IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 April 2023, accepted 25 April 2023, date of publication 2 May 2023, date of current version 10 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3272479

TOPICAL REVIEW

A Review of Recurrent Neural Network Based Camera Localization for Indoor Environments

MUHAMMAD SHAMSUL ALAM[1,4], (Graduate Student Member, IEEE), FARHAN BIN MOHAMED[1,3], (Senior Member, IEEE), ALI SELAMAT[2,3], (Member, IEEE), AND AKM BELLAL HOSSAIN[1,4]

[1]Department of Emergent Computing, School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, Johor 81310, Malaysia
[2]Malaysia–Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia, Kuala Lumpur 54100, Malaysia
[3]Media and Game Innovation Centre of Excellence (MaGICX), Universiti Teknologi Malaysia, Johor Bahru, Johor 81310, Malaysia
[4]Department of Computer Science, Faculty of Science and Arts at Belqarn and Al Namas, University of Bisha, Bisha 61985, Saudi Arabia

Corresponding authors: Muhammad Shamsul Alam (shamsul20@graduate.utm.my) and Farhan Bin Mohamed (farhan@utm.my)

ABSTRACT Camera localization involves the estimation of the camera pose of an image from a random scene. We used a single image or sequence of images or videos as the input. The output depends on the representation of the scene and method used. Several computer vision applications, such as robot navigation and safety inspection, can benefit from camera localization. Camera localization is used to determine the position of an object on the camera in an image containing multiple images in a sequence. Structure-based localization techniques have achieved considerable success owing to a combination of image matching and coordinate regression. Absolute and relative pose regression techniques can provide end-to-end learning; however, they exhibit poor accuracies. Despite the rapid growth in computer vision, there has been no thorough review of the categorization, evaluation, and synthesis of structures and regression-based techniques. Input format and loss strategies for recurrent neural networks (RNN) have not been adequately described in the literature. The main topic is indoor camera pose regression, which is a part of the camera localization techniques. First, we discuss certain application areas for camera localization. We then discuss different camera localization techniques, such as feature and structure-based, absolute and relative pose regression techniques, and simultaneous localization and mapping (SLAM). We evaluated the frequently used datasets and qualitatively compared the absolute and relative camera pose estimation approaches. Finally, we discuss potential directions for future research, such as optimizing the computational cost of the features and evaluating the end-to-end characteristics of multiple cameras.

INDEX TERMS Camera pose regression, absolute pose regression, indoor positioning, camera localization, robot navigation, SLAM.

I. INTRODUCTION

Camera localization is a critical problem in robotics and computer vision. Camera localization is necessary for many applications such as mobile robot navigation and safety inspection. Various localization procedures have been developed, owing to the significance of these issues. By applying image descriptors to 3D scene point clouds using structure-from-motion (SfM), point-based localization techniques can identify correlations between the local features retrieved from an image and image descriptors, as shown in Figure 1. This collection of 2D-3D matches allows the camera pose to be determined. However, this low-level matching procedure does not provide reliable and accurate results in certain situations such as motion blur, fewer textured surfaces, considerable changes in lighting, occlusions, and repeated structures. Various machine-learning algorithms have been successfully applied to camera localization problems, including scene coordinate regression forests (SCoRF). SCoRF generates an initial set of camera pose hypotheses based on the projected 3D location of four pixels in an input image, and is then refined using a random sample consensus (RANSAC)

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

loop. Although these algorithms are helpful, researchers must match the input images with depth maps during time-limited training.

The development of large-scale indoor localization [1], [2] systems has become critical in computer vision research over several years. Accurate pose data are crucial for various applications including indoor robot navigation [3]. Although the sensors used for localization vary depending on the requirements and equipment, cameras are the most popular because of their low cost, ease of integration, and high output quality. Cameras are useful in various applications including large-scale smartphone-based localization in indoor environments. Several schemes have been proposed for localizing cameras. However, its performance is poor when it is used for large-scale indoor pose estimation. This characteristic is an uneven motion blur, which is common in indoor environments [4]. Applications, such as localizing historical scenes and estimating the starting position of robots, require single-image localization methods. In many applications such as indoor mobile robot, cameras capture image sequences rather than recording individual images. Substantial research has been conducted on image sequences for visual localization [5].

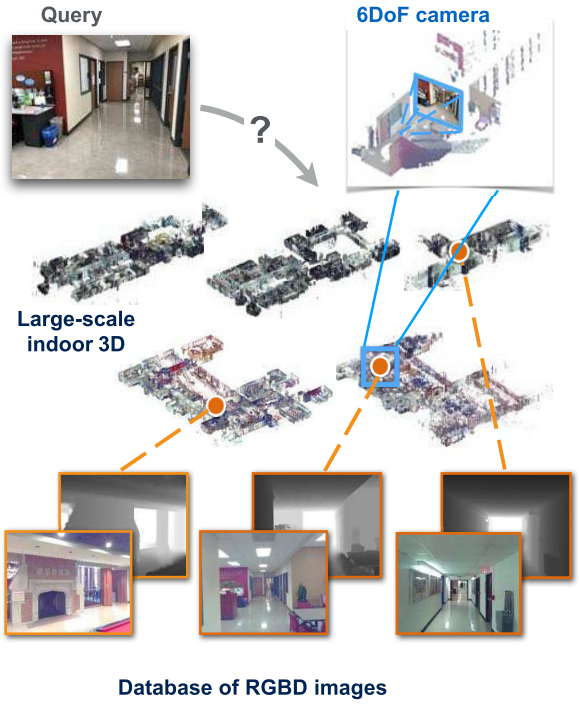

FIGURE 1. Large scale indoor camera localization from [6].

Mobile robot navigation [7], [8], and image-based indoor positioning are essential aspects that have recently attracted significant interest from academia and industry. A comprehensive overview of indoor camera localization is necessary. Determining camera localization is required to enhance augmented reality. Therefore, camera localization has a wide range of applications. Image-based camera localization is the most commonly used feature. Camera localization using images is a broad research field. We strive to include relevant studies and comprehensively categorize image-based camera-localization methods. The indoor camera localization system inputs can come from stationary or moving cameras and can be a single image, sequence of images, or videos. Finally, the system performs camera localization. However, structure-based localization techniques can yield earlier results. Structure-based and regression-based localizations are the two main types of camera localizations. This essay analyzes the quantitative and qualitative findings and suggests directions for further investigation.

A. APPLICATION AREAS

To the best of our knowledge, there has been no comprehensive analysis of camera localization applications. Augmented reality (AR) [9] is widely used for image generation. Using three-dimensional (3D) virtual objects outside the physical world can enhance real-time images and create a perfect connection between virtual and real images. Potential applications include education, gaming, and medical and military trainings. It is essential to obtain six degrees-of-freedom for the camera, which requires accurate positioning. Typically, a mobile camera is used for AR-related technologies that are used indoors because it prevents human micro-movements and limb movements when compared with a person-moving mounted camera. Human activities and indoor camera localization may improve the augmented reality game experience and have several uses, such as three-dimensional reconstruction. A location module commonly obtains information from light detection and ranging (LiDAR) [10], and video sensors are required to locate self-driving cars. Robots and vehicles require in-depth and accurate positional knowledge for sensing, mapping, scheduling, and managing. Robotic or automotive devices measure the present pose regarding the starting pose using sensor data, and then employ a matching algorithm and navigation to position themselves and navigate. Because outdoor items may be larger than those found indoors, the localization accuracy for navigation may be worse. Real-time localization may be crucial for autonomous driving. Robotic applications that require visual input from cameras include object localization and help with conventional routine assessments for performance assurance and security inspections. Robot walking involves adaptively controlled trajectory. Visual information can follow pathways, identify barriers, and recognize signs in the surrounding area in order to find a robot and avoid obstacles.

B. RESEARCH GAP

Reviews of camera localization techniques that have already been published do not specifically compare their approaches to deep neural networks and recurrent neural networks in terms of performance and input types, such as single images, image sequences, and video streams. In addition, some pose estimation techniques are based on image matching or retrieval. In this study, we extracted the image

closest to the reference image or obtained the matching correspondences from a sequence of images. We also included regression- based localization, which uses an image regression technique to regress the camera pose. This study categorized image-matching methods for estimating camera poses as structure-based pose estimation. Regression-based and structure-based approaches to recurrent deep neural network analysis, including performance analysis, are also included in our study.

C. PAPER ORGANIZATION

In Section I, we describe the most common camera localization applications. In Section II, we introduce the camera localization. Related studies on indoor camera localization are presented in Section III. The recurrent neural network (RNN) architecture is discussed in Section IV. Indoor camera localization techniques are examined in detail in Section V. Several techniques for camera localization include point, depth, synthetic image, feature, regression-based, and SLAM methods. We explain the subtopics in Section V based on a single image, an image sequence, and a video stream. It includes a qualitative comparison of the absolute and relative pose regression in Section V, and summarizes the performance of the camera localization technique. We discuss the previously used benchmark and popular datasets in Section VI. In Section VII, we discuss the constraints on the indoor camera localization. In Section VIII, we discuss future research directions, and conclude the paper.

II. PROBLEM IDENTIFICATION

We plan to understand the concept of camera localization. We subjected an image to a depth camera to obtain the coordinates in 3D space and determine camera localization. The camera localization problem determines the pose in a known environment that matches a query image that fits the model with dataset images. Monocular cameras [11] and depth cameras that combine RGB and RGB-D images are used for the camera pose estimation problem. Each pose p includes the camera location and rotation. We can represent the changes in the position and orientation. A 3×3 rotation matrix, four-digit quaternion, and Euler angles can be translated into each other to express the rotation. We independently selected the position and orientation representations using three-digit 3D coordinates x and a three-digit normalized quaternion q. Thus, translation and processing could be used to define the ground truth p and estimate the pose vectors p.

$$p = (x, q) \text{ and } \hat{p} = (\hat{x}, \hat{q}) \tag{1}$$

The evaluation procedures for the localization tasks also varied according to the localization process. To obtain a more accurate estimation result that matches the ground-truth result, when assessing the efficiency of camera pose estimation, we must contrast the pose determined by the estimating technique with the actual pose. The standard method for calculating the pose is structure-from-motion (SfM), because the camera position is associated with the coordinates of the 3D model. Figure 2 shows a schematic of an indoor positioning [12], [13], [14], [15] application.

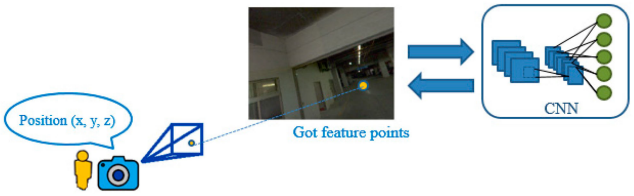

FIGURE 2. Sketch map of indoor positioning application [12].

Absolute and relative pose errors are two standard measurement measures for camera-localization techniques. For datasets that directly offer ground-truth poses, the difference between the estimated and ground-truth poses is used to evaluate the pose accuracy of the method. Absolute pose error is a useful tool for assessing the performance of simultaneous localization and mapping (SLAM) [16] systems. The relative pose error helps to determine the extent to which the optical odometry system deviates. When a single image was used as the algorithm input, the absolute pose error was calculated by adding the ultimate inaccuracies of the position and orientation. The Euclidean distance between the calculated and real-world positions was used to calculate the position error. The absolute orientation error, an angle in degrees, represents the minimum rotational angle required for the estimated and actual orientation. The traces of the existing and estimated rotation matrices, which can be alternatively described using the real and estimated quaternions, can be used to compute the backward rotation error. A sequence of time-stamped images was inputted into the algorithm. The relative pose error was calculated by combining the relative position and orientation errors with the absolute pose error. We calculated the position error using the Euclidean distance between the calculated close location and the position relative to the ground truth. The lowest angle deviation rate in degrees between the estimated relative quaternion and the actual relative orientation is the computation of the orientation inaccuracy when using a quaternion representation. Statistical data metrics are frequently used to report position and rotation errors.

III. RELATED WORKS

Indoor Camera localization has achieved tremendous success over the last decade and several studies have been published on this topic. There are several subcategories of cameras localization. In this paper, we discuss several studies in the following subcategories. Figure 3 shows a picture of the most influential published papers in the past five years that we have included in our study. The highest number of published papers on point-based and image-based localization in 2020 was ten, while the lowest number was seven in 2018. In feature-based localization, ten published articles were from 2019, and seven were from 2021. For regression-based, there were 18 published papers from 2022 and thirteen from 2020. In the SLAM, the highest number of articles were thirteen from 2022, and the lowest was nine from 2020 As

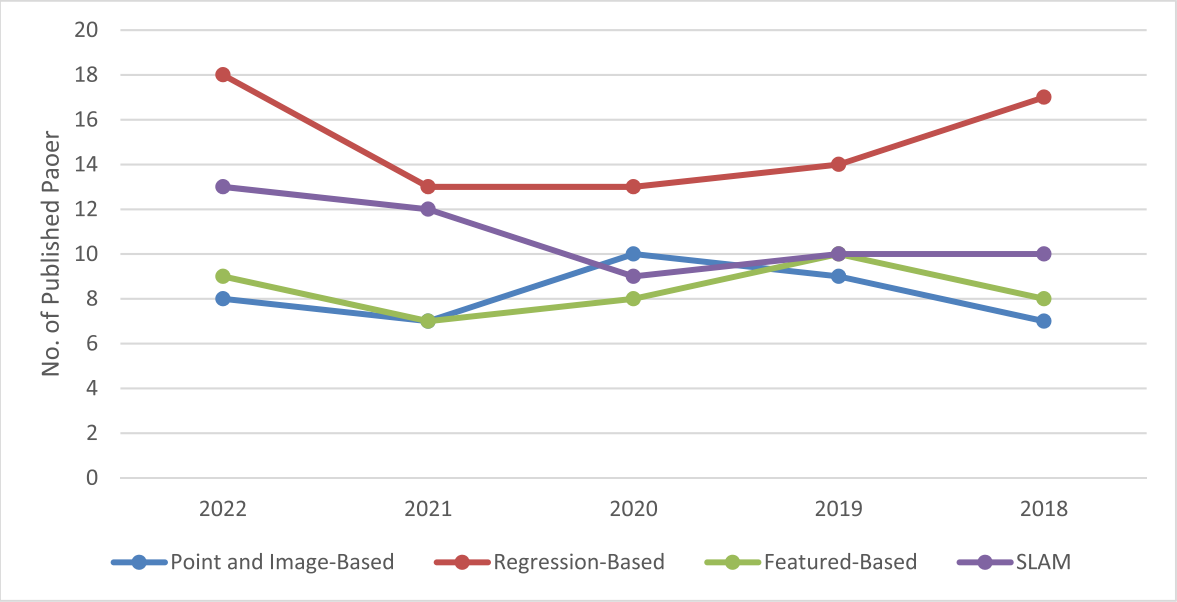

FIGURE 3. Last five years research contribution of indoor camera localization.

shown in Figure 3, regression-based research has recently become comparatively more popular than structure-based methods, which inspired us to study RNN-based camera-pose regression.

Based on simultaneous localization and mapping (SLAM) [17], the authors reviewed indoor camera localization techniques based on simultaneous localization and mapping. Based on the SLAM model [18], the localization process is classified into known and unknown environments, and real-time or offline mapping procedures. Structure-based and retrieval-based pipelines were empirically contrasted by [19], who mathematically modelled an entire pose regression system. This review concludes that, despite the performance gap between them, it might improve the absolute pose regression approaches to compete with relative pose regression. In [20], the authors described research on SLAM focusing on fundamental localization techniques based on deep learning. Researchers have described camera-pose estimation techniques that require an image input, an algorithm pipeline, and broad ideas to increase the efficiency of such processes. In [21], different observed data were used in direct and indirect image-positioning systems, particularly the influence of features on location under changes in the illumination. This is achieved by classifying diverse and extensive input data into dimensional, contextual, and merged types. Reviews that have already been written frequently concentrate on excellent SLAM systems or specific indoor camera-localization methods. Current camera pose estimation methods do not compare their approaches with deep neural networks in terms of datasets, loss functions, and input formats. Some pose-estimation techniques focus on image matching and retrieval. Researchers have assessed and categorized image-matching methods for camera-pose estimation to address the lack of a description of such matching.

TABLE 1. Summary of most related reviews of indoor camera localization.

Reference	Main Coverage	Research Type
[22]	From sensor networks without cameras to smart cameras in 3D environments	Review
[23]	Introduce an approach to classifying methods and using video as a localization sensor in multi-sensor systems.	Review
[18]	To provide a complete set of classifications for image-based camera localization.	Review
[21]	Incorporating contemporary techniques into an identified visual capture environment.	Survey
[24]	Explain the idea, the evolution of the approaches, and the benefits and drawbacks	Review
[17]	A complete investigation of visual-LiDAR SLAM	Review
[25]	Describe the assessment criteria for camera localization, as well as specific applications.	Critical Analysis
[26]	Define the SLAM characteristics based on visual odometry and perceptual methods.	Review
[27]	Solutions for indoor positioning are proposed based on novel classifications, sensor data, kind of detected items, and localization technique.	Survey
[28]	Various indoor and outdoor positioning techniques and methods, with a focus on indoor techniques and ideas	Survey

This study examines regression and structure-based strategies for recurrent neural network research, including loss

function analysis. This study also demonstrates the effectiveness of localization-emphasis formulations for various models. Table 1 summarizes the significant reviews on indoor camera localization.

The ability to simultaneously generate a map and estimate an indoor camera's pose in an unknown environment, simultaneous localization and mapping (SLAM) algorithms are the subjects of extensive research. At first monocular SLAM (MonoSLAM) [29] system proposed in 2007, and visual SLAM (V-SLAM) quickly became famous as a research area. The Semi-direct Visual Odometry (SVO) [30] technique is quick and doesn't require calculating several descriptors, uses semi-direct visual odometry. Because this method does not require feature extraction for every frame [31], it may operate with high frame rate. It can therefore used in low-cost embedded systems, such as the embedded platform under consideration [30]. Because of the correlation of short-term data, the SVO's accuracy is limited [32]. SVO does not use loop closure or global optimization strategies. A direct approach that provides a semi-dense reconstruction is the large-scale direct monocular SLAM (LSD-SLAM) [33]. The LSD-SLAM map estimate algorithm, which has a lesser accuracy than others like PTAM and ORB-SLAM [31], is primarily based on pose-graph optimization [32].

The main component of the V-SLAM is the Oriented FAST and Rotated BRIEF (ORB-SLAM) introduced by Mur-Artal [31]. It is a well-known system that employs the feature point technique. Scale drift was an issue with the monocular camera used as the visual sensor in ORB-SLAM [31]. The first SLAM system with monocular, stereo, and RGB-D cameras, ORB-SLAM2 [34], was developed in 2016 in response to ORB drawbacks. The most advanced feature-based method is the ORB-SLAM2 [34] method, built on the ORB-SLAM algorithm [31]. Tracking, local mapping, and loop closing are three concurrent threads it uses to function. Yu et al. [35] and Abouzahir et al. [36] deployed the ORB-SLAM method on several CPU and GPU-based systems. The direct sparse odometry (DSO) technique [37] combines a straightforward strategy with a sparse reconstruction. The DSO algorithm takes into account a recent frame window. It carried continuous optimization by using a local bundle adjustment to optimize both the inverse depth map and the keyframes window.

IV. RNN TECHNIQUES OVERVIEW

A recurrent neural network (RNN) is a network-based memory space and loop that deals with sequential data. The RNN architecture is at the heart of long short-term memory (LSTM). Figure 4 shows the simple RNN architecture. At each time iteration t the hidden state h_t is:

$$h_t = \sigma_h(W_{xh}h_t + W_{hh}h_{t-1} + b_h) \tag{2}$$

where σ_h is the activation function, W_{xh} is the weight matrix between the input and hidden layers, W_{hh} is the weight matrix between the two hidden layers, and b_h is the bias vector of the hidden layer. The network output y_t is

$$y_t = \sigma_y(W_{hy}h_t + b_y) \tag{3}$$

where σ_y is the output layer activation function, W_{hy} is the weight matrix between the hidden layer and the output, b_y is the bias vector of the output layer.

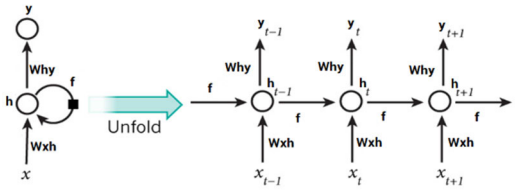

FIGURE 4. The architecture of simple RNN [38].

A. LONG-SHORT TERM MEMORY (LSTM)

The LSTM is an RNN that prevents gradients from disappearing. The LSTM uses a technique known as a gate. Can learn long-term dependencies gates indicate whether data in a sequence should be retained or discarded. The three gates of LSTM are the input, forget, and output. Several advanced recurrent architectures, including LSTM [39] and gated recurrent units (GRU), have addressed the aforementioned RNN. LSTMs effectively solve sequence-based problems with long-term constraints, whereas GRU, a much simpler LSTM architecture, was recently developed and implemented in machine learning. The control flow of an LSTM is similar to that of a recurrent neural network. As it travels, it receives input and relay information. The mechanisms that occur within the LSTM cells differ. The first gate of LSTM was forgotten. This procedure determines whether data are retained or discarded. The sigmoid function transports the data from the previous hidden layer and the current input data. Next, we examine the output gates. The output gate determines the hidden gate's concealed state. It is important to remember that the hidden state includes information from previous inputs. The hidden state was also used to make the predictions. Figure 5 shows the LSTM architecture.

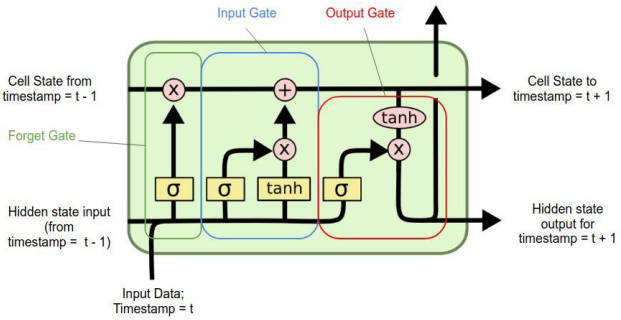

FIGURE 5. The architecture of the LSTM [38].

B. BI-DIRECTIONAL LSTM (BI-LSTM)

Bidirectional LSTMs are LSTM models that use existing data and the future of a single time step as inputs. At each moment, we can preserve knowledge from both the past

and future in Bi-LSTM. A bidirectional RNN [40] is a bi-LSTM concept that analyzes sequence inputs in the front and rear directions by using two hidden layers. Bi-LSTMs connect the two hidden layers to a virtually identical output layer. Bidirectional long-term memory (Bi-LSTM) is an approach for storing sequence information from forward and backward directions in a neural network. Bi-LSTM is a sequential processing system for two LSTMs processing the input ahead and the other processing it backwards. Bidirectional LSTMs (Bi-LSTMs) are LSTM systems that incorporate input data from a single time step in the past and future. In Bi-LSTMs, information can be stored both in the past and in the end. Bi-LSTM [41] is commonly used for activities requiring sequencing. Subsequently, we built a bidirectional long short-term memory (LSTM) network. The conventional LSTM update equations simultaneously compute the forward- and backward-level outputs. Figure 6 shows the Bi-LSTM architecture.

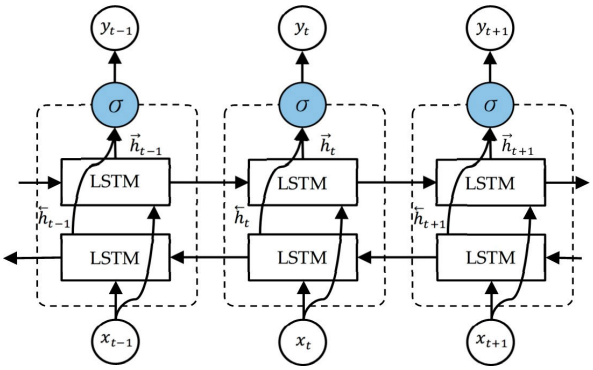

FIGURE 6. The architecture of the Bi-LSTM [42].

C. GATED RECURRENT UNIT (GRU)

The GRU is a relatively new recurrent neural network that follows LSTM. GRUs [39], [43] reject cell states in favor of data transfer via hidden states. There are two gates: one for resetting, and the other for updating. The update gate operated as an LSTM network gate. Consequently, input gates operate accordingly. It determines which data should be deleted and which should be re-entered. The reset gate is another gate that is used to determine the amount of past information that should be lost. GRUs are faster than LSTMs, because they contain fewer tensor operations. Figure 7 shows the GRU architecture.

V. OVERVIEW OF INDOOR CAMERA LOCALIZATION

A. POINT-BASED

The point-based indoor camera localization method estimates the camera pose directly using traditional photogrammetry strategies, such as Perspective-n-Point (PnP) [45]. The PnP problem is the camera pose of the domains comprising 3D space points n. When n is $3 \leq n < 6$, it is nonlinear PnP. When $n \geq 6$, the linear PnP [46] estimates the camera pose of an unknown environment by using real-time video and online data. This is known as simultaneous localization and

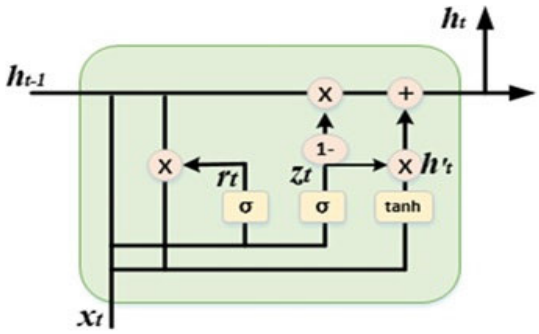

FIGURE 7. The architecture of the GRU [44].

mapping (SLAM) [47]. PnP problem recovers each feature of the 3D information through point clouds. The limitation of these methods is their dependence on point clouds through SfM, a three-dimensional (3D) reconstruction of the indoor environment. Camera localization depends on known and unknown territories. A particular case of PnP for n=3 is called Perspective-Three-Point (P3P). P3P refers to the minimum number of control points that produce a finite number of solutions [46]. In [48], the authors investigated a PnP problem with an uncertain focal length by using points and lines. They explored the PnP problem with an indefinite focus length by using radially distorted images [49]. In [50], a generalized pose-and-scale approach was proposed as the minimum solution. In [51], an angle restriction was incorporated, and for each P3P, it derived a compact bivariate polynomial equation. Subsequently, it presented a generic approach using iterations for a PnP problem with an uncertain focal length.

B. CNN-BASED

Convolutional neural networks (CNN), a deep-learning-based camera localization method [52], [53], perform convolution operations on RGB images to estimate the camera pose. The first attempt to use CNNs for direct camera pose regression was performed using PoseNet [54]. PoseNet computers with fully connected layers use GoogleNet as a framework for feature extraction [55], [56]. The proposed pose-regression architecture comprises three components: an encoder that creates the visual encoding vector, localizer that produces the localization feature vector, and regressor that regresses the pose. The encoder, localizer, and regressor are the three main components of the PoseNet architecture, as shown in Figure 8. In Bayesian-PoseNet [56], researchers introduced PoseNet to account for uncertainty in pose estimation The LSTM-PoseNet [57] architecture reduces dimensionality and improves localization accuracy. Pose regression is based on the hourglass architecture [58]. Figure 9 shows the LSTM-PoseNet architecture. Other studies have focused on frameworks for improving camera localization. They combined global poses with relative poses by predicting comparative poses from the image sequence [59]. This strategy focuses on the geometrically important features [60]. They achieved pose regression through multitasking learning, which combines information from the

associated activities. The VidLoc architecture uses CNN-RNN networks to constrain the network using the temporal smoothness of camera motion [61]. In [62], recurrent neural networks (RNNs) were incorporated into a nonlinear structure, dramatically improving performance.

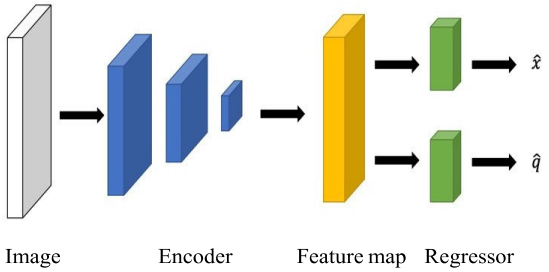

FIGURE 8. PoseNet architecture [54].

A deep learning-based system usually requires large annotated images to achieve high accuracy [63]. To overcome this challenge, a 3D model was used to generate synthetic images. To develop a map of benchmarks that approximates the difference between synthetic and original images in the pattern representations [64]. They evaluated feature search images against synthetic images in a database, using synthesized images to create a dataset of geolocation images [65]. Their research shows a deep architecture that uses synthetic images for training, and the recurrent neural network-based PoseNet directly estimates camera localization. BIM-PoseNet uses synthetic image sequences to estimate the camera pose to improve localization performance [66], [67]. This process reduces the localization performance by accounting for the range changes between synthetic and original images [68]. The localization performance degradation problem was solved using a domain-matching approach [69].

The domain adaptation process uses labelled data from different primary domains to achieve good results. Recently, deep learning-based models have been proposed to address the problem of domain matching between the input and output domains for object identification, segmentation, and classification [70]. These approaches aim to align feature mapping between the input and output domains. Recent studies have sought to increase the capacity of deep-learning models to match the environment [71]. Some current systems attempt domain matching in the pixel space [72]. Researchers have proposed methods for partitioning images into texture representations, and domain-specific and domain-invariant structures. Many studies have suggested changing synthetic images to narrow the domain gap and improve the efficacy of image translation strategies [73]. Several researchers have studied domain adaptation for semantic segmentation. A multilevel adversarial network exploits the structural similarities between source and target domains [74]. It introduces entropy loss to minimize poor trust predictions for the target domain [70]. A multilevel adversarial network [71], AdaptSegNet [75], was developed to exploit the structural similarities between source and target domains. Many current initiatives have addressed the domain-adaptation process, such as 2D-to-3D correspondence-based [75] and image-search-based localization [76].

C. DEPTH IMAGE-BASED

In image-based indoor camera localization, depth cameras assign the position of each pixel to an image [77]. An offline-trained regression forest was used to regress the position in a new indoor environment [78], [79], [80] using training examples to update the learned model dynamically. A scene coordinate regression forest (SCoRF) was trained to explicitly predict the correspondence between any image pixel and points in a scene's 3D world coordinate frame. Typical feature detection, description, and matching can be performed without forest support. RGB-D photos with a recognized camera pose trained using SCoRF in a specific scenario. The depth maps and indoor camera locations adequately compute the scene coordinate training labels for each pixel. A typical regression method uses these labels to learn the forests. SCoRF uses straightforward and rapidly added RGB and depth image pixel-comparison characteristics [77]. Changes in DSAC [81] and DSAC++ [82]. DSAC and DSAC++ were optimized and exhibited excellent accuracy. Depth image-based techniques use depth cameras to create depth maps of indoor environments.

D. SYNTHETIC IMAGE-BASED

Creating synthetic images improves camera pose regression by using a 3D model created from authentic images [83]. In coarse visual localization using images, researchers compared natural and artificial images based on features derived from a CNN using a similarity metric to compare natural and synthetic images based on the information extracted from a CNN [65]. Using known camera poses, researchers classified actual images based on their similarity to synthetic images. The BIM-PoseNet [66] model trains artificial images extracted from a 3D model to predict camera position and orientation from authentic images. By storing natural and synthetic images, they considered results of less than 2m. Synthetic images were then used to simulate the uncertainty of the pose estimation using Bayesian BIM-PoseNet [66]. However, compared with systems that use authentic images, the predicted camera poses of these techniques are less accurate. The requirement of 3D reconstruction using the SfM approach is a limitation [61]. However, this study had the problem of relying on depth cameras, which makes them unsuitable for most smartphone cameras [81]. The lack of precise localization and inability to interpolate the exact camera position hampers the results [65].

E. FEATURE-BASED

Pose estimation based on features and regression is the primary method used for camera localization. A structural feature-based camera pose estimation method recovers the camera position and orientation by establishing a correspondence between the test image features and structural

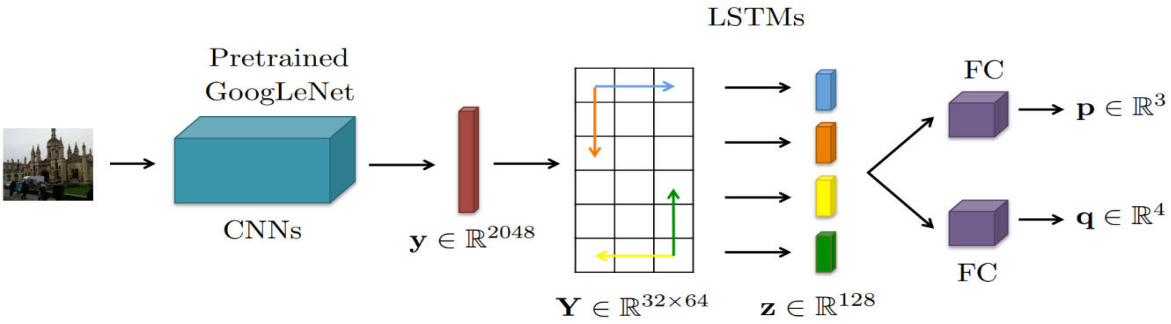

FIGURE 9. LSTM-PoseNet architecture [57].

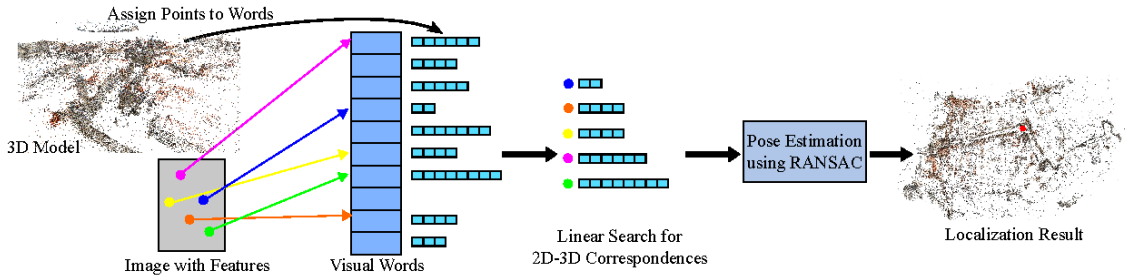

FIGURE 10. Direct 2D-to-3D matching [88].

features [84] of the 3D visual model using structure from motion (SFM) to create a 3D point cloud model (PCM). The feature-based method depends on various data regarding the 3D scene model compared with strategies that regress pose as a function of image object features. After establishing a correlation between the 3D point cloud model and query images and recovering the camera pose, the perspective-n-point (PnP) algorithm is traditionally used to localize the camera in random sample consensus (RANSAC) [85] loops.

1) MATCHING BASED

We used feature descriptors in matching-based localization approaches to establish a relationship between the object and query images. We usually assign each 3D point one or more visible local descriptors by the 3D scene model. We must extract consistent and distinct characteristics to build a connection between the images and environmental representations from a query image. Therefore, a localization process based on matching descriptors is transformed into a feature descriptor matching task. Usually, the distances between descriptors are compared to check features from the scene model. The image-based localization of large-scale cameras was previously considered to be difficult for location determination [86]. The image that was most similar to the database was used to find the searched image. However, the localization accuracy does not require applications that require precise camera positioning. Researchers are increasingly proposing and using three-dimensional (3D) scene models to predict poses and achieve higher accuracy [87]. Direct matching may seem straight forward, in that a match can be made by directly matching the 2D feature of the query image with the point of the 3D model. Figure 10 shows the direct 2D-3D matching architecture. The main problem with natural matching approaches is the identification of sufficiently high-quality correspondences to allow for fast and efficient pose estimation. Researchers have devoted considerable efforts to the development of matching algorithms [88]. However, 3D point matching and 2D image features do not indicate whether the order of a person's visual elements is superior to that of the others.

Considering all features in a search image is time consuming [89]. A supervised, trained random forest is generated by assigning information for multiple features to the same 3D point. In [90], a productive and practical pipeline is proposed. Quantitative feature descriptors were used to speed up the 2D-3D matching. It maintains the quantization loss in 3D-2D matching methods. Similar or repetitive feature points in a large-scale environment always lead to location determination errors. To overcome this difficulty, a more effective localization system using global contextual information is required [91], [92]. In [92], a reliable and tractable outlier suppression strategy is used to handle several outliers. Improving 3D localization using vertical coordinates [80]. Instead of simply enhancing the fitting accuracy, a new method was used, as described in [93].

To match 3D points in RGB-D images with the scene's point cloud model, The iterative closest point approach (ICP) is typically used when descriptors have with the scene's point-cloud model. Extract properties such as shape and density without directly accessing the spatial information. To construct a 3D-3D relationship by comparing descriptors [94], the hand creates low-level 3D geometric feature descriptions. 3D-ShapeNet uses 3D deep learning to create point cloud shapes [95]. In [96], a reconstruction technique

was proposed that combined local features within a narrow domain to produce a complete description while working with unresolved data hindered by various interference patterns and inequalities.

2) HIERARCHICAL MATCHING-BASED

Researchers have analyzed each 3D point for a query feature using direct matching algorithms; however, this approach is insufficient. Many researchers' attempts to improve the speed and accuracy of matching still exhibit weak robustness [97] compared with recurrent local features [98]. To improve localization accuracy while maintaining computing costs below acceptable limits. It is necessary to ensure that the computational capacity does not exceed that of the model, and that it can process a certain number of repetitive features. A hierarchical localization model was proposed [99]. They searched for the smallest subset of the scene model by using a retrieval-based strategy [46]. We can estimate the correlations in this subset and perform real-time localization on large datasets.

To solve this image-search problem, a system can find similar images in a database when an image is queried. Previous research has also used the activation function as an image descriptor in the CNN layer, which appears in the empirical retrieval results, to demonstrate the benefits of learning-based features [100]. The training dataset was used to create a 3D model using the unlabeled images [101]. Additionally, the search technique used to find combinations of strongly positive and negative images for training influenced the results. This result encourages the use of locally aggregated match kernels to exploit specific image regions and provide discriminative image representation [102]. The Remap technique was introduced to address the aggregation process [103]. It uses the discriminating power of regions, where the Kullback divergence [104] values are used to investigate a situation in which region-based features are combined with multiple resolutions.

In contrast to direct matching methods that use local features [105], we entered matching problems by using a search image and a series of similar scene images. The matching rule states that local features are extracted and compared to identical 3D feature points in a scene model. After keypoint detection, the feature descriptors were extracted from the displayed vital points. Local features sensitive to scale changes, rotations, and viewpoint variations, while remaining constant. A detection regression problem was used to derive covariance constraints that can be used for automatic learning. By introducing new concepts in the legal field, canonical features extend the covariant restriction [106], increase the robustness of the learning process, and reduce its sensitivity to initialization settings.

The detected vital points were more stable when the detection was postponed to a later phase. We identified significant locations using local feature representations along the spatial and depth directions, and relative and absolute feature estimations [107]. A unique technique, DGC-Net [108], takes advantage of recently developed optical-flow methods. New method for finding image-pair correspondence with significant changes [109]. Develop a framework for learning a conditioned feature and unique score used by an attentional mechanism to select the best matches. It presents a dataset with camera poses related to the InLoc model for extensive indoor navigation [110], [111] and dense feature extraction. This plan was derived from several detailed inspections.

In contrast to techniques focused on matching, where matching descriptors are used to create 2D-3D correspondences explicitly, localization technologies are based on scene coordinate regression, where 3D scene coordinates are regressed directly from the query image. To predict the 3D scene coordinates of pixels, they can use a random forest or a neural network. Researchers have proposed a probabilistic selection-based differential RANSAC [112] and integrated a camera localization process for end-to-end learning. The trainable localization process outperformed the competition. To facilitate unsupervised learning, [113], [114] geometric constraints are applied to multiple views. After training, a hybrid classification regression forest can estimate the scene IDs and coordinates [115]. They described scene coordinate regression [116] as a combination of two tasks: the detection of object instances and the regression of local coordinates. Instead of embedding individual scene information into network parameters, SANet [117] is a technique for obtaining the representation of a scene from specific reference images and 3D coordinates. Table 2 lists the pose errors in structure-based camera-localization research. In addition to camera pose regression, these findings demonstrate the effectiveness of the structure-based localization techniques.

F. REGRESSION-BASED

In this section, we describe the regression-based estimation of camera pose, divided into relative and absolute camera pose regression, using a recurrent neural network or image retrieval to extract the pose of a reference image.

1) ABSOLUTE CAMERA POSE REGRESSION

CNN use absolute camera pose regression (APR) techniques to predict the camera pose of an input image [118] by implicitly expressing a scene using network weights. They adhere to the same process: a base network extracts the absolute features [119] that are then embedded in a high-dimensional space. The camera pose in the scene is then regressed using this embedding method. The training loss function and underlying base architecture are where existing methods differ. Three steps of absolute camera pose regression are shown depending on whether the network input is an image, image sequence, or video.

ARP through single image PoseNet [54] is the first method used to successfully and directly regress the camera pose from a single image. PoseNet uses similar expressions for the encoder, the localizer, and the regressor, uses a single

TABLE 2. The pose error of published structure-based approach on the 7-scenes dataset.

Reference	Published Year	Methods	Chess	Fire	Head	Office	Pumpkin	Kitchen	Stairs
[54]	2015	SCoRF (Uses RGB-D)	0.03m, 0.66°	0.05m, 1.50°	0.06m, 5.50°	0.04m, 0.78°	0.04m, 0.68°	0.04m, 0.76°	0.32m, 1.32°
[82]	2018	-	0.02m, 0.5°	0.02m, 0.9°	0.01m, 0.8°	0.03m, 0.7°	0.04m, 1.1°	0.04m, 1.1°	0.09m, 2.6°
[120]	2018	-	0.02m, 0.7°	0.03m, 1.1°	0.12m, 6.7°	0.03m, 0.8°	0.05m, 1.1°	0.05m, 1.3°	0.29m, 5.1°
[6]	2018	NetVlad+DensePE	0.03m, 1.05°	0.03m, 1.07°	0.02m, 1.16°	0.03m, 1.05°	0.05m, 1.55°	0.04m, 1.31°	0.09m, 2.47°
[6]	2018	NetVlad+SparsePE	4m, 1.83°	4m, 1.55°	2m, 1.65°	5m, 1.49°	7m, 1.87°	5m, 1.61°	12m, 3.41°
[113]	2019	-	0.02m, 0.8°	0.02m, 1.0°	0.04m, 2.7°	0.03m, 0.8°	0.04m, 1.1°	0.04m, 1.1°	0.18m, 3.9°
[117]	2019	SANet	0.03m, 0.88°	0.03m, 1.08°	0.02m, 1.48°	0.03m, 1.00°	0.05m, 1.32°	0.04m, 1.40°	0.16m, 4.59°
[121]	2021	RGB	0.19m, 1.11°	0.19m, 1.24°	0.11m, 1.82°	0.26m, 1.18°	0.42m, 1.41°	0.30m, 1.70°	0.41m, 1.42°
[121]	2021	RGB+3D Model	0.18m, 1.10°	0.19m, 1.24°	0.22m, 1.82°	0.25m, 1.15°	0.39m, 1.34°	0.38m, 1.68°	0.29m, 1.16°
[121]	2021	RGB-D	0.10m, 1.03°	0.11m, 1.05°	0.10m, 1.88°	0.12m, 1.03°	0.20m, 1.17°	0.21m, 1.41°	0.26m, 1.15°

image as the input for deep absolute pose regression [122], [123]. The overall pose result, which combines the position and orientation, is the output. This technique uses a single image to extract high-dimensional features and presents them as features, with poses represented sequentially as a six-dimensional vector. PoseNet [54] was the first study to recover camera poses [124] from RGB images by training the CNNs. PoseNet was accomplished without requiring additional techniques such as keyframes to estimate the pose. PoseNet outperformed the SIFT-based SfM technique, which failed abruptly when the training sample size decreased to below a certain threshold. PoseNet has been employed to create methods that improve the localization accuracy, rearrange the network, and change the loss function from a single image [125]. All the approaches using fixed loss employed the same method. Researchers have used stochastic gradient descent with an objective loss function to concurrently learn the position and orientation. A Bayesian-PoseNet [56] using Bernoulli distributions was developed to enhance localization performance and understand model uncertainty. The real benefit of Bayesian-PoseNet [56] is that it can be extended to a Bayesian model that can estimate localization uncertainty. To achieve this, we added dropout layers following the final output layer and sub-net to obtain the stochastic pose samples. Because it may predict position errors using uncertainty, the evaluation revealed a significant correlation between uncertainty estimation and location error to increase the re-localization precision of the direct PoseNet for indoor [126] and outdoor environments [127].

Hourglass-PoseNet [58] adds a second component to encode the rich and comprehensive information from coarse object structures and a third component to recover fine-grained object features to further increase localization accuracy. Instead of optimizing the hyperparameters for every training dataset, SVS-PoseNet [128] developed a new approach that relies on a classification system that employs the same parameters and performs better than the Cambridge Landmark dataset [54]. The training strategy was individually tuned for the translation and orientation. PoseNet's orientation expression is not unique. The computation of the sparse frames requires additional time. To efficiently minimize the sparsity of sampled poses, BranchNet [129] developed a novel two-branch network to address these issues. This network concurrently learns the orientation and translation representations. The encoder, decoder, and regressor are the three parts of hourglass-PoseNet [58]. It employs a modified version of ResNet34 [130] as an encoder-decoder, which is comparable to the encoder of the pipeline. BranchNet [129] uses an alternative design to that of PoseNet [54]. Following the fifth inception module, two distinct branches are used to predict the orientation and translation vectors. The loss function used throughout is the same as that used in PoseNet [54]. However, there are issues with the choice of the balance elements. Because of the joint loss that the loss function

utilizes, careful adjustment is required, particularly in a unique scene. Geometric PoseNet [131] provides learnable weight reduction to balance performance and improve robustness [132] for better localization [133], [134].

Unlike PoseNet, this technique maintains scalability and resilience without changing the fixed-balancing-factor hyperparameters of the loss function. Fixed Euclidean loss utilizes balanced hyperparameters, and can independently learn the location and orientation of an item from an image. However, determining their weights is expensive. This might mutually constrain the loss by training the estimate of the homoscedasticity task uncertainty [56] to represent the uncertainty terms and residual multiple regression analysis to reflect the regression performance [135]. In addition, a learnable geometric loss function [136] can be applied in various ways to add additional modules or functionalities, while obtaining geometric [137] restrictions. Before calculating the regression coordinates, the attention module added by AtLoc [60] directs the network to concentrate on the critical region of the input images, that is, a discrete, static, and stable area. AdPR [138] uses a discriminator network in addition to adversarial learning.

The 2048-dimensional fully connected layer of PoseNet was regressed using ResNet34, which is the encoder network used by AtLoc [60]. This may regress the position and increase accuracy. AdPR [138] uses the ResNet-18 network to extract features, because it performs better than VGG16 and AlexNet. To generate images that resemble the source image and to determine the camera pose more precisely, APANet [139] uses an adversarial network. To minimize the impact of the uncertainty of dynamic objects in dynamic contexts, PVL [140] implements a prior guided dropout mask. A dropout module is introduced before the feature [141] extractor encoder to output various uncertainty possibilities, which might increase the pose resilience under challenging circumstances, such as illimitation and perspective shifts. The feature map is reweighted after extraction using a self-attention module. Synthetically generating training data is another way to boost the localization performance. GeoPoseNet [142] and SPP-Net [143] use the same loss function. However, the SPP-Net uses a unique DNN architecture based on spatial pyramid max-pooling units.

APR through image sequences is another method for regressing the absolute camera pose [144]. Complete pose regression [145] and additional task constraints were combined to form the auxiliary learning. APR and extra loss comprise most of the loss functions in the auxiliary learning methodologies. The aforementioned approaches have been applied to obtain a perfect camera pose. In a pipeline, researchers may employ the relative pose regression loss. Deep neural networks still have disadvantages compared with conventional structure-based techniques [146], [147]. Therefore, some studies have suggested the use of absolute pose regression with image sequences [148]. Auxiliary learning uses image sequences. In contrast to approaches that employ a single image, auxiliary learning learns the final pose by estimating the relative pose using additional constraints. Globally consistent pose predictions may be required to enhance localization performance. According to MapNet [149], adding loss terms from the image sequence as a geometric restriction might improve the localization performance. Similar to MapNet, other techniques include auxiliary learning, which employs a weight coefficient factor to minimize the sum of the relative pose losses between image sequences and the absolute pose loss per image.

To improve the performance of AtLoc using a single image, AtLoc+ [60] integrates temporal constraints to simultaneously learn complete pose loss and relative pose loss. MapNet and AtLoc+ use the same loss function. DGRNet [114] proposed a novel architecture that includes a fully connected fusion layer (FCFL) to extract features from the images. Cross-transformation constraints (CTC) and mean square error (MSE) were added to the loss function to enhance regression performance. This method can be used to obtain visual odometry (VO) [150], and the localization results impose geometry-aware temporal and other limitations when using the image sequences. Additionally, DGRNet can access the network to obtain semantic segmentation results [151]. An image sequence regression network frequently extracts features using ResNet-50 [130] with some modifications.

APR through video is use video clips can substitute images or sequences of an image in pose regression by adding the temporal uniformity requirement [152], [153]. Mobile devices can quickly access videos and other sensor data. It is possible to synchronize videos with input data such as visual odometry [154], IMU [155], [156] sensors such as accelerometers and gyroscopes, and GNSS data by aligning timestamps. Similar to only one image or image sequence-based ARP work, video-based ARP approaches use CNN feature extraction and a localizer regressor to recover the rotation and orientation and incorporate multiple additional data, as in videos. VidLoc is a convolutional neural network and recurrent neural network-based model used to smooth pose estimation from image or video inputs and regress the camera pose. We constructed a network using a bidirectional LSTM component to represent temporal features with memory elements, a few gates, and GoogleNet [55] inception to retrieve the visual features without fully connected layers [157]. Researchers can use a bidirectional LSTM network with two hidden states to process forward and backward directions [158]. They used this method, with only one hidden state, obtain the camera pose. The weighted translation and orientation errors from the LSTM [159] output are used to calculate the VidLoc [61] network loss.

MapNet [149] and MapNet+ [149] employ a global average pooling layer to extract features using ResNet34 [130], and have the same network design. Visual odometry loss and complete pose loss were computed to increase the accuracy of MapNet estimates. To further enhance pose regression, the method uses GNSS and IMU data. The self-supervised

TABLE 3. Qualitative comparison of absolute and relative camera localization.

Reference	Methods	Robustness
[54]	PoseNet	Lighting, motion blur, camera intrinsic
[56]	Bayesian PoseNet	Large viewpoint or appearance changes
[57]	LSTM-PoseNet	Motion blur and illumination changes
[61]	VidLoc	Temporal smoothness
[58]	Hourglass PoseNet	Continuous pose optimization
[83]	BranchNet	Motion-blur, flat surfaces, lighting
[160]	GPoseNet	Choice of hyperparameters
[149]	MapNet	Online, locally smooth and drift-free
[60]	AtLoc	Dynamic objects, illumination
[66]	BIM-PoseNet	Motion blur, light flare

learning method combines labelled and unlabeled data through visual odometry or sensors and exhibits enhanced performance in testing settings. It uses videos as input by VidLoc, MapNet, and MapNet+, some of which combine unlabeled data to enhance supervised learning. To regress the camera pose and output the probability of the pose estimate, VidLoc included a bidirectional RNN. To improve the efficiency of regression, MapNet and MapNet+ use visual odometry in the loss function. Table 3 summarizes the qualitative comparisons based on the published statistical regression-based techniques. The prime factors for evaluating camera localization performance are robustness and accuracy. The localization performance improved with increased robustness and flexibility in response to changes in the scene environment.

2) RELATIVE POSE REGRESSION

The coordinate system determines the mapping from the pixels in the object images to the camera poses, which are learned by using an absolute camera-pose regression model. Thus, bounded coordinate transfers that offer learnable physical geometric information are realized through cross-scene learning. Relative camera pose regression [161], [162], [163], [164] techniques produce reference images. For the relative camera pose, regression may be computed using the previous image retrieval method [165], which first determines the image in the database that is most similar to the query image, and then calculates the absolute pose of the target image after predicting their respective relative poses. NNnet [59] initially proposed an image-retrieval-based relative pose regression approach. The approach inputs a search image and an image database containing ground-truth poses. Using a sequence of images, we regressed the relative pose using a Siamese network with two customized ResNet34 [130] nodes and a constant-loss function. The network branch developed a feature extractor that can locate the closest neighbor image of the query image. We can then obtain the absolute pose of the query image by combining the relative pose [166] with the ground-truth pose of the neighboring image. RelocNet [167] further enhances NNnet [59] by applying geometric relative pose loss and continuous metric learning to learn global image features with a camera pose to improve the performance. Relative pose loss employs rotation and translation matrix representations to understand the difference between two pose matrices. The performance limitation of previous retrieval-based relative regression algorithms that use the same features for the retrieval and regression modules was addressed by CamNet [168], which provides a novel pipeline divided into three phases. The phases included coarse, fine, and relative-pose regressions. Each phase is based on Siamese architecture with three branches. Regression is more accurate and scalable because of its coarse-to-fine design.

In analyzing the earlier relative pose regression approach based on image retrieval in [169], a new framework for computing the absolute pose was proposed, which included fundamental matrices and a modified RANSAC [170]. A matching score map for an additional regression, that is, the important matrix, was trained using the Siamese ResNet34 [130] network. They optimized the fundamental matrix with two 9D vectors by using the Euclidean distance loss function. Several systems have attempted to retrieve the relative pose using artificial neural networks to avoid extensive database collection and drawn-out test periods. A relative NN [171] suggests that an end-to-end system can regress the relative pose between two cameras by using two input images. They used a Siamese network of two branches to conduct regression using a fixed Euclidean loss, which worked well with the robot image dataset [172]. AnchorNet [173] defines anchor points as visible landmarks to learn the relative anchors and offsets of the query image and defines anchor points as visible landmarks. The multitask model identifies the offsets of classified anchor points and categorizes the query image to which particular anchor points are located. Retrieval-based techniques use a multistage strategy, with the retrieval phase serving as the central element of the process to achieve absolute and regressed relative poses. Another method for implicitly regressing a closed stance within a network is the use of CNN-based algorithms.

Table 4 shows the pose error of regression-based camera pose estimation. We also investigated the most prominent indoor localization approach. Most participants exhibited temporal uniformity throughout their journey and contained significant errors that made them prone to motion blurring. Although the LSTM-PoseNet [57] and VidLoc [61] approaches produced substantially smoother trajectories, they did not produce consistent trajectory outputs.

TABLE 4. Summary of published absolute and relative camera localization accuracy on the 7-scenes dataset.

Reference	Methods	Chess	Fire	Head	Office	Pumpkin	Kitchen	Stairs
[54]	PoseNet	0.32m, 4.06°	0.47m, 7.33°	0.29m, 12.0°	0.48m, 6.00°	0.47m, 4.21°	0.59m, 4.32°	0.47m, 6.93°
[56]	Bayesian PoseNet	0.37m, 7.24°	0.43m, 13.7°	0.31m, 12.0°	0.48m, 8.04°	0.61m, 7.08°	0.58m, 7.54°	0.48m, 13.1°
[57]	LSTM PoseNet	0.24 m, 5.77°	0.34 m, 11.9 °	0.21 m, 13.7°	0.30 m, 8.08°	0.33 m, 7.00°	0.37 m, 8.83°	0.40 m, 13.7 °
[58]	Hourglass PoseNet	0.13m, 6.46°	0.26m, 12.72°	0.14m, 12.34°	0.21m, 7.35°	0.24m, 6.35°	0.24m, 8.03°	0.27m, 11.82°
[129]	BranchNet (Multi-task CNN)	0.18m, 5.17°	0.34m, 8.99°	0.20m, 14.15°	0.30m, 7.05°	0.27m, 5.10°	0.33m, 7.40°	0.38m, 10.26°
[61]	VidLoc	0.18m, N/A	0.26m, N/A	0.21m, N/A	0.36m, N/A	0.31m, N/A	0.26m, N/A	0.14m, N/A
[149]	MapNet	0.08m, 3.25°	0.27m, 11.69°	0.18m, 13.25°	0.17m, 5.15°	0.22m, 4.02°	0.23m, 4.93°	0.30m, 12.08°
[160]	GPoseNet	0.20m, 7.11°	0.38m, 12.3°	0.21m, 13.8°	0.28m, 8.83°	0.37m, 6.94°	0.35m, 8.15°	0.37m, 12.5°
[60]	AtLoc	0.10m, 4.07°	0.25m, 11.4°	0.16m, 11.8°	0.17m, 5.34°	0.21m, 4.37°	0.23m, 5.42°	0.26m, 10.5°
[66]	CamNet	0.04m, 1.73°	0.03m, 1.74°	0.05m, 1.98°	0.04m, 1.62°	0.04m, 1.64°	0.04m, 1.63°	0.04m, 1.51°

BIM-PoseNet [66] predicts a smoother and more consistent trajectory than previous approaches.

- Critical Analysis: The camera poses are estimated using a fixed loss function, which takes time to minimize the loss of a dataset because it uses a balance factor to various weighted components. Later, it suggested using a learnable loss that outperforms fixed loss methods by automatically balance the pose losses with the addition of homoscedastic uncertainty. Some approaches suggested reprojection loss, GPoseNet loss, fixed loss, and learnable loss methods to include different information formats. In image sequence regression networks, the modified ResNet-34 and ResNet-50 are widely used to extract features. Relative and absolute pose losses improve regression in MapNet, VlocNet, and AtLoc. The semantic restriction is added to the loss function by VlocNet+. DGRNet combines CTC and MSE to determine loss. Videos are used as input by VidLoc and MapNet, and some of these algorithms use unlabeled data to improve supervised learning. When regressing the camera's 6DoF pose, VidLoc incorporates a bidirectional RNN and outputs the probabilistic for pose estimation. MapNet+ primarily incorporates visual odometry into the loss function to enhance the efficiency of regression.

G. INDOOR SLAM ALGORITHMS

The ability to simultaneously generate a map and estimate an indoor camera's pose in an unknown environment, simultaneous localization and mapping (SLAM) algorithms are the subjects of extensive research. Because of their roots in a small, inexpensive camera system that ensures their advantages over other camera-based SLAM techniques, visual-based SLAM methods play a crucial role in this area. The simplicity of sensor construction, reduced size, and the low cost of visual-based SLAM algorithms make them particularly attractive. Visual SLAM (V-SLAM), visual-inertial SLAM, and RGB-D SLAM are the three primary divisions of the visual-based approaches. Using the deep learning application paradigm as a starting point, researchers address the application position of deep learning to use VSLAM. At first monocular SLAM (MonoSLAM) [29] system proposed in 2007, and V-SLAM quickly became famous as a research area. The underlying theoretical study on classic V-SLAM has reached a mature stage before the appearance of relatively mature V-SLAM solutions like Oriented FAST and Rotated BRIEF SLAM (ORB-SLAM) [31], Semi-direct Visual Odometry (SVO) [30], and direct sparse odometry (DSO) [37]. Most of the ensuing effort focused on improving suitable scenes [174] and expanding sensors [175]. The study of the conventional V-SLAM method has increasingly

become application-focused a pose regression using images as input and pose as its output.

1) VISUAL SLAM

The V-SLAM algorithm became popular after the researcher proposed it in [176]. They also presented several SLAM algorithms on visual sensors, including Parallel Tracking and Mapping (PTAM) [177], which made it possible to carry out many SLAM tasks in parallel, and MonoSLAM [29], which is based on monocular cameras. Based on PTAM, proposed Oriented FAST and Rotated BRIEF (ORB-SLAM) [31]. V-SLAM has been proposed as a direct approach called Large-Scale Direct Monocular SLAM (LSD-SLAM) [33]. Direct Sparse Odometry SLAM (DSO-SLAM) [37] has been suggested because of its high accuracy and efficiency. Figure 11 shows the timeline of most representative visual SLAM.

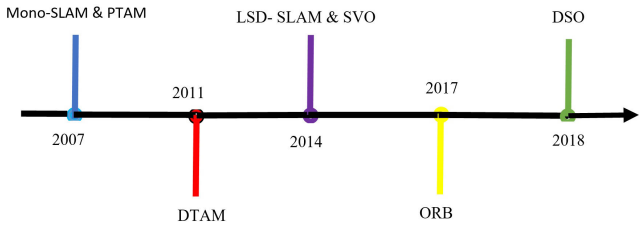

FIGURE 11. Timeline of most representative visual SLAM.

- Monocular SLAM: Davison et al. [29] suggested MonoSLAM as the initial monocular SLAM algorithm. Initializing the system is the first phase in the algorithm. The state vector is then updated while considering a steady-speed motion model. The extended Kalman filter (EKF) estimates the real-time camera movement and environment structure. Vincke et al. evaluated the MonoSLAM algorithm's implementation in [178] by merging many cameras and a multi-processor architecture. The initialization process of MonoSLAM causes a known target that is only sometimes reachable. They used neither loop closure detection nor global optimization strategies in this algorithm. Finally, it simply reconstructs a map containing features, which could be problematic for applications that need a more precise reconstruction.
- Parallel Tracking and Mapping: The Parallel Tracking and Mapping (PTAM) [177] technique is another innovative approach. PTAM uses the correspondences to calculate the camera pose to minimize re-projection error. Several features can be represented on the map thanks to PTAM. This method also performs global optimizations. The map produced by PTAM does not consider loop closure, making it better suited for locating landmarks. It has a non-negligible energy consumption and causes user intervention to establish the basic keyframes, making it inappropriate for inexpensive embedded devices [179].
- Dense Tracking and Mapping: The concept of Dense Tracking and Mapping (DTAM) was first forth by Newcombe et al. [180]. A dense mapping algorithm and a dense tracking algorithm are essential components. The first stage computes the data expenditure volume representing the average photometric error over many frames calculated for the inverse depth of the current frame to estimate the depth values. DTAM calculates the movement control in the dense tracking step by matching the current frame with an image from a dense model presented in a virtual camera. Although the algorithm produces a precise and thorough reconstruction, the amount of density reconstruction impacts how much computing power is required to store and analyze the data. DTAM does not use global optimization or loop closure techniques.
- Semi-Direct Visual Odometry: The Semi-direct Visual Odometry (SVO) [30] technique is quick and doesn't require calculating several descriptors, uses semi-direct visual odometry. Consumer computers can produce 300 frames per second, and an unmanned aerial vehicle (UAV) can produce 55 frames per second. SVO makes using a probabilistic mapping technique and direct pixel correspondences. Because this method does not require feature extraction for every frame [31], it may operate with high frame rate. It can therefore used in low-cost embedded systems, such as the embedded platform under consideration [30]. Because of the correlation of short-term data, the SVO's accuracy is limited [32]. SVO does not use loop closure or global optimization strategies.
- Large-Scale Direct Monocular SLAM: A direct approach that provides a semi-dense reconstruction is the large-scale direct monocular SLAM (LSD-SLAM) [33]. To determine the pose of the sensor, this SLAM reduces the photometric error. The LSD-SLAM then completes the depth map estimation step's keyframe selection. The depth map of the algorithm is initialized if a new keyframe is added; otherwise, the depth map of the currently selected keyframe is improved via some small-baseline stereo comparisons. The LSD-SLAM inserts the new keyframe into the map at this point and optimizes it using a pose-graph optimization technique. This method uses loop closure and global optimization to create large-scale maps in real-time. In [181], Boikos and Bouganis used FPGA technologies to build the LSD-SLAM method. To handle the tracking thread's more expensive activities, the designers of [181] constructed two processors on the FPGA. The LSD-SLAM map estimate algorithm, which has a lesser accuracy than others like PTAM and ORB-SLAM [31], is primarily based on pose-graph optimization [32].
- ORB-SLAM 2: The main component of the V-SLAM is the Oriented FAST and Rotated BRIEF (ORB-SLAM) introduced by Mur-Artal et al. [31]. It is a well-known system that employs the feature point technique. Scale

drift was an issue with the monocular camera used as the visual sensor in ORB-SLAM [31]. The first SLAM system with monocular, stereo, and RGB-D cameras. ORB-SLAM2 [34] was developed in 2016 in response to ORB drawbacks. It incorporated a lightweight positioning system that employed VO to track the unmapped regions and match map points to achieve zero-drift positioning. The most advanced feature-based method is the ORB-SLAM2 [34] method, built on the ORB-SLAM algorithm [31]. Tracking, local mapping, and loop closing are three concurrent threads it uses to function. Yu et al. [35] and Abouzahir et al. [36] deployed the ORB-SLAM method on several CPU and GPU-based systems, and they assessed the accuracy of each thread on each platform.

- Direct Sparse Odometry: The direct sparse odometry (DSO) technique [37] combines a straightforward strategy with a sparse reconstruction. The DSO algorithm takes into account a recent frame window. It carried continuous optimization by using a local bundle adjustment to optimize both the inverse depth map and the keyframes window. Although Gao et al. [182] suggested an update of the DSO technique that includes loop closure recognition and pose graph optimization, the original version of this approach does not contain universal optimization or loop closure.

2) VISUAL INERTIAL SLAM (VI-SLAM)

- Multi-State Constraint Kalman Filter: Stereo and monocular cameras can both be used to create the multi-state constraint Kalman filter (MSCKF) [182]. Because of the MSCKF's low computing cost [182] and reputation as one of the best filter-based algorithms in the literature, embedded implementations are a good fit for this approach. Delmerico and Scaramuzza [183] employed various hardware platforms depending on CPU architectures to build visual-inertial SLAM algorithms. Figure 11 shows the timeline of most representative visual inertial SLAM.

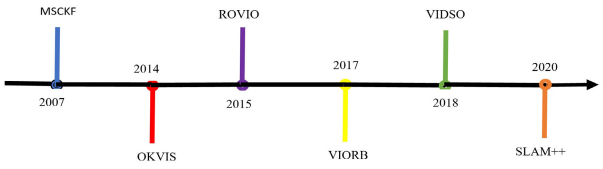

FIGURE 12. Timeline of most representative visual SLAM.

- Open Keyframe-Based Visual-Inertial SLAM: The optimization-based approach Open Keyframe-based Visual-Inertial SLAM (OKVIS) [184] uses keyframes. It creates an objective function out of the IMU information, and re-projection terms enable the algorithm to jointly optimize the weighted re-projection defect and the temporal error from the IMU. It implemented the OKVIS algorithm using various CPU platforms in Delmerico and Scaramuzza's [183] work. Nikolic et al. [185] assessed the performance of the OKVIS algorithm using an FPGA-CPU architecture.
- Robust Visual Inertial Odometry: Another filter-based approach that uses the EKF is the Robust Visual Inertial Odometry (ROVIO) technique [186]. Like other filter-based approaches, it propagates the state using IMU data and updates the filter with camera data. The prediction and update phase uses the patches to get the innovation term. It suited the ROVIO technique for embedded implementations because it delivers high accuracy and robustness with minimal resource usage [183]. The method, however, turned out to be less accurate and more susceptible to per-frame process time than other algorithms, including VI-DSO [187].
- Visual Inertial ORB-SLAM: The previously presented ORB-SLAM method [34] provides the foundation for the Visual-Inertial ORB-SLAM (VIORB) technique [188]. The VIORB algorithm, which first used map reuse in a visual-inertial approach, offers good predictive performance [189] and memory use. The authors in [32] suggested the ORB-SLAM3 method based on the ORB-SLAM2 and VIORB algorithms. Compared to VIORB, the system's predecessor, initialization times are less.
- Monocular Visual-Inertial System: A monocular visual-inertial state estimator is the Monocular Visual-Inertial System (Mono-VINS) [190]. The first step of the measurement process that extracts and tracks features is the pre-integration of the IMU information in-between frames. The algorithm executes an initialization step to supply the input data for a non-linear optimization problem that minimizes the optical and inertial errors. It combined the IMU measurements and feature observations with a re-localization and pose-graph optimization module implemented by the VINS. They can use the method while considering stereo and binocular approaches [191].
- Visual-Inertial Direct Sparse Odometry: The DSO [37] algorithm, which has already been given, is the foundation of the Visual-Inertial Direct Sparse Odometry (VI-DSO) [187] method. The approach seeks to minimize a nonlinear dynamic model-based energy function incorporating photometric and inertial errors. The VI-DSO expands the DSO considering inertial information compared to the original DSO and other algorithms. Unfortunately, the startup process could be faster because it depends on bundle change [32]. The approach does not detect loop closures or perform global optimization, and no embedded implementations exist in the literature.
- ORB-SLAM3: A method that integrates the ORB-SLAM and VIORB techniques is called the ORB-SLAM3 algorithm [192]. An active map used by the tracking thread and non-active maps needed for re-localization and location recognition is maintained by

ORB-multi-map SLAM3'srepresentation, known as Atlas. This algorithm uses loop closures and global optimizations, and they can apply it to monocular, stereo, and RGB-D cameras. However, the performance of ORB-SLAM3 online was shown to have many things that the authors could have improved [175]. Although the algorithm in [193] performed well, it could only evaluate some sequences and produced incorrect outdoor sequence estimates.

3) RGB-D SLAM

SLAM systems based on RGB-D sensors comprise a depth sensor and a monocular RGB camera, enabling SLAM systems to immediately obtain depth information with practical accuracy in real-time using affordable hardware. Since it uses a lot of memory and power, this method is best suited to indoor settings [194]. Since the SLAM receive the depth map directly from the RGB-D devices. Most RGB-D-based devices use the ICP algorithm, which uses iterative closest points to locate the sensor. The benefits of RGB-D systems include providing dense depth maps and color image data. Figure 13 shows the timeline of most representative RGB-D SLAM.

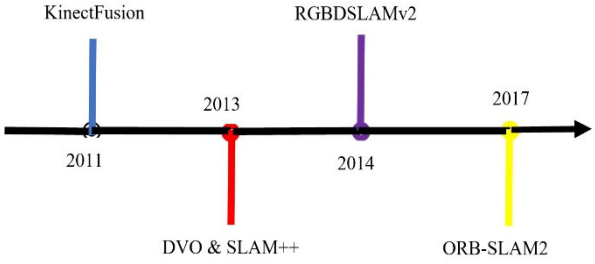

FIGURE 13. Timeline of most representative RGBD SLAM.

- KinectFusion: The first RGB-D sensor-based algorithm to function in real-time was the KinectFusion algorithm [195]. In maximum medium-sized spaces, the KinectFusion algorithm can map robots effectively in Robotics [195]. Yet, when loop termination is not performed, drift errors accrue [196]. In [197], Nardi et al. suggest a KinectFusion implementation and evaluate it on several CPU- and GPU-based systems. Bodin et al. [198] used the framework proposed by [197] to develop the KinectFusion on two separate CPU and GPU architectures.
- Dense Visual Odometry: Keyframe-based algorithms are used in the dense visual odometry SLAM (DVO-SLAM) method developed by Kerl et al. [199]. It reduces camera motion and photometric inaccuracy when gaining depth values and pixel coordinates between keyframes. The algorithm calculates an entropy value for each input frame compared to a predefined threshold. Even though loop detection employs a different threshold value, the same approach is applied. A SLAM graph represents the map, with camera poses as the vertex and keyframe transformations as the edges. This algorithm performs loop closure identification and is resistant to textureless situations. The technique does not undertake a detailed map reconstruction; the map representations depend on a presentation of the keyframes. Table 5 shows the latest SLAM for indoor environment.
- SLAM++: The object-oriented SLAM algorithm SLAM++ [200] uses previously observed scenes with recurring objects and structures, like a classroom. Besides performing loop closure detection, SLAM++ further improves scene description by considering the object's repeatability. Yet, the algorithm works best with scenes that are already well-known.
- RGBDSLAMv2: One of the most well-known RGB-D-based algorithms, the RGBDSLAMv2 [201], depends on feature extraction. The ICP algorithm estimates pose after performing the RANSAC method to predict the transition between the matched features. For RGBD-SLAMv2 to operate properly, the sensor must move slowly and consume a lot of computation [196].
- Critical Analysis: The feature-based technique ORB-SLAM performed better than the LSD-SLAM in terms of absolute translation RMSE and relative rotation RMSE. The LSD-SLAM and ORB-SLAM more diverse indoor environment.

H. MILTI-CAMERA LOCALIZATION

Araar et al. [202] proposed a system with multi-cameras for indoor localization. The experimental findings showed the usefulness of their system in attaining indoor localization, and the authors comprehensively explained its implementation. The study claims that it offers a "low-cost" solution, but it does not specify the actual cost of the suggested system. Ince and Kim [203] presented a novel technique for simultaneous localization and mapping (SLAM) in a multi-camera system without calibration. The proposed approach integrated collaborative scene mapping with independent object tracking for each camera view. They described the proposed algorithm and its implementation using experimental findings on a real-world dataset.

Sewtz et al. [204] presented a multi-camera localization for dynamic scenes. Their proposed solutions address occlusions, moving objects, and shifting illumination. These strategies include feature-based and deep learning-based approaches, which assess the effectiveness of each using data obtained from a real-world scenario. The findings show that the proposed methods produce high precision and robustness, making them suitable for applications such as robotics and surveillance. Liu et al. [205] present a multi-camera localization for indoor environments. This paper thoroughly explains the system design, details of the calibration procedure, and algorithms for object tracking and position estimation. However, a thorough assessment of the system, including the accuracy of the positioning findings under various circumstances, the system's resistance to noise and occlusions, and

TABLE 5. Some latest slam for indoor environment.

Latest SLAM	Reference	Camera Type	Environment	Characteristics
PL-SLAM	[206]	Stereo camera	Rooms	More diverse, less computational time
CubeSLAM	[207]	Monocular camera	Indoor	Static and dynamic scene, 3D cuboid object detection
DymSLAM	[208]	Stereo camera	Indoor	Improving accuracy, dynamic environment
TIMA SLAM	[203]	Multi-camera	laboratory	Good compatibility
FSD-SLAM	[209]	Monocular camera	Indoor	Accurate camera pose estimation, Point cloud integration
DSP-SLAM	[210]	monocular, stereo	chairs	Pose with less drift, real-time performance.

TABLE 6. Common datasets used for indoor camera localization tasks.

No	Datasets	Dataset used in researches	Environment	Affiliation and Year	Imagery	Train and Test images
1	7-Scenes	[6], [19], [53], [54], [56], [58]–[61], [77], [78], [81]–[83], [113], [115], [117]–[121], [128], [131], [132], [135], [138], [141]–[143], [147], [149], [151], [154], [167], [168], [172], [173], [211], [216], [219]	Indoor	Microsoft, 2013	RGB-D sensor	26000, 17000
2	TUM LSI	[57], [123], [127], [154], [220]	Indoor	Technical University Munich, 2017	Monocular camera	875, 220
3	InLoc	[2], [4], [6], [105], [117], [165]	Indoor	Tokyo Technology, 2018	Panoramic images	9970, 350

the computing effectiveness of the algorithms, is lacking in this work.

VI. COMMON DATASET

The lack of adequate datasets is a problem for the success of deep learning-based camera localization. Recently, many public databases have become available for indoor localization activities. This study presents examples of datasets commonly used for indoor localization using a deep-learning approach. We focused on the image processing. However, there is a growing trend in the research community to compile lists of these datasets. Only a few public indoor databases, such as Microsoft Researchers 7-Scenes [211], TUM LSI [57], InLoc [6], and InteriorNet [212] are available. The few image processing objects used for localization have contributed to the development of public databases. A complete list of public databases is available online. The 7-Scenes dataset comprises seven separate interior scenes and is a commonly used RGB-D dataset. The RGB-D images were captured using a 640 × 480 resolution handheld Kinect camera and linked to the ground-truth camera positions captured using Kinect fusion [213]. A detailed three-dimensional (3D) model was also applied to each scene. Each scene comprises multiple sequences of tracked RGB-D camera frames that are split into training and testing data. In [214], a hybrid image dataset for object recognition that includes both natural and artificial images was developed.

We used the publicly available 7-Scenes [211] dataset to evaluate recurrent neural network [215], [216] related algorithms. Indoor camera localization is challenging because it contains many blurred images and exhibits motion-blurring [217]. The 7-Scenes are adopted to evaluate the proposed method. 7-Scenes [211] have a very high number of images in a small spatial extent; hence, they are more suitable for indoor camera localization. Some studies used the InteriorNet [212] dataset for model training and evaluation. Imperial College London released this dataset in 2018. The dataset comprised 10 K scenes, 1.7 M rooms, and 5M frames. RGB, depth, and semantic instances are available for this dataset. The image resolution of the dataset to create photorealistic images and related ground-truth data. 640 × 480 pixels. An end-to-end pipeline was suggested in produced various trajectory types by adjusting the linear velocity ratio and angular velocity, including the open and closed shutter positions in a single-trajectory view. To create motion-blurred renderings, many renderings were averaged from intermediate poses. The InteriorNet dataset uses path tracing [62] to provide high-quality image renderings. Path tracing is a Monte Carlo technique that renders images with an accurate global illumination. One feature that it mimics from the actual world is motion blur, and the renderer simulates camera motion blur [218]. The common datasets used for the camera indoor localization listed in Table 6. Table 7 comprise the publicly accessible benchmark dataset for evaluating the original studies' stated SLAM algorithms.

Datasets directly provided ground truth poses. The difference between predicted and ground truth poses measured pose accuracy. To evaluate sophisticated camera localization

TABLE 7. Common datasets used for indoor SLAM.

Dataset	Reference	Year	Environment	Camera
TUM RGB-D	[221]	2012	Indoor	RGB-D camera
ICL-NUIM	[222]	2014	Indoor	RGB-D camera
EuRoC	[223]	2016	Indoor	Stereo-cameras
TUM MonoVO	[37]	2017	Indoor	Non-stereo cameras
TUM VI	[224]	2018	Indoor	Stereo-camera

algorithms, large-scale multidimensional datasets that include multiple collection platforms, environments, and images such as illumination, viewpoints, and scene changes are required. Table 6 lists the popular datasets used in camera localization tasks, including 7-Scenes [211], TUM-LSI [57], and InLoc [6].

The TUM RGB-D database [127] comprises some image sequences with depth and color images captured using a Microsoft Kinetic within indoor settings on two separate platforms. The ICL-NUIM [129] is another significant benchmark dataset. Eight artificially created it used indoor scenarios in the dataset to evaluate 3D reconstruction, which focuses on RGB-D methods. It produced the sequences through a handheld RGB-D camera.

The EuRoC standard dataset [22] is frequently utilized to evaluate visual-inertial SLAM algorithms. It provided eleven stereo image sequences and IMU data from the data gathered in two indoor situations. The TUM MonoVO is an extensively used dataset to investigate monocular systems [29]. It includes various photometrically adjusted indoor sequences captured using two non-stereo monocular cameras. The TUM VI dataset [131] is made publicly available for evaluating visual-inertial systems. It offers some scenes shot indoors using a stereo camera and an IMU. Because the sensing system is portable, it could not determine actual ground truth for all the sequences using the TUM MonoVO.

VII. LIMITATIONS

However, there are still some difficulties in image-based indoor-camera localization. The localization accuracy must include the training costs and offline training. It is difficult to view online, and is not functional. It can perform online localization in different contexts and achieve excellent indoor performances. The most challenging aspect of the 3D structure-based localization approach is that it is difficult to adjust to scenes with changes in the environment and repeating elements, textures with less surface, illumination changes, motion blur, and significantly reduced viewpoint adjustments and localization results.

PoseNet results in a reduction in the intensity of noisy things when responding to them. The ConvNet has identified this dynamic object as inappropriate for indoor localization [54]. The LSTM-PoseNet model could not get accurate reconstructions of SfM from the TUM-LSI dataset [57]. In most images, there is no texture, so SfM and COLMAP fold repeating structures on top of each other. Traditional visual SLAM systems cannot expand maps to unknown territories, unlike MapNet and MapNet+.

LSTM-PoseNet [57] and VidLoc [61] approaches produced smoother trajectories, but they did not produce consistent trajectory outputs. BIM-PoseNet [66] predicts a smoother and more consistent trajectory than the PoseNet and LSTM-PoseNet.

Using a constant loss function to estimate the indoor camera pose utilizing a balance factor to balance different weights requires considerable time to minimize the loss. Later, researchers proposed a learnable loss that performs better than fixed loss strategies by effectively minimizing the pose losses with homoscedastic uncertainty.

DGRNet can get the localization result by using image sequences and imposing geometric-aware time constraints. Moreover, DGRNet could use the network to obtain semantic segmentation results. Further to the localization results DGRNet might access the network to obtain semantic segmentation results.

One significant problem is tracking failure of the large indoor environments [225]; the algorithms still need to identify and correlate features in the currently received image, leading to inaccurate pose estimation. Authors have been investigating innovative approaches to the SLAM issue to address this challenge. Recent research proposed integrating recurrent neural networks and spectral methods to improve the system's resilience [226].

The assumption of fixed scenarios, whereas the real world has dynamic surroundings, is a significant problem that reduces the robustness of SLAM algorithms and may lead to tracking and reconstructing failures [227]. Modern SLAM algorithms consider computing resources. The current topic brings up the open issue of memory consumption by map storing.

Map sparsity is one of the storage issues that affect resource use. Dense or semi-dense maps provide a more accurate representation of the scene, but this characteristic has implications for resource utilization. The sparse map uses less power than a semi-dense or dense model [228].

The performance of several deep-learning algorithms can be improved when applied to the indoor environment. These include PoseNet, Bayesian-PoseNet, LSTM-

PoseNet, VidLoc, AtLoc, and InLoc. For the 7-Schenes dataset, these algorithms produced good results. The best SLAM algorithms are PL-SLAM, CubeSLAM, DymSLAM, TIMA-SLAM, FSD-SLAM, and DSP-SLAM. These algorithms provide good results for TUM RGB-D, TUM MonoVO, and TUM VI datasets in indoor environments.

VIII. CONCLUSION

We have made remarkable progress in various indoor camera localization studies. This paper addresses the main sections and compares the different techniques. Many structure-based studies have been based on 3D point characteristics obtained from 2D scene images. We limited this challenging scenario to texture-less surfaces, less lighting, and fewer changes in weather. However, multiple features can improve the localization. Deep learning can be applied to solve features from many cameras end-to-end and speed up feature computations. An essential step in developing an accurate positioning system is learning to enhance the positioning performance under challenging situations. Applications that employ camera positions will become lightweight in the future, making them more rapidly and easily applicable to small-time computing devices.

We also provided an overview of the primary visual-based SLAM methodologies and a concise description and in-depth assessments of several of the most notable techniques. Researchers can use the offered article as a starting point for their initial analysis and analyze each criterion in light of their application. Also, we examined the key benchmarking datasets for evaluating V-SLAM and visual odometry algorithms, presented some significant difficulties, and proposed future approaches for the subject.

IX. FUTURE WORK

Some sensors, such as LiDAR, Wi-Fi, IMU, and Bluetooth, can gather more detailed localization information. We should overcome the difficulty of heterogeneous features from many sensor data to achieve more precise positioning. We need to develop a more accurate and potent localization system by effectively combining the data from many sensors as one of the future directions.

Real-world scenes sometimes significantly depend on the semantics of features. The semantics of the features makes a new approach. Semantic data simplify the exclusion of dynamic object attributes that affect localization outcomes. In the future, a semantics model should be helpful for a more accurate indoor positioning system.

A multi-camera setup may improve robotic applications and offer a 360-degree panoramic field of view. One potential future research direction is to develop a deep learning system to analyze the features of many cameras end-to-end and speed up feature calculation time.

Analyze the nuclear decommissioning situations in subsequent publications using the suggested criteria. Researchers will choose the optimum SLAM algorithm by considering the various qualities and particularities of such an environment and application.

ACKNOWLEDGMENT

The authors are grateful for support from the Faculty of Computing, Universiti Teknologi Malaysia.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] B. Morawska, P. Lipinski, K. Lichy, and K. Adamkiewicz, "Transfer learning-based UWB indoor localization using MHT-MDC and clusterization-based sparse fingerprinting," *J. Comput. Sci.*, vol. 61, May 2022, Art. no. 101654.

[2] N. Li and H. Ai, "EfiLoc: Large-scale visual indoor localization with efficient correlation between sparse features and 3D points," *Vis. Comput.*, vol. 38, no. 6, pp. 2091–2106, Jun. 2022.

[3] B. Hu, Q. Yu, and H. Yu, "Global vision localization of indoor service robot based on improved iterative extended Kalman particle filter algorithm," *J. Sensors*, vol. 2021, pp. 1–11, Jul. 2021.

[4] J. Zhang, S. Tang, K. Qiu, R. Huang, C. Fang, L. Cui, Z. Dong, S. Zhu, and P. Tan, "RenderNet: Visual relocalization using virtual viewpoints in large-scale indoor environments," 2022, *arXiv:2207.12579*.

[5] S. Hausler, M. Xu, S. Garg, P. Chakravarty, S. Shrivastava, A. Vora, and M. Milford, "Improving worst case visual localization coverage via place-specific sub-selection in multi-camera systems," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10112–10119, Oct. 2022.

[6] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "InLoc: Indoor visual localization with dense matching and view synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7199–7209. Accessed: Mar. 10, 2022. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/

[7] G. Ge, Z. Qin, and L. Fan, "An improved VSLAM for mobile robot localization in corridor environment," *Adv. Multimedia*, vol. 2022, pp. 1–10, May 2022.

[8] X. Lei, F. Zhang, J. Zhou, and W. Shang, "Visual localization strategy for indoor mobile robots in the complex environment," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2022, pp. 1135–1140.

[9] W. Ma, S. Zhang, and J. Huang, "Mobile augmented reality based indoor map for improving geo-visualization," *PeerJ Comput. Sci.*, vol. 7, p. e704, Sep. 2021.

[10] C. Shi, J. Li, J. Gong, B. Yang, and G. Zhang, "An improved lightweight deep neural network with knowledge distillation for local feature extraction and visual localization using images and LiDAR point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 184, pp. 177–188, Feb. 2022.

[11] V. Upadhyay and M. Balakrishnan, "Monocular localization using invariant image feature matching to assist navigation," in *Proc. Int. Conf. Comput. Helping People With Special Needs*. Cham, Switzerland: Springer, 2022, pp. 178–186.

[12] E. H.-C. Lu and J.-M. Ciou, "Integration of convolutional neural network and error correction for indoor positioning," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 2, p. 74, Jan. 2020.

[13] S. B. A. Khattak, Fawad, M. M. Nasralla, M. A. Esmail, H. Mostafa, and M. Jia, "WLAN RSS-based fingerprinting for indoor localization: A machine learning inspired bag-of-features approach," *Sensors*, vol. 22, no. 14, p. 5236, Jul. 2022.

[14] G. Zhao, "Deep-learning approach for indoor image-based visible light positioning," M.S. thesis, Nanyang Technol. Univ., Singapore, 2022. [Online]. Available: https://hdl.handle.net/10356/161620

[15] A. Raza, L. Lolic, S. Akhter, and M. Liut, "Comparing and evaluating indoor positioning techniques," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Nov. 2021, pp. 1–8.

[16] R. Xia, K. Jiang, X. Wang, and Z. Zhan, "Structural line feature selection for improving indoor visual SLAM," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 327–334, Jun. 2022.

[17] C. Debeunne and D. Vivet, "A review of visual-LiDAR fusion based simultaneous localization and mapping," *Sensors*, vol. 20, no. 7, p. 2068, 2020.

[18] Y. Wu, F. Tang, and H. Li, "Image-based camera localization: An overview," *Vis. Comput. Ind., Biomed., Art*, vol. 1, no. 1, pp. 1–13, Dec. 2018.
[19] T. B. Bach, T. T. Dinh, and J.-H. Lee, "FeatLoc: Absolute pose regressor for indoor 2D sparse features with simplistic view synthesizing," *ISPRS J. Photogramm. Remote Sens.*, vol. 189, pp. 50–62, Jul. 2022.
[20] Y. Shavit and R. Ferens, "Introduction to camera pose estimation with deep learning," 2019, *arXiv:1907.05272*.
[21] N. Piasco, D. Sidibé, C. Demonceaux, and V. Gouet-Brunet, "A survey on visual-based localization: On the benefit of heterogeneous data," *Pattern Recognit.*, vol. 74, pp. 90–109, Feb. 2018.
[22] A. Ben-Afia, L. Deambrogio, D. Salos, A. Escher, C. Macabiau, L. Soulier, and V. Gay-Bellile, "Review and classification of vision-based localisation techniques in unknown environments," *IET Radar, Sonar Navigat.*, vol. 8, no. 9, pp. 1059–1072, Dec. 2014.
[23] R. Raman, S. Bakshi, and P. K. Sa, "Multi-camera localisation: A review," *Int. J. Mach. Intell. Sensory Signal Process.*, vol. 1, no. 1, pp. 91–109, 2013.
[24] X. Xin, J. Jiang, and Y. Zou, "A review of visual-based localization," in *Proc. Int. Conf. Robot., Intell. Control Artif. Intell.*, Sep. 2019, pp. 94–105.
[25] M. Xu, Y. Wang, B. Xu, J. Zhang, J. Ren, S. Poslad, and P. Xu, "A critical analysis of image-based camera pose estimation techniques," 2022, *arXiv:2201.05816*. Accessed: Mar. 11, 2022.
[26] M. R. Razali, A. A. M. Faudzi, and A. U. Shamsudin, "Visual simultaneous localization and mapping: A review," *PERINTIS eJ.*, vol. 12, no. 1, pp. 23–34, 2022, doi: 10.1007/s10462-012-9365-8.
[27] A. Morar, A. Moldoveanu, I. Mocanu, F. Moldoveanu, I. E. Radoi, V. Asavei, A. Gradinaru, and A. Butean, "A comprehensive survey of indoor localization methods based on computer vision," *Sensors*, vol. 20, no. 9, p. 2641, May 2020, doi: 10.3390/s20092641.
[28] A. Yassin et al., "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1327–1346, 2nd Quart., 2016.
[29] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.
[30] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 15–22.
[31] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[32] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.
[33] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, Zurich, Switzerland: Springer, Sep. 2014, pp. 834–849.
[34] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[35] J. Yu, F. Gao, J. Cao, C. Yu, Z. Zhang, Z. Huang, Y. Wang, and H. Yang, "CNN-based monocular decentralized SLAM on embedded FPGA," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops (IPDPSW)*, May 2020, pp. 66–73.
[36] M. Abouzahir, A. Elouardi, R. Latif, S. Bouaziz, and A. Tajer, "Embedding SLAM algorithms: Has it come of age?" *Robot. Auton. Syst.*, vol. 100, pp. 14–26, Feb. 2018.
[37] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.
[38] T. J. J. Ryan, "LSTMs explained: A complete, technically accurate, conceptual guide with keras," Anal. Vidhya, Sep. 2020. [Online]. Available: https://medium.com/analytics-vidhya/lstms-explained-a-complete-technically-accurate-conceptual-guide-with-keras-2a650327e8f2
[39] G. Yigit and M. F. Amasyali, "Simple but effective GRU variants," in *Proc. Int. Conf. Innov. Intell. Syst. Appl. (INISTA)*, Aug. 2021, pp. 1–6.
[40] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.
[41] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
[42] Y.-H. Li, L. N. Harfiya, K. Purwandari, and Y.-D. Lin, "Real-time cuffless continuous blood pressure estimation using deep learning model," *Sensors*, vol. 20, no. 19, p. 5606, Sep. 2020.
[43] S. Kim, I. Kim, L. F. Vecchietti, and D. Har, "Pose estimation utilizing a gated recurrent unit network for visual localization," *Appl. Sci.*, vol. 10, no. 24, p. 8876, Dec. 2020.
[44] S. Dobilas, "LSTM recurrent neural networks—How to teach a network to remember the past," Medium, Towards Data Sci., Tech. Rep. Accessed: Jun. 28, 2022. [Online]. Available: https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c2ff22e
[45] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *Proc. CVPR*, Jun. 2011, pp. 2969–2976.
[46] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2599–2606.
[47] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 994–1001, Apr. 2018.
[48] Y. Kuang and K. Astrom, "Pose estimation with unknown focal length using points, directions and lines," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 529–536.
[49] Z. Kukelova, M. Bujnak, and T. Pajdla, "Real-time solution to the absolute pose problem with unknown radial distortion and focal length," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2816–2823.
[50] J. Ventura, C. Arth, G. Reitmayr, and D. Schmalstieg, "A minimal solution to the generalized pose-and-scale problem," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 422–429.
[51] Y. Zheng, S. Sugimoto, I. Sato, and M. Okutomi, "A general and simple method for camera pose and focal length determination," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 430–437.
[52] B. Wattanacheep and O. Chitsobhuk, "Camera pose estimation using CNN," in *Proc. 3rd Int. Conf. Control Comput. Vis.*, Aug. 2020, pp. 84–88.
[53] A. Moreau, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle, "LENS: Localization enhanced by NeRF synthesis," in *Proc. Conf. Robot Learn.*, 2022, pp. 1347–1356.
[54] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," 2015, *arXiv:1505.07427*.
[55] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014, *arXiv:1409.4842*.
[56] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," 2015, *arXiv:1509.05909*.
[57] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using LSTMs for structured feature correlation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 627–637, doi: 10.1109/ICCV.2017.75.
[58] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Image-based localization using hourglass networks," 2017, *arXiv:1703.07971*.
[59] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala, "Camera relocalization by computing pairwise relative poses using convolutional neural network," 2017, *arXiv:1707.09733*.
[60] B. Wang, C. Chen, C. X. Lu, P. Zhao, N. Trigoni, and A. Markham. (2020). *AtLoc: Attention Guided Camera Localization*. [Online]. Available: https://github.com/BingCS/AtLoc
[61] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen, "VidLoc: A deep spatio-temporal model for 6-DoF video-clip relocalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6856–6864. Accessed: Mar. 9, 2022. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2017/html/
[62] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8174–8182.
[63] H. Q. Tran, T. V. Nguyen, T. V. Huynh, and N. Q. Tran, "Improving accuracy of indoor localization system using ensemble learning," *Syst. Sci. Control Eng.*, vol. 10, no. 1, pp. 645–652, Dec. 2022.

[64] M. Rad, M. Oberweger, and V. Lepetit, "Feature mapping for learning fast and accurate 3D pose inference from synthetic images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4663–4672. Accessed: Mar. 9, 2022. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/

[65] I. Ha, H. Kim, S. Park, and H. Kim, "Image retrieval using BIM and features from pretrained VGG network for indoor localization," *Building Environ.*, vol. 140, pp. 23–31, Aug. 2018, doi: 10.1016/j.buildenv.2018.05.026.

[66] D. Acharya, K. Khoshelham, and S. Winter, "BIM-PoseNet: Indoor camera localisation using a 3D indoor model and deep learning from synthetic images," *ISPRS J. Photogramm. Remote Sens.*, vol. 150, pp. 245–258, Apr. 2019. Accessed: Mar. 9, 2022. [Online]. Available: https://www.sciencedirect.com/

[67] J. Chen, S. Li, and W. Lu, "Align to locate: Registering photogrammetric point clouds to BIM for robust indoor localization," *Building Environ.*, vol. 209, Feb. 2022, Art. no. 108675.

[68] D. Acharya, S. S. Roy, K. Khoshelham, and S. Winter, "A recurrent deep network for estimating the pose of real indoor images from synthetic image sequences," *Sensors*, vol. 20, no. 19, p. 5492, 2020. Accessed: Mar. 9, 2022. [Online]. Available: https://www.mdpi.com/838776

[69] Q. Li, R. Cao, J. Zhu, X. Hou, J. Liu, S. Jia, Q. Li, and G. Qiu, "Improving synthetic 3D model-aided indoor image localization via domain adaptation," *ISPRS J. Photogramm. Remote Sens.*, vol. 183, pp. 66–78, Jan. 2022, doi: 10.1016/j.isprsjprs.2021.10.005.

[70] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Perez, "ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2517–2526. Accessed: Mar. 9, 2022. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/

[71] P. P. Busto and J. Gall, "Open set domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 754–763.

[72] W.-L. Chang, H.-P. Wang, W.-H. Peng, and W.-C. Chiu, "All about structure: Adapting structural information across domains for boosting semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1900–1909. Accessed: Mar. 9, 2022. [Online]. Available: http://openaccess.thecvf.com/

[73] Y. Zou, Z. Yu, B. V. K. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 289–305.

[74] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7472–7481.

[75] S. Baik, H. J. Kim, T. Shen, E. Ilg, K. M. Lee, and C. Sweeney, "Domain adaptation of learned features for visual localization," 2020, *arXiv:2008.09310*.

[76] H. Hu, Z. Qiao, M. Cheng, Z. Liu, and H. Wang, "DASGIL: Domain adaptation for semantic and geometric-aware image-based localization," *IEEE Trans. Image Process.*, vol. 30, pp. 1342–1353, 2020. Accessed: Mar. 9, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9296559/

[77] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2930–2937. Accessed: Mar. 9, 2022. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2013/html/

[78] T. Cavallari, S. Golodetz, N. A. Lord, J. Valentin, L. D. Stefano, and P. H. S. Torr, "On-the-fly adaptation of regression forests for online camera relocalisation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4457–4466. Accessed: Mar. 9, 2022. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2017/html/

[79] Y. Ding, D. Jiang, Y. Yang, Y. Liu, T. He, and D. Zhang, "P2-loc: A person-2-person indoor localization system in on-demand delivery," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 6, no. 1, pp. 1–24, Mar. 2022.

[80] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson, "City-scale localization for cameras with known vertical direction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1455–1461, Jul. 2017. Accessed: Mar. 10, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7534854/

[81] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "DSAC—Differentiable RANSAC for camera localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6684–6692.

[82] E. Brachmann and C. Rother, "Learning less is more–6D camera localization via 3D surface regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4654–4662. Accessed: Mar. 9, 2022. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/

[83] J. Wu, L. Ma, and X. Hu, "Delving deeper into convolutional neural networks for camera relocalization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5644–5651. Accessed: Mar. 10, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7989663/

[84] T. Xie, K. Dai, K. Wang, R. Li, J. Wang, X. Tang, and L. Zhao, "A deep feature aggregation network for accurate indoor camera localization," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3687–3694, Apr. 2022.

[85] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[86] D. Robertsone and R. Cipolla, "An image-based system for urban navigation," in *Proc. Brit. Mach. Vis. Conf.*, 2004, pp. 1–10. Accessed: Mar. 10, 2022. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.422.1523&rep=rep1&type=pdf

[87] Y. Li, N. Snavely, and D. P. Huttenlocher, "Location recognition using prioritized feature matching," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 791–804.

[88] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2D-to-3D matching," in *Proc. Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 667–674, doi: 10.1109/ICCV.2011.6126302.

[89] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys, "Hyperpoints and fine vocabularies for large-scale location recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2102–2110.

[90] T. Sattler, B. Leibe, and L. Kobbelt, "Improving image-based localization by active correspondence search," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7572, 2012, pp. 752–765, doi: 10.1007/978-3-642-33718-5_54.

[91] L. Liu, H. Li, and Y. Dai, "Efficient global 2D-3D matching for camera localization in a large-scale 3D map," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2372–2381. Accessed: Mar. 10, 2022. [Online]. Available: http://openaccess.thecvf.com/content_iccv_2017/html/

[92] L. Svarm, O. Enqvist, M. Oskarsson, and F. Kahl, "Accurate localization and pose estimation for large 3D models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 532–539. Accessed: Mar. 10, 2022. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2014/html/

[93] B. Zeisl, T. Sattler, and M. Pollefeys, "Camera pose voting for large-scale image-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2704–2712.

[94] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.

[95] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[96] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3DMatch: Learning local geometric descriptors from RGB-D reconstructions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1802–1811.

[97] J. Ding, Y. Wang, H. Si, S. Gao, and J. Xing, "Three-dimensional indoor localization and tracking for mobile target based on WiFi sensing," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21687–21701, Nov. 2022.

[98] S. Song and J. Xiao, "Deep sliding shapes for amodal 3D object detection in RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 808–816.

[99] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12716–12725. Accessed: Mar. 10, 2022. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/

[100] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 647–655. Accessed: Mar. 10, 2022. [Online]. Available: https://proceedings.mlr.press/v32/donahue14.html

[101] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11065–11074. Accessed: Mar. 10, 2022. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Dai_Second-Order_Attention_Network_for_Single_Image_Super-Resolution_CVPR_2019_paper.html

[102] M. Teichmann, A. Araujo, M. Zhu, and J. Sim, "Detect-to-retrieve: Efficient regional aggregation for image search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5109–5118. Accessed: Mar. 10, 2022. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Teichmann_Detect-To-Retrieve_Efficient_Regional_Aggregation_for_Image_Search_CVPR_2019_paper.html

[103] S. S. Husain and M. Bober, "REMAP: Multi-layer entropy-guided pooling of dense CNN features for image retrieval," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5201–5213, Oct. 2019. Accessed: Mar. 10, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8720226/

[104] J. M. Joyce, "Kullback–Leibler divergence," in *International Encyclopedia of Statistical Science*. Berlin, Germany: Springer, 2011, pp. 720–722.

[105] N. Li, W. Tu, and H. Ai, "A sparse feature matching model using a transformer towards large-view indoor visual localization," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–12, Jul. 2022.

[106] X. Zhang, F. X. Yu, S. Karaman, and S.-F. Chang, "Learning discriminative and transformation covariant local feature detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6818–6826. Accessed: Mar. 10, 2022. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2017/html/Zhang_Learning_Discriminative_and_CVPR_2017_paper.html

[107] Y. Tian, V. Balntas, T. Ng, A. Barroso-Laguna, Y. Demiris, and K. Mikolajczyk, "D2D: Keypoint extraction with describe to detect approach," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, 2020, pp. 1–18. Accessed: Mar. 10, 2022. [Online]. Available: https://openaccess.thecvf.com/content/ACCV2020/html/Tian_D2D_Keypoint_Extraction_with_Describe_to_Detect_Approach_ACCV_2020_paper.html

[108] I. Melekhov, A. Tiulpin, T. Sattler, M. Pollefeys, E. Rahtu, and J. Kannala, "DGC-Net: Dense geometric correspondence network," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1034–1042. Accessed: Mar. 10, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8658868/

[109] O. Wiles, S. Ehrhardt, and A. Zisserman. (2020). *D2D: Learning to Find Good Correspondences for Image Matching and Manipulation*. Accessed: Mar. 10, 2022. [Online]. Available: https://openreview.net/forum?id=1B4FQXZIttM

[110] Z. Xu, Z. Jia, X. Zhou, H. Wen, and Y. Li, "An improved indoor navigation scheme based on vision-language localization," in *Proc. 33rd Chin. Control Decis. Conf. (CCDC)*, May 2021, pp. 1047–1051.

[111] L. Ruotsalainen, A. Morrison, M. Mäkelä, J. Rantanen, and N. Sokolova, "Improving computer vision-based perception for collaborative indoor navigation," *IEEE Sensors J.*, vol. 22, no. 6, pp. 4816–4826, Mar. 2022.

[112] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 3951, 2006, pp. 430–443, doi: 10.1007/11744023_34.

[113] M. Cai, H. Zhan, C. S. Weerasekera, K. Li, and I. Reid, "Camera relocalization by exploiting multi-view constraints for scene coordinates regression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Seoul, South Korea, 2019, pp. 3769–3777, doi: 10.1109/ICCVW.2019.00469.

[114] F. Xue, X. Wu, S. Cai, and J. Wang, "Learning multi-view camera relocalization with graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11372–11381.

[115] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother, "Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3364–3372. Accessed: Mar. 10, 2022. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2016/html/Brachmann_Uncertainty-Driven_6D_Pose_CVPR_2016_paper.html

[116] I. Budvytis, M. Teichmann, T. Vojir, and R. Cipolla, "Large scale joint semantic re-localisation and scene understanding via globally unique instance coordinate regression," 2019, *arXiv:1909.10239*. Accessed: Mar. 10, 2022.

[117] L. Yang, Z. Bai, C. Tang, H. Li, Y. Furukawa, and P. Tan, "SANet: Scene agnostic network for camera localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 42–51. Accessed: Mar. 10, 2022. [Online]. Available: http://openaccess.thecvf.com/content_ICCV_2019/html/Yang_SANet_Scene_Agnostic_Network_for_Camera_Localization_ICCV_2019_paper.html

[118] T. Ng, A. Lopez-Rodriguez, V. Balntas, and K. Mikolajczyk, "Reassessing the limitations of CNN methods for camera pose regression," 2021, *arXiv:2108.07260*.

[119] M. A. Musallam, V. Gaudilliere, M. O. del Castillo, K. Al Ismaeil, and D. Aouada, "Leveraging equivariant features for absolute pose regression," 2022, *arXiv:2204.02163*.

[120] X. Li, J. Ylioinas, J. Verbeek, and J. Kannala, "Scene coordinate regression with angle-based reprojection loss for camera relocalization," in *Computer Vision—ECCV 2018 Workshops* (Lecture Notes in Computer Science), vol. 11131, L. Leal-Taixé and S. Roth, Eds. Cham, Switzerland: Springer, 2019, pp. 229–245, doi: 10.1007/978-3-030-11015-4_19.

[121] E. Brachmann and C. Rother, "Visual camera re-localization from RGB and RGB-D images using DSAC," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5847–5865, Sep. 2022. Accessed: Mar. 10, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9394752/

[122] W. Zhou, X. Hao, K. Wang, Z. Zhang, Y. Yu, H. Su, K. Li, X. Cao, and A. Kuijper, "Improved estimation of motion blur parameters for restoration from a single image," *PLoS ONE*, vol. 15, no. 9, Sep. 2020, Art. no. e0238259.

[123] C. M. Parameshwara, G. Hari, C. Fermuller, N. J. Sanket, and Y. Aloimonos, "DiffPoseNet: Direct differentiable camera pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6845–6854.

[124] M.-A. Chung and C.-W. Lin, "An improved localization of mobile robotic system based on AMCL algorithm," *IEEE Sensors J.*, vol. 22, no. 1, pp. 900–908, Jan. 2022.

[125] I. R. Ward, M. A. Jalwana, and M. Bennamoun, "Improving image-based localization with deep learning: The impact of the loss function," in *Proc. Pacific-Rim Symp. Image Video Technol.* Cham, Switzerland: Springer, 2019, pp. 111–124.

[126] A. A. Mahdi, A. Chalechale, and A. AbdelRaouf, "A hybrid indoor positioning model for critical situations based on localization technologies," *Mobile Inf. Syst.*, vol. 2022, pp. 1–15, Apr. 2022.

[127] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla, "Benchmarking 6DOF outdoor visual localization in changing conditions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8601–8610.

[128] T. Naseer and W. Burgard, "Deep regression for monocular camera-based 6-DoF global localization in outdoor environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1525–1530. Accessed: Mar. 10, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8205957/

[129] S. Zangeneh, S. Pruett, S. Lym, and Y. N. Patt, "BranchNet: A convolutional neural network to predict hard-to-predict branches," in *Proc. 53rd Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Oct. 2020, pp. 118–130.

[130] 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.

[131] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5974–5983. Accessed: Mar. 10, 2022. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2017/html/Kendall_Geometric_Loss_Functions_CVPR_2017_paper.html

[132] S. Chen, Z. Wang, and V. Prisacariu, "Direct-PoseNet: Absolute pose regression with photometric consistency," in *Proc. Int. Conf. 3D Vis. (3DV)*, Dec. 2021, pp. 1175–1185.

[133] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, and T. Sattler, "Back to the feature: Learning robust camera localization from pixels to pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3247–3257.

[134] Y. Zhan, F. Li, R. Weng, and W. Choi, "Ray3D: Ray-based 3D human pose estimation for monocular absolute 3D localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13116–13125.

[135] Y. Shavit, R. Ferens, and Y. Keller, "Learning multi-scene absolute pose regression with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2733–2742.

[136] Y. Wan, W. Gao, S. Han, and Y. Wu, "Boosting image-based localization via randomly geometric data augmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 688–692.

[137] M. Xu, L. Wang, J. Ren, and S. Poslad, "Use of LSTM regression and rotation classification to improve camera pose localization estimation," in *Proc. IEEE 14th Int. Conf. Anti-Counterfeiting, Secur., Identificat. (ASID)*, Oct. 2020, pp. 6–10.

[138] M. Bui, C. Baur, N. Navab, S. Ilic, and S. Albarqouni, "Adversarial networks for camera pose regression and refinement," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Seoul, South Korea, 2019, pp. 3778–3787, doi: 10.1109/ICCVW.2019.00470.

[139] B. Chidlovskii and A. Sadek, "Adversarial transfer of pose estimation regression," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 646–661.

[140] Z. Huang, Y. Xu, J. Shi, X. Zhou, H. Bao, and G. Zhang, "Prior guided dropout for robust visual localization in dynamic environments," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2791–2800.

[141] S. Chen, X. Li, Z. Wang, and V. A. Prisacariu, "DFNet: Enhance absolute pose regression with direct feature matching," 2022, *arXiv:2204.00559*.

[142] M. Cai, C. Shen, and I. Reid. (2019). *A Hybrid Probabilistic Model for Camera Relocalization*. Accessed: Mar. 10, 2022. [Online]. Available: https://digital.library.adelaide.edu.au/dspace/handle/2440/124684

[143] P. Purkait, C. Zhao, and C. Zach, "Synthetic view generation for absolute pose regression and image synthesis," in *Proc. BMVC*, 2018, p. 69.

[144] L. Jin, C. Xu, X. Wang, Y. Xiao, Y. Guo, X. Nie, and J. Zhao, "Single-stage is enough: Multi-person absolute 3D pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13086–13095.

[145] X. Li and H. Ling, "PoGO-Net: Pose graph optimization with graph neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5895–5905.

[146] J. K. Duggal, "Design space exploration of DNNs for autonomous systems," Graduate School, Purdue Univ., West Lafayette, IN, USA, Tech. Rep., Oct. 2019., doi: 10.25394/PGS.8980463.v1.

[147] X. Li, "Image-based localization using deep neural networks," AEE-Master's Programme Automat. Elect. Eng. (TS2013), Aalto Univ., Finland, Tech. Rep. Aalto 9687, 2017. [Online]. Available: https://aaltodoc.aalto.fi/handle/123456789/28561

[148] H. Blanton, C. Greenwell, S. Workman, and N. Jacobs, "Extending absolute pose regression to multiple scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 38–39.

[149] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2616–2625. Accessed: Mar. 10, 2022. [Online]. Available: https://openreview.net/forum?id=lTawICyy5HP

[150] B. Gao, B. Lian, D. Wang, and C. Tang, "Low drift visual inertial odometry with UWB aided for indoor localization," *IET Commun.*, vol. 16, no. 10, pp. 1083–1093, Jun. 2022.

[151] Y. Lin, Z. Liu, J. Huang, C. Wang, G. Du, J. Bai, and S. Lian, "Deep global-relative networks for end-to-end 6-DoF visual localization and odometry," in *Proc. Pacific Rim Int. Conf. Artif. Intell.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 11671, 2019, pp. 454–467, doi: 10.1007/978-3-030-29911-8_35.

[152] J. Zhen, Q. Fang, J. Sun, W. Liu, W. Jiang, H. Bao, and X. Zhou, "SMAP: Single-shot multi-person absolute 3D pose estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 550–566.

[153] A. El Kaid, D. Brazey, V. Barra, and K. Baïna, "Top-down system for multi-person 3D absolute pose estimation from monocular videos," *Sensors*, vol. 22, no. 11, p. 4109, May 2022.

[154] F. Ott, T. Feigl, C. Loffler, and C. Mutschler, "ViPR: Visual-odometry-aided pose regression for 6DoF camera localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 42–43.

[155] Y. Liu, C. Zhao, and Y. Wei, "A robust localization system fusion vision-CNN relocalization and progressive scan matching for indoor mobile robots," *Appl. Sci.*, vol. 12, no. 6, p. 3007, Mar. 2022.

[156] N. Kayhani, W. Zhao, B. McCabe, and A. P. Schoellig, "Tag-based visual-inertial localization of unmanned aerial vehicles in indoor construction environments using an on-manifold extended Kalman filter," *Autom. Construct.*, vol. 135, Mar. 2022, Art. no. 104112.

[157] Y.-J. Kim and M. Chi, "Temporal belief memory: Imputing missing data during RNN training," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 1–7.

[158] Z. Cui, L. Pan, and S. Liu, "Hybrid BiLSTM-Siamese network for relation extraction," in *Proc. 18th Int. Conf. Auton. Agents MultiAgent Syst.*, 2019, pp. 1907–1909.

[159] M. Li, J. Qin, D. Li, R. Chen, X. Liao, and B. Guo, "VNLSTM-PoseNet: A novel deep ConvNet for real-time 6-DOF camera relocalization in urban streets," *Geo-Spatial Inf. Sci.*, vol. 24, no. 3, pp. 422–437, Jul. 2021.

[160] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2616–2625.

[161] H. Li, J. Zhao, J.-C. Bazin, W. Chen, K. Chen, and Y.-H. Liu, "Line-based absolute and relative camera pose estimation in structured environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6914–6920.

[162] Z. Yang, J. Z. Pan, L. Luo, X. Zhou, K. Grauman, and Q. Huang, "Extreme relative pose estimation for RGB-D scans via scene completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4531–4540.

[163] C. Yang, Y. Liu, and A. Zell, "RCPNet: Deep-learning based relative camera pose estimation for UAVs," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Sep. 2020, pp. 1085–1092.

[164] S. Zhang, D. Jin, Y. Dai, and F. Yang, "Relative pose estimation for light field cameras based on LF-point-LF-point correspondence model," *IEEE Trans. Image Process.*, vol. 31, pp. 1641–1656, 2022.

[165] M. Humenberger, Y. Cabon, N. Pion, P. Weinzaepfel, D. Lee, N. Guérin, T. Sattler, and G. Csurka, "Investigating the role of image retrieval for visual localization," *Int. J. Comput. Vis.*, vol. 130, pp. 1–26, May 2022.

[166] B. Guan, J. Zhao, Z. Li, F. Sun, and F. Fraundorfer, "Relative pose estimation with a single affine correspondence," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10111–10122, Oct. 2022.

[167] V. Balntas, S. Li, and V. Prisacariu, "RelocNet: Continuous metric learning relocalisation using neural nets," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 751–767. Accessed: Mar. 10, 2022. [Online]. Available: http://openaccess.thecvf.com/content_ECCV_2018/html/Vassileios_Balntas_RelocNet_Continous_Metric_ECCV_2018_paper.html

[168] M. Ding, Z. Wang, J. Sun, J. Shi, and P. Luo, "CamNet: Coarse-to-fine retrieval for camera re-localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2871–2880. Accessed: Mar. 10, 2022. [Online]. Available: http://openaccess.thecvf.com/content_ICCV_2019/html/Ding_CamNet_Coarse-to-Fine_Retrieval_for_Camera_Re-Localization_ICCV_2019_paper.html

[169] Q. Zhou, T. Sattler, M. Pollefeys, and L. Leal-Taixe, "To learn or not to learn: Visual localization from essential matrices," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3319–3326. Accessed: Mar. 10, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9196607/

[170] H. Fan, J. Kileel, and B. Kimia, "On the instability of relative pose estimation and RANSAC's role," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8935–8943.

[171] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Relative camera pose estimation using convolutional neural networks," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.* Cham, Switzerland: Springer, 2017, pp. 675–687.

[172] H. Aanæs, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl, "Large-scale data for multiple-view stereopsis," *Int. J. Comput. Vis.*, vol. 120, no. 2, pp. 153–168, 2016.

[173] S. Saha, G. Varma, and C. V. Jawahar, "Improved visual relocalization by discovering anchor points," in *Proc. Brit. Mach. Vis. Conf.*, 2019, pp. 1–11.

[174] H. Lim, J. Jeon, and H. Myung, "UV-SLAM: Unconstrained line-based SLAM using vanishing points for structural mapping," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1518–1525, Apr. 2022.

[175] O. Seiskari, P. Rantalankila, J. Kannala, J. Ylilammi, E. Rahtu, and A. Solin, "HybVIO: Pushing the limits of real-time visual-inertial odometry," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 701–710.
[176] J. Neira, A. J. Davison, and J. J. Leonard, "Guest editorial special issue on visual SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 929–931, Oct. 2008.
[177] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.
[178] B. Vincke, A. Elouardi, A. Lambert, and A. Merigot, "Efficient implementation of EKF-SLAM on a multi-core embedded system," in *Proc. 38th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2012, pp. 3049–3054.
[179] A. A. J. Serrata, S. Yang, and R. Li, "An intelligible implementation of FastSLAM2.0 on a low-power embedded architecture," *EURASIP J. Embedded Syst.*, vol. 2017, no. 1, pp. 1–11, Dec. 2017.
[180] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2320–2327.
[181] K. Boikos and C.-S. Bouganis, "Semi-dense SLAM on an FPGA SoC," in *Proc. 26th Int. Conf. Field Program. Log. Appl. (FPL)*, Aug. 2016, pp. 1–4.
[182] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2198–2204.
[183] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2502–2509.
[184] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.
[185] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, and R. Siegwart, "A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 431–437.
[186] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 298–304.
[187] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2510–2517.
[188] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 796–803, Apr. 2017.
[189] O. C. B. Silveira, J. G. O. C. de Melo, L. A. S. Moreira, J. B. N. G. Pinto, L. R. L. Rodrigues, and P. F. F. Rosa, "Evaluating a visual simultaneous localization and mapping solution on embedded platforms," in *Proc. IEEE 29th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2020, pp. 530–535.
[190] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.
[191] M. K. Paul, K. Wu, J. A. Hesch, E. D. Nerurkar, and S. I. Roumeliotis, "A comparative analysis of tightly-coupled monocular, binocular, and stereo VINS," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 165–172.
[192] C. Campos, J. M. M. Montiel, and J. D. Tardós, "Inertial-only optimization for visual-inertial initialization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 51–57.
[193] A. Merzlyakov and S. Macenski, "A comparison of modern general-purpose visual SLAM approaches," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 9190–9197.
[194] B. Canovas, M. Rombaut, A. Nègre, D. Pellerin, and S. Olympieff, "Speed and memory efficient dense RGB-D SLAM in dynamic scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 4996–5001.
[195] R. A. Newcombe, A. Fitzgibbon, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, and S. Hodges, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, Oct. 2011, pp. 127–136.
[196] Q. Jin, Y. Liu, Y. Man, and F. Li, "Visual SLAM with RGB-D cameras," in *Proc. Chin. Control Conf. (CCC)*, Jul. 2019, pp. 4072–4077.
[197] L. Nardi, B. Bodin, M. Z. Zia, J. Mawer, A. Nisbet, P. H. J. Kelly, A. J. Davison, M. Lujan, M. F. P. O'Boyle, G. Riley, N. Topham, and S. Furber, "Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 5783–5790.
[198] B. Bodin, L. Nardi, M. Z. Zia, H. Wagstaff, G. S. Shenoy, M. Emani, J. Mawer, C. Kotselidis, A. Nisbet, M. Lujan, B. Franke, P. H. J. Kelly, and M. O'Boyle, "Integrating algorithmic parameters into benchmarking and design space exploration in 3D scene understanding," in *Proc. Int. Conf. Parallel Archit. Compilation*, Sep. 2016, pp. 57–69.
[199] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 2100–2106.
[200] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1352–1359.
[201] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 177–187, Feb. 2014.
[202] O. Araar, S. Bouhired, S. Moussiou, and A. Laggoune, "Towards low-cost indoor localisation using a multi-camera system," in *Proc. 19th Int. Conf. Comput. Sci. (ICCS)*, Faro, Portugal, Cham, Switzerland: Springer, Jun. 2019, pp. 137–148.
[203] O. F. Ince and J.-S. Kim, "TIMA SLAM: Tracking independently and mapping altogether for an uncalibrated multi-camera system," *Sensors*, vol. 21, no. 2, p. 409, Jan. 2021.
[204] M. Sewtz, X. Luo, J. Landgraf, T. Bodenmuller, and R. Triebel, "Robust approaches for localization on multi-camera systems in dynamic environments," in *Proc. 7th Int. Conf. Autom., Robot. Appl. (ICARA)*, Feb. 2021, pp. 211–215.
[205] H. Liu, H. Du, L. Wang, and X. Jin, "Indoor positioning system based on multi-camera joint calibration," in *Proc. 36th Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, May 2021, pp. 129–135.
[206] F.-A. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 734–746, Jun. 2019.
[207] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D object SLAM," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 925–938, Aug. 2019.
[208] C. Wang, B. Luo, Y. Zhang, Q. Zhao, L. Yin, W. Wang, X. Su, Y. Wang, and C. Li, "DymSLAM: 4D dynamic scene reconstruction based on geometrical motion segmentation," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 550–557, Apr. 2021.
[209] X. Dong, L. Cheng, H. Peng, and T. Li, "FSD-SLAM: A fast semi-direct SLAM algorithm," *Complex Intell. Syst.*, vol. 8, pp. 1–12, Mar. 2021.
[210] J. Wang, M. Runz, and L. Agapito, "DSP-SLAM: Object oriented SLAM with deep shape priors," in *Proc. Int. Conf. 3D Vis. (3DV)*, Dec. 2021, pp. 1362–1371.
[211] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, "Real-time RGB-D camera relocalization," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Oct. 2013, pp. 173–179.
[212] W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang, and S. Leutenegger, "InteriorNet: Mega-scale multi-sensor photo-realistic indoor scenes dataset," 2018, *arXiv:1809.00716*.
[213] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. 24th Annu. ACM Symp. User interface Softw. Technol.*, Oct. 2011, pp. 559–568.
[214] E. Bayraktar, C. B. Yigit, and P. Boyraz, "A hybrid image dataset toward bridging the gap between real and simulation environments for robotics," *Mach. Vis. Appl.*, vol. 30, no. 1, pp. 23–40, Feb. 2019, doi: 10.1007/s00138-018-0966-3.
[215] M. Z. Karakusak, H. Kivrak, H. F. Ates, and M. K. Ozdemir, "RSS-based wireless LAN indoor localization and tracking using deep architectures," *Big Data Cognit. Comput.*, vol. 6, no. 3, p. 84, Aug. 2022.
[216] N. Radwan, A. Valada, and W. Burgard, "VLocNet++: Deep multitask learning for semantic visual localization and odometry," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 4407–4414, Oct. 2018. Accessed: Mar. 10, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8458420/

[217] N. Li, W. Tu, H. Ai, H. Deng, J. Tao, T. Hu, and X. Sun, "VISEL: A visual and magnetic fusion-based large-scale indoor localization system with improved high-precision semantic maps," *Int. J. Intell. Syst.*, vol. 37, no. 10, pp. 7992–8020, Oct. 2022.
[218] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "SceneNet RGB-D: 5M photorealistic images of synthetic indoor trajectories with ground truth," 2016, *arXiv:1612.05079*.
[219] H. Q. Tran, T. V. Nguyen, T. V. Huynh, and N. Q. Tran, "Improving accuracy of indoor localization system using ensemble learning," *Syst. Sci. Control Eng.*, vol. 10, no. 1, pp. 645–652, Dec. 2022.
[220] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe, "Understanding the limitations of CNN-based absolute camera pose regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3302–3312.
[221] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "RGB-D SLAM dataset and benchmark," Comput. Vis. Group, TUM Dept. Inform. Tech., Univ. Munich, Munich, Germany, Tech. Rep., 2012.
[222] O. F. Yanik and H. A. Ilgin, "Comparison of power consumption of modern SLAM methods on various datasets," in *Proc. Int. Conf. Technol. Adv. Innov. (ICTAI)*, Nov. 2021, pp. 75–80.
[223] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 965–972, Apr. 2018.
[224] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stuckler, and D. Cremers, "The TUM VI benchmark for evaluating visual-inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1680–1687.
[225] A. Jaenal, D. Zuñiga-Nöel, R. Gomez-Ojeda, and J. Gonzalez-Jimenez, "Improving visual SLAM in car-navigated urban environments with appearance maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 4679–4685.
[226] Q. Xu, H. Kuang, L. Kneip, and S. Schwertfeger, "Rethinking the Fourier-Mellin transform: Multiple depths in the camera's view," *Remote Sens.*, vol. 13, no. 5, p. 1000, Mar. 2021.
[227] J. C. V. Soares, M. Gattass, and M. A. Meggiolaro, "Visual SLAM in human populated environments: Exploring the trade-off between accuracy and speed of Yolo and mask R-CNN," in *Proc. 19th Int. Conf. Adv. Robot. (ICAR)*, Dec. 2019, pp. 135–140.
[228] Z. Wan, B. Yu, T. Y. Li, J. Tang, Y. Zhu, Y. Wang, A. Raychowdhury, and S. Liu, "A survey of FPGA-based robotic computing," *IEEE Circuits Syst. Mag.*, vol. 21, no. 2, pp. 48–74, 2021.

MUHAMMAD SHAMSUL ALAM (Graduate Student Member, IEEE) received the B.Sc. degree from the Department of Computer Science and Engineering, Jahangirnagar University, Dhaka, Bangladesh, in 2003, and the M.Sc. degree in ICT from the Bangladesh University of Engineering and Technology (BUET), Dhaka, in 2007. He is currently pursuing the Ph.D. degree with the School of Computing, Universiti Teknologi Malaysia (UTM), Malaysia. He was with King Khalid University (KKU) and the University of Bisha (UB), under the Ministry of Education, Saudi Arabia. His current research interests include computer vision, machine learning, and deep learning.

FARHAN BIN MOHAMED (Senior Member, IEEE) received the B.Sc. degree in computer science from Universiti Teknologi Malaysia, in 2003, and the Ph.D. degree in computer science (data visualization) from Swansea University, in 2014. He was the Deputy Director of the Media and Game Innovation Centre of Excellence (MaGICX), Universiti Teknologi Malaysia. He was previously appointed as a Chief Technology Officer (CTO) of MaGICX Sdn. Bhd., from 2015 to 2017. He is currently the Director of emergent computing with the Faculty of Computing, Universiti Teknologi Malaysia. Currently, his work focuses on digital branding, gamification, data visualisation and augmented, and virtual reality. His research interests include visual analytics, visual metaphors, human–computer interactions, virtual environment, and procedural computer graphics. He is a Committee Member of the IEEE Computer Society Malaysia and a member of ACM MyHCI-UX Malaysia. He was a member of the 100 ACM Global Practitioner Advisory Committee, from 2017 to 2018. He was a fellow of CEO@faculty 2.0, from 2017 to 2018, attached to Huawei Malaysia. He received the Royal Academy of Engineering, U.K., Leaders in Innovation Fellowship, in 2017. He received the Excellence Student Award by Juita-UTM in his bachelor's degree. He is working on web performance optimization for his M.Sc. study, which the work has received the Gold Award from Malaysian Technology Expo 2006 and the Gold Award from the International Exhibition of Inventions, Geneva, in 2006.

ALI SELAMAT (Member, IEEE) has been the Dean of the Malaysia–Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia (UTM), Malaysia, since 2018. MJIIT is an academic institution established under the cooperation of the Japanese International Cooperation Agency (JICA) and the Ministry of Education Malaysia (MOE) to provide the Japanese style of education in Malaysia. He is currently a Full Professor with UTM. He is also a Professor with the Software Engineering Department, School of Computing, UTM, and the IEEE Computer Society Malaysia Section Chair. He has published more than 120 research articles with IF JCR, with more than 2400 citations received in the Web of Science and an H-index of 26. His research interests include software engineering, software process improvement, software agents, web engineering, information retrievals, pattern recognition, genetic algorithms, neural networks, soft computing, collective computational intelligence, strategic management, key performance indicator, and knowledge management. He is on the editorial board of the *Knowledge-Based Systems* (Elsevier).

AKM BELLAL HOSSAIN received the B.Sc. and B.C.S. degrees from the University of Pune, Pune, India, in 2000, and the Master of Science in Communication Engineering (MSCE) degree from United International University (UIU), Dhaka, Bangladesh, in 2009. He is currently pursuing the Ph.D. degree with the Faculty of Engineering, School of Computing, Universiti Teknologi Malaysia (UTM), Malaysia. Previously, he was with King Khalid University (KKU), the University of Bisha (recent), Saudi Arabia, and the Department of Information Systems, under the Ministry of Education. His current research interests include data augmentation, deep learning, and segmentation.