3), 108–117 (2006).

[4] J. Cheng, L. Zhang, Q. Chen, X. Hu and J. Cai, "A review of visual SLAM methods for autonomous driving vehicles," *Eng. Appl. Artif. Intel.* **114**, 104992 (2022).

[5] A. J. Davison, I. D. Reid, N. D. Molton and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intel.* **29**(6), 1052–1067 (2007).

[6] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," **In:** *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality* (2007) pp. 1–10.

[7] J. Engel, T. Schöps and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," **In:** *Computer Vision – ECCV 2014*, vol. 8690 (2014) pp. 834–849.

[8] J. Engel, V. Koltun and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intel.* **40**(3), 611–625 (2018).

[9] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.* **33**(5), 1255–1262 (2017).

[10] J. Mo, M. J. Islam and J. Sattar, "Fast direct stereo visual SLAM," *IEEE Robot. Autom. Lett.* **7**(2), 778–785 (2022).

[11] C. Forster, M. Pizzoli and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry," **In:** *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014) pp. 15–22.

[12] M. Kaess and F. Dellaert, "Probabilistic structure matching for visual SLAM with a multi-camera rig," *Comput. Vis. Image Underst.* **114**(2), 286–296 (2010).

[13] K. M. Yi, E. Trulls, V. Lepetit and P. Fua, "LIFT: Learned Invariant Feature Transform," **In:** *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VI 14* (Springer, 2016) pp. 467–483.

[14] D. DeTone, T. Malisiewicz and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," **In:** *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2018) pp. 337–33712.

[15] L. Liu and J. M. Aitken, "HFNet-SLAM: An accurate and real-time monocular SLAM system with deep features," *Sensors* **23**(4), 2113 (2023).

[16] X. Zhang, H. Dong, H. Zhang, X. Zhu, S. Li and B. Deng, "A real-time, robust and versatile visual-SLAM framework based on deep learning networks," *IEEE Trans. Instrum. Meas.* **74**, 1–13 (2025).

[17] R. Mur-Artal, J. M. M. Montiel and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.* **31**(5), 1147–1163 (2015).

[18] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel and J. D. Tardos, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.* **37**(6), 1874–1890 (2021).

[19] J. Engel, J. Stückler and D. Cremers, "Large-Scale Direct SLAM with Stereo Cameras," **In:** *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2015) pp. 1935–1942.

[20] D. Wang, J. Wang, Y. Tian, K. Hu and M. Xu, "PAL-SLAM: A feature-based SLAM system for a panoramic annular lens," *Opt. Express* **30**(2), 1099 (2022).

[21] D. Caruso, J. Engel and D. Cremers, "Large-Scale Direct Slam for Omnidirectional Cameras," **In:** *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2015) pp. 141–148.

[22] R. Castle, G. Klein and D. W. Murray, "Video-Rate Localization in Multiple Maps for Wearable Augmented Reality," **In:** *2008 12th IEEE International Symposium on Wearable Computers* (2008) pp. 15–22.

[23] A. Harmat, I. Sharf and M. Trentini, "Parallel Tracking and Mapping with Multiple Cameras on an Unmanned Aerial Vehicle," **In:** *Intelligent Robotics and Applications* (2012) pp. 421–432.

[24] S. Urban and S. Hinz, Multicol-SLAM - A modular real-time multi-camera slam system (2016).

[25] H. Seok and J. Lim, "ROVO: Robust Omnidirectional Visual Odometry for Wide-Baseline Wide-FOV Camera Systems," **In:** *2019 International Conference on Robotics and Automation (ICRA)* (2019).

[26] A. Li, D. Zou and W. Yu, "Robust Initialization of Multi-Camera SLAM with Limited View Overlaps and Inaccurate Extrinsic Calibration," **In:** *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2021) pp. 3361–3367.

[27] J. Kuo, M. Muglikar, Z. Zhang and D. Scaramuzza, "Redesigning SLAM for Arbitrary Multi-Camera Systems," **In:** *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020) pp. 2116–2122.

[28] J.-P. Tardif, Y. Pavlidis and K. Daniilidis, "Monocular Visual Odometry in Urban Environments Using an Omnidirectional Camera," **In:** *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2008) pp. 2531–2538.

[29] Y. Yang, D. Tang, D. Wang, W. Song, J. Wang and M. Fu, "Multi-camera visual SLAM for off-road navigation," *Robot. Auton. Syst.* **128**, 103505 (2020).

[30] P. Kaveti, S. N. Vaidyanathan, A. Thamilchelvan and H. Singh, "Design and evaluation of a generic visual SLAM framework for multi-camera systems," *IEEE Robot. Autom. Lett.* **8**(11), 7368–7375 (2023).

[31] J. Liu and H. Deng, "Improve generalization for neural visual-SLAM with Bayes online learning," *Robotica* **43**(5), 1660–1674 (2025).

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use , available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093

[32] K. Zhang, C. Dong, H. Guo, Q. Ye, L. Gao, S. Xiang, X. Chen and Y. Wu, "A semantic visual SLAM based on improved mask R-CNN in dynamic environment," *Robotica* **42**(10), 3570–3591 (2024).
[33] P.-E. Sarlin, C. Cadena, R. Siegwart and M. Dymczyk, "From Coarse to Fine: Robust Hierarchical Localization at Large Scale," **In:** *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019) pp. 12708–12717.
[34] J. Tang, L. Ericson, J. Folkesson and P. Jensfelt, "GCNv2: Efficient correspondence prediction for real-time SLAM," *IEEE Robot. Autom. Lett.* **4**(4), 3505–3512 (2019).
[35] H. M. S. Bruno and E. L. Colombini, "LIFT-SLAM: A deep-learning feature-based monocular visual SLAM method," *Neurocomputing* **455**, 97–110 (2021).
[36] D. Li, X. Shi, Q. Long, S. Liu, W. Yang, F. Wang, Q. Wei and F. Qiao, "DXSLAM: A Robust and Efficient Visual SLAM System with Deep Features," **In:** *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2020) pp. 4958–4965.
[37] J. Kannala and S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Trans. Pattern Anal. Mach. Intel.* **28**(8), 1335–1340 (2006).
[38] E. Zhang, "fastseg,". https://github.com/ekzhang/fastseg. accessed: 2025-09-18 (2020).
[39] L. Kneip, P. Furgale and R. Siegwart, "Using Multi-Camera Systems in Robotics: Efficient Solutions to the NPNP Problem," **In:** *2013 IEEE International Conference on Robotics and Automation* (2013) pp. 3770–3776.
[40] Q. Wu, X. Shen, Y. Jin, Z. Chen, S. Li, A. H. Khan and D. Chen, "Intelligent beetle antennae search for UAV sensing and avoidance of obstacles," *Sensors* **19**(8), 1758 (2019).
[41] A. H. Khan, S. Li, D. Chen and L. Liao, "Tracking control of redundant mobile manipulator: An RNN based metaheuristic approach," *Neurocomputing* **400**, 272–284 (2020).

Cite this article: K. Li, X. An, D. Han, J. Xu, J. Li and Z. Ma, "Design and evaluation of deep learning-based arbitrary multiple camera simultaneous localization and mapping system", Robotica. https://doi.org/10.1017/S0263574726104093

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 11 Nov 2026 at 20:29:59, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726104093