sensors

MDPI

Review

Visual SLAM for Unmanned Aerial Vehicles: Localization and Perception

Licong Zhuang [1], Xiaorong Zhong [1], Linjie Xu [2], Chunbao Tian [1] and Wenshuai Yu [2,*]

[1] Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Yutang Street, Guangming District, Shenzhen 518132, China; 2210273099@email.szu.edu.cn (L.Z.); zhongxiaorong@gml.ac.cn (X.Z.); tianchunbao@gml.ac.cn (C.T.)
[2] The College of Civil and Transportation Engineering, Shenzhen University, 3688 Nanhai Avenue, Nanshan District, Shenzhen 518060, China; 2210474138@email.szu.edu.cn
* Correspondence: ywsh@szu.edu.cn; Tel.: +86-1324-388-8102

Abstract: Localization and perception play an important role as the basis of autonomous Unmanned Aerial Vehicle (UAV) applications, providing the internal state of movements and the external understanding of environments. Simultaneous Localization And Mapping (SLAM), one of the critical techniques for localization and perception, is facing technical upgrading, due to the development of embedded hardware, multi-sensor technology, and artificial intelligence. This survey aims at the development of visual SLAM and the basis of UAV applications. The solutions to critical problems for visual SLAM are shown by reviewing state-of-the-art and newly presented algorithms, providing the research progression and direction in three essential aspects: real-time performance, texture-less environments, and dynamic environments. Visual–inertial fusion and learning-based enhancement are discussed for UAV localization and perception to illustrate their role in UAV applications. Subsequently, the trend of UAV localization and perception is shown. The algorithm components, camera configuration, and data processing methods are also introduced to give comprehensive preliminaries. In this paper, we provide coverage of visual SLAM and its related technologies over the past decade, with a specific focus on their applications in autonomous UAV applications. We summarize the current research, reveal potential problems, and outline future trends from academic and engineering perspectives.

Keywords: localization; perception; visual SLAM; UAV; odometry; feature extraction; visal–inertial SLAM; NeRF

Citation: Zhuang, L.; Zhong, X.; Xu, L.; Tian, C.; Yu, W. Visual SLAM for Unmanned Aerial Vehicles: Localization and Perception. *Sensors* **2024**, *24*, 2980. https://doi.org/10.3390/s24102980

Academic Editor: Udo Frese

Received: 28 March 2024
Revised: 1 May 2024
Accepted: 4 May 2024
Published: 8 May 2024

1. Introduction

UAVs have attracted much interest in their applications, such as unknown space perception, industrial defect inspection, and military operations, because of their flexibility, portability, and speed [1–3]. Localization and perception as the basis play an important role in those applications in which autonomous ability is needed. Internal states of movements and external understanding of environments are provided to enable UAV autonomous execution of missions. Localization determines whether the autonomous UAV moves accurately and acts precisely, while perception also supports basic movement in unknown space by detecting obstacles; moreover, a high-level understanding like semantic segmentation of environments enables more intelligent behavior, thereby enhancing the performance of autonomous UAVs and expanding the cover area of UAV applications. In particular, the above-mentioned applications have encountered the challenge of Global Navigation Satellite System (GNSS) denial where GNSS provides global localization capabilities. Therefore, an alternative approach to UAV localization and perception is needed, since it is the fundamental building block for autonomous upper missions such as navigation. The SLAM [4–6] technique has been widely researched for robot localization, and many excellent works have been presented. The SLAM technique is designed to

simultaneously estimate the state (position, orientation) of the robot body and construct a map of the surrounding unknown environment through the data collected by the sensors. This technique requires not the GNSS signal but onboard sensors such as camera, Light Detection And Ranging (LiDAR), and sonar, and it provides robust, real-time localization and perception for autonomous robots, including UAVs.

Earlier SLAM techniques were intended to use LiDAR or multiple sensors to achieve accurate and robust localization [7]. The sensor configurations, however, required extensive cost and computation resources. With the advances in computer vision, visual SLAM techniques [8–11] take cameras as the only sensor input and have gained much popularity because of the image sensor's low cost and simple configuration. Additionally, this sensor has great potential due to its ability to capture rich information about the surrounding environment. It is widely applied in lightweight devices such as smartphones, UAVs, and AR/VR equipment. This technique has a long history since [12] used the Kalman filter to estimate ego motion for the camera by extracting feature points in images in 1988. Nowadays, several well-designed and outstanding visual SLAM algorithms show incredible localization and mapping capability. To comprehensively understand the modern keyframe-based visual SLAM algorithm structure, we briefly introduce its workflow as following three main modules:

- Odometry: this is the basic module of the SLAM algorithm [13]. Its primary functionality is to process the latest received image by feature-based methods or direct methods, finding the correspondence between the current image and the reference image (or map). Once the correspondence is established, the camera pose can be estimated by epipolar geometry (2D–2D matches) [14], PnP (2D–3D matches) [15–17], or ICP (3D–3D matches) [18,19]. More recently, learning-based methods have been used for end-to-end estimating of the camera pose [20].
- Back end: this module maintains a global insistent map by performing Bundle Adjustment (BA) [21] for most state-of-the-art visual SLAM algorithms. On the one hand, the sliding window strategy is adopted, which keeps a fixed number of keyframes by marginalizing the old frame for controlling the BA cost in real time. On the other hand, a sparse map is constructed to optimize the motion and structure for more accurate results, since joint optimization with motion and dense map fail to run in real time.
- Loop closure: this module eliminates the accumulated error caused by large-scale, long-time estimation. To this end, a loop detection procedure is performed to detect the potential loop. Once a loop is detected, a lightweight pose graph optimization correlates with the trajectory, significantly improving the SLAM algorithm's accuracy. Notably, the precision rate of the loop detection is critical and must be ensured. Otherwise, the wrong detection could directly lead the algorithm to fail.

With these three modules, a general visual SLAM pipeline can be depicted as in Figure 1, the real-time camera egomotion can be estimated, and the sparse (or dense) map can be reconstructed.

Returning to UAV localization and perception, compared to another widely used sensor, LiDAR, here are some reasons why visual SLAM algorithms are more suitable:

- Low-cost sensor: Visual sensors are cheap and low-power. This is important for UAVs, which have relatively unstable control, poor loadability, and low power consumption. Unstable control results in a high damage rate and destruction of sensors; poor loadability and low power consumption mean the weight and power consumption are better to be lower. These factors make visual sensors popular in UAV applications.
- High frame rate: Visual sensors can capture images at a higher frame rate, enabling algorithms to provide more localization information. To utilize the flexibility of UAV, high-frame-rate odometry is necessary for precision of control.
- Capturing rich texture information: This is beneficial for UAV perception; rich texture information brings a high understanding of environments, subsequently enabling more intelligent missions for UAVs such as object tracking, semantic segmentation, and implicit reconstruction.

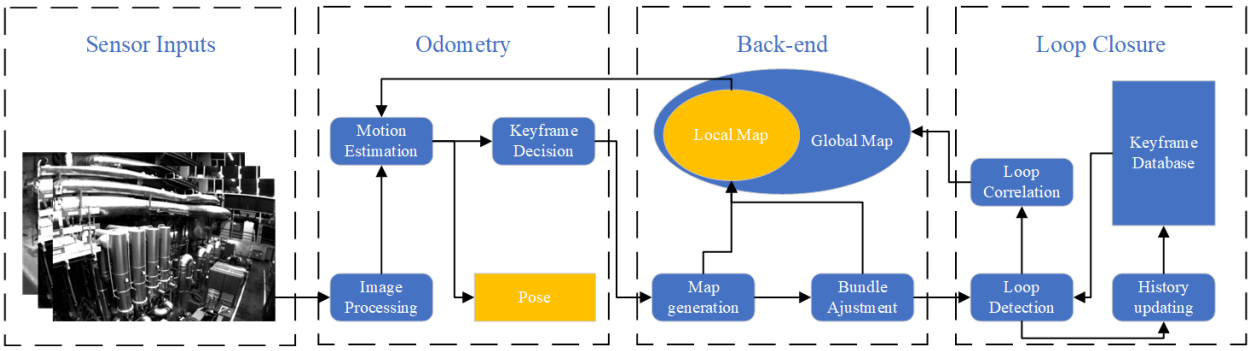


Figure 1. General keyframe-based visual SLAM algorithm pipeline (similar to [22–24]). 'Image Processing' includes image distortion, feature extraction, etc., and procedures for image matching to find correspondence; 'Keyframe Decision' considers whether the current map supports odometry to estimate the current state; 'Map generation' triangulates the pixels to 3D map points; 'History updating' updates the dataset by inserting new keyframes for potential loop detection.

Prior to our review, there have been excellent surveys on visual SLAM and related techniques, and it is necessary to summarize those surveys for comprehensive covering research: Ref. [25] reviews Visual Odometry (VO) and SLAM solutions for solving robot localization and mapping, introducing different types of VO methods and map representation, and EKF-based, particle filter-based, and RGB-D SLAM. It systematically explains algorithm, theory, and performance; Ref. [26] focuses on the types, approaches, challenges, and applications of VO. It shows the advantages of VO by comparing it with other sensor-based localization methods; Ref. [27] completely analyzes the SLAM problem and reveals SLAM's capability and challenges. Moreover, it summarizes the SLAM systems over the past 30 years and discusses the long-term autonomy of SLAM algorithms, representation of mapping, theoretical tools, active SLAM, unconventional sensors, and learning methods, providing a comprehensive review and open problems. With the great advance of visual SLAM algorithms from 2010 to 2016, Ref. [7] summarizes the state-of-the-art algorithms, from conventional feature-based methods and direct methods to RGB-D camera-based methods, within this period; Ref. [8] reviews visual SLAM algorithms through their flowchart, providing a clear understanding of the advantages and shortcomings of the main algorithms in three aspects (visual only, visual–inertial, and RGB-D camera-based); Ref. [9] provides an in-depth literature survey on visual SLAM algorithms in various aspects including multi-sensor, feature type, environment, resource constraint, and odometry method. Discussing the current trends of this technique, Ref. [10] reviews the state-of-the-art visual SLAM algorithms to date—specifically, sensors, feature extraction and matching methods, deep learning techniques, and major datasets are mentioned. However, in our survey, we provide coverage of visual SLAM and its related technologies over the past decade, with a specific focus on their applications in autonomous UAV applications.

This survey focuses on the visual SLAM algorithm applied in UAV applications and discusses its trend in UAV applications. The paper structure is described as follows: Section 2 introduces sensor configurations and data processing methods for some preliminaries. In Section 3, we review the development of visual SLAM (or odometry) algorithms by features: real-time performance, texture-less environment, and dynamic environment emphasizing state-of-the-art algorithms. Additionally, Section 4 introduces the widely applied SLAM algorithms in UAVs called visual–inertial SLAM and illustrates how it improves the robustness and accuracy of UAV localization. Finally, learning-based modules for UAV perception are discussed in Section 5, and a comprehensive conclusion is drawn in Section 6.

2. Camera Configuration and Data Processing

The visual SLAM algorithm takes visual sensors, which are low-cost and have great potential, as the input. It can be varied by different camera configurations and data

processing methods, which determine the algorithm inputs and lead to the successive modules changing. In this section, we introduce the different camera configurations and their properties, respectively, particularly for the feature-based method, the traditional feature extraction algorithms are reviewed, and at the end, the Inertial Measurement Unit (IMU) pre-integration method is briefly described to give the preliminary of the later section.

2.1. Camera Configuration

Different camera configurations significantly influence the performance and application scenes of visual SLAM algorithms. The advantages, shortcomings, and algorithms relative to sensor types are summarized in Table 1.

Table 1. Advantages, shortcomings, and algorithms relative to sensor types.

Type	Advantage	Shortcoming	Algorithms
Monocular	simple and cheap	suffers from scale ambiguity	[22,28,29]
Stereo	indoor and outdoor depth sensing	relies on texture of environment and requires computing	[23,30,31]
RGB-D	high-quality depth sensing	only suitable for indoor scenes and has high power consumption	[32–34]
Event	sensitive to dynamic information	unable to capture regular-intensity images	[35–37]
Multi-camera	captures more information and arbitrary views combination	larger data need to be processed	[38–40]

2.1.1. Regular Camera Type

Monocular, stereo, and RGB-D cameras are the most common configurations for visual SLAM algorithms. The monocular system takes a single camera as the input. This cheap and straightforward sensor brings several challenges, and many researchers have dedicated themselves to overcoming them to outperform results with this simple sensor. One of the biggest challenges for the monocular system is scale ambiguity because it cannot measure the scene's depth and estimate the up-to-scale motion; subsequently, the system inevitably suffers from scale drift, which can significantly reduce its accuracy. On the contrary, the stereo system can measure depth with a pair of cameras fixed with a constant baseline (or, more generally, with an overlapping field of view and fixed extrinsics). Subsequently, the disparity of the two cameras is computed by the stereo matching algorithm to produce the depth point with the actual scale. Notably, dense depth points can be made by epipolar searching. However, the quality is undermined when facing repetitive texture and poor illumination. Similarly, the RGB-D system can measure depth by active stereo or time-of-flight sensing. Therefore, a dense depth image can be directly produced without computation resources and does not rely on environment texture and illumination. However, a limitation exists, since the sunlight can firmly interrupt active sensing. This sensor is only suitable for indoor or short-distance depth measurements.

2.1.2. Special Camera Type

In addition to these regular camera configurations, other practical configurations have emerged for more challenging environments. The event camera, a bio-inspired sensor, is sensitive to dynamics and intensity changes, captures every pixel asynchronously with low latency, and is suitable for dynamic object detection. As a new technique, it is barely researched, and there is still much potential to be unearthed. A multi-camera configuration can enlarge the Field of View (FOV) of the system, thus enabling the system to receive more information that resists the texture-less, dynamic environment and motion blur [40]. Furthermore, a more complete, well-distributed map can be constructed. For designing the multi-camera system, the multiple inputs must be carefully processed to extract the most useful information that enhances performance and maintains acceptable cost.

2.2. Data Processing

A critical step in the Visual Odometry algorithm is building the connection between the new incoming frame and the current estimation (referent frame or local map). Traditionally, there are two methods to do this: feature-based methods and direct methods. Feature-based methods require the feature extractor to extract features and match the features to find correspondences in the reference frame, as in Figure 2. Those methods, however, rely on the feature extractor to extract the invariant salient point, which is robust for rotation, view change, and illumination change. Additionally, there is always a trade-off between robustness and efficiency, while the robust extractor can increase the computation cost. On the contrary, direct methods discard the feature extractor and directly utilize the pixel intensity to align the new incoming frame by minimizing the photometric error. Notably, those methods use full-image information that subsequently achieves more accurate results and is naturally robust against poor texture environments and is efficient, since there is no need to extract features.

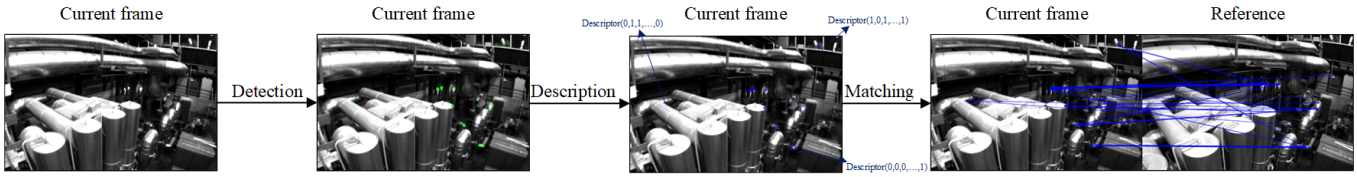


Figure 2. Finding correspondences between the current frame and the reference for the feature-based method (images come from EuRoC [41] dataset). Detection detects pixels with distinctiveness and repeatability; description creates the unique descriptor of features for feature matching; matching compares the similarity of those descriptors to match features.

2.2.1. Feature Extraction Algorithms

While the direct method seems to outperform the feature-based method, some disadvantages make the feature-based method worthwhile to develop continually. First of all, the direct method is based on the assumption that intensity invariance and scene illumination changes will be a disaster for this method. Secondly, this method has failed to build a strong association that limits the performance in long-term large-scale SLAM algorithms. Anyway, the feature-based method and feature extraction algorithms are still important. Feature extraction algorithms are widely researched in computer vision, and they traditionally detect features with pixels that are distinctive by a manually designed formulation. The Harris corner detector [42] is broadly used in computer vision tasks that extract the points by computing their intensity change in a small region. Features with rotation and illumination invariance can be extracted efficiently. The Tomasi corner detector [43] is similar to the Harris corner detector, with a proposed feature selection criterion. This detector extracts the good features that are more robust and less outlying, afterward improving the SLAM algorithms since the outliers could be harmful and interrupt the motion estimation.

However, the Harris and Shi–Tomasi corners lack scale invariance that cannot be matched in close or away motion. A Scale-Invariant Feature Transform (SIFT) algorithm was proposed by [44] to extract robust features to scale, including illumination, view changes, image rotation, and noise. This algorithm uses a coarse-to-fine approach to detect the features, which initially using the algorithm efficiently to identify potential features and refine them for solid invariance. It creates the descriptor for feature matching by computing the gradient magnitude and orientation in a designed region. However, the high computational cost hinders its usage in the visual SLAM algorithm, which requires real-time performance in limited computation devices. By simplifying the existing Hessian matrix-based detector and gradient distribution descriptor, the SURF algorithm [45] extracts the features with comparable repeatability, distinctiveness, and robustness against SIFT but improves efficiency. This algorithm uses Hessian matrix approximation to reduce the computation in the detector. It describes a distribution of Haar-wavelet responses in a

64-dimensional region of the feature, thus reducing the computation cost for detection, description, and matching.

Aiming at low-power CPU devices with limited computation and parallelizing ability, Rublee et al. proposed an efficient feature extraction algorithm called ORB [46]. Building on the FAST corner detector [47] and the BRIEF binary descriptor [48], they designed an efficient method to compute the orientation of FAST corners (Oriented FAST) and a rotation-aware BRIEF descriptor, improving performance while rotating. BRISK [49] is another efficient feature extraction algorithm that experimentally shows more scale invariance against ORB features with an acceptable computational cost. This algorithm adopts the scale space feature detection method inspired by AGAST [50] to improve scale invariance. It uses a compact binary string to describe the features similarly to BRIEF but with different sampling patterns. For those two algorithms, feature detection can quickly detect extensive features, and the efficient binary descriptor further improves the efficiency. This is contrary to the idea that the quality of features is better than the quantity, and it introduces the outliers into the visual SLAM algorithm. However, this is the trend of feature extraction algorithms in visual SLAM; robustness and accuracy are compromised for efficiency. This is because the outlier ejection method can later filter the extracted features to improve the performance of the SLAM algorithm, similar to the coarse-to-fine procedure that puts the main computation into outlier ejection, which deals with feature-level data to reduce overall computation cost, instead of feature extraction, which deals with pixel-level data. Of course, with acceptable efficiency depending on the devices and applications, the higher quality of features can still enhance visual SLAM algorithms.

KAZE [51] detects and describes features in a nonlinear scale space by means of nonlinear diffusion filtering to improve their repeatability and distinctiveness. AKAZE [52] improves the efficiency of the KAZE algorithm to make it available for embedded devices; it uses Fast Explicit Diffusion (FED) to dramatically accelerate feature detection in nonlinear scale spaces, and a Modified-Local Difference Binary (M-LDB) descriptor to efficiently describe features. to evaluate their detection, description, and matching performance. In [53], 14 feature extraction algorithms in 10 extremely variant image pairs were compared. A comprehensive comparison of the above-mentioned algorithms was presented by [54].

2.2.2. IMU Pre-Integration

Specifically, the visual SLAM algorithm can be integrated with an Inertial Measurement Unit (IMU), usually containing an accelerometer and a gyroscope. The algorithm can be more accurate and can solve temporary visual tracking fails by leveraging the self-motion measurement provided by IMU. For those algorithms (visual–inertial SLAM or odometry), the IMU regularly measures the acceleration and angular velocity at a high rate, since the modern loosely coupled visual–inertial algorithm considers IMU measurements as variables in the back-end factor graph to perform optimization. However, problems arise. On the one hand, high-rate IMU measurements will dramatically increase the number of variables, increasing the optimization and computation cost scale. On the other hand, the optimizable variables are generated by IMU integration (acceleration is integrated as velocity, velocity is integrated as translation, and angular velocity is integrated as rotation): they contain the integration operation and, thus, are hard to optimize. To solve these problems, IMU pre-integration methods are proposed. In general, IMU pre-integration [55] summarizes IMU measurements between two concussive image frames to a single compound measurement, as Figure 3, which constrains the frame-to-frame motion and is easy to optimize.

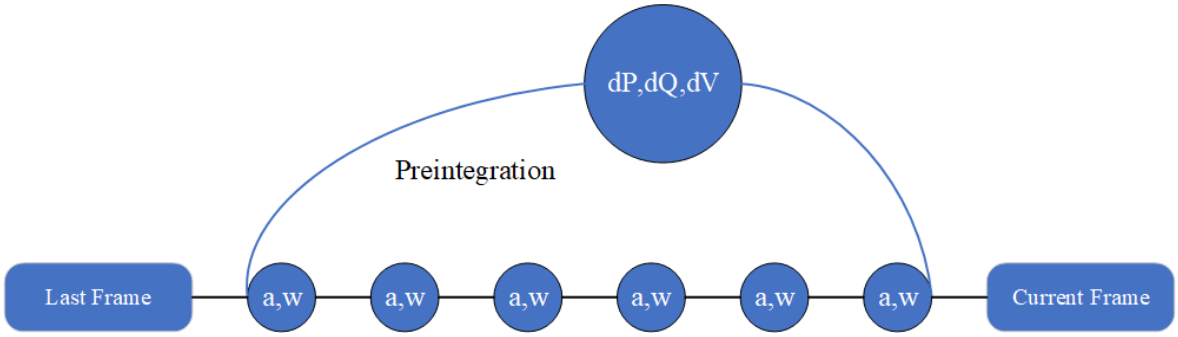


Figure 3. IMU measurements pre-integration: letters a and w, respectively, represent acceleration and angular velocity; dP, dQ, and dV are increments of position, orientation, and velocity between the last frame to the current frame.

According to the above illustrations, some essential preliminaries are given. The algorithm input and property can be determined by the sensor configuration and the scheme to process it. For instance, the monocular inertial SLAM algorithm, jointly visual and inertial initialization, can recover the real scale to avoid scale ambiguity. Furthermore, the IMU constraints are created by pre-integration to perform visual–inertial optimization, which refines the estimated scale. Especially for UAV applications, low-power, light-weight, and low-cost sensors are prior considerations due to the characteristics of limited power consumption and vulnerability. Additionally, heavy data processing cannot be afforded for embedded devices.

3. Visual SLAM Algorithms

Like Structure From Motion (SFM) [56], which estimates camera motion and constructs the unknown environment, the visual SLAM solves problems. However, they have a different emphasis: the SFM technique is a classical subject in the computer vision area; it reconstructs the 3D scene from a set of images (video stream or random images) and is allowed for offline computation. The visual SLAM technique was initially proposed for robotics applications and required real-time computation, emphasizing accurate, robust localization rather than mapping. In other words, the map served for localization. This section reviews the state-of-the-art or classic visual SLAM and odometry algorithms and three features: real-time performance, texture-less environment, and dynamic environment. We categorize those algorithms to illustrate the trend of development. We start from the early feature-based algorithms for real-time performance and show how to achieve real-time processing. At the same time, the essential requirement is satisfied, and the robustness and accuracy of the algorithms are considered, specifically in two typical environments: texture-less and dynamic.

3.1. Real-Time Performance

Global Bundle Adjustment (GBA) is widely used in SFM algorithms to optimize whole structures and poses jointly; it achieves accurate results but with too much computation cost. Therefore, this scheme was deemed unable to be applied in visual SLAM algorithms at an early stage. To this end, filtering schemes were adopted to solve the SLAM problem [57–60]. Among those filtering schemes, the Extended Kalman Filter (EKF) is the most widely used because it efficiently propagates the state and uncertainty. A top-down Bayesian framework to perform visual SLAM was proposed by [57]. In this framework, state estimation is computed by first-order uncertainty propagation in constant time. MonoSLAM [58] has successfully applied visual SLAM in interactive augmented reality and humanoid robots in room-sized domains with a single free camera. Using the EKF scheme, the state of the sensor can be incrementally estimated. However, this scheme can drift over time, since the state estimation only considers the last state, yet there are other past states. Furthermore,

the computation cost increases along with new incoming features. These problems lead to filtering-based visual SLAM failing to deal with long-time large-scale scenes.

While filtering-based visual SLAM seems to reach its limitations, to overcome these limitations, Refs. [61,62] used the Smooth Variable Structure Filter (SVSF) to solve the SLAM problem. This filter is significantly robust against uncertain parameters and unknown noise characteristics. Both methods were shown to outperform conventional filtering-based methods in terms of accuracy and robustness. The Adaptive Smooth Variable Structure Filter (ASVSF) was proposed by [63], introducing a covariance matrix to assess the estimated uncertainty of the original SVSF and achieving more robust localization performance, especially in unstable noise disturbance. Furthermore, Ref. [64] deals with the dynamic environment by removing the dynamic information. Recently, Ref. [65] proposed a monocular–inertial SLAM algorithm based on SVSF, achieving a real-scale localization solution for UAV navigation. Overall, SVSF-based SLAM algorithms show great performance compared to conventional filtering methods and are capable of handling uncertainty and extensive noise.

Moreover, experiment [66] further shows that the BA scheme is more suitable for visual SLAM, in terms of accuracy, robustness, and efficiency. Local BA instead optimizes a batch of local keyframes and map points for real-time processing and has become the mainstream of current visual SLAM algorithm research. Along with advances in semiconductor technology, the CPU has become more and more parallel. To utilize this property, PTAM [67] is presented for tracking a hand-held camera in a small AR workspace by splitting the SLAM algorithm into tracking and mapping, respectively, running in two separate threads. In the tracking thread, a coarse-to-fine procedure is executed to estimate the current camera pose based on the feature-based method. At the same time, local and global BA are performed in the mapping thread to optimize the poses and sparse map points jointly. A keyframe selection strategy is adopted to control the optimization scale, which intensively reduces the amount and improves the quality of the optimizable variables. This algorithm shows the great advantage of the BA scheme against the EKF scheme. Furthermore, this parallel pipeline has been acknowledged and has significantly influenced the later visual SLAM algorithms. A dense visual SLAM algorithm for RGB-D cameras, similar to PTAM, was proposed by [32]. They split the algorithm into two components: fast odometry to register the current frame to the keyframe by direct method and a pose graph built by keyframe selection and optimized when a loop is detected. This algorithm uses a novel entropy-based keyframe selection strategy, inserting the frame when estimation uncertainty grows. However, the above two algorithms are only suitable for small-scale and indoor scenes. LSD-SLAM [28] can track camera motion using a monocular camera and can build a large-scale, semi-dense map in real time on a CPU device. This algorithm uses a filtering method to estimate the semi-dense depth map of keyframes, and the 3D similarity transforms between keyframes as edges for scale-aware global optimization. As we see, the feature-based, dense direct, semi-dense direct SLAM algorithms are designed by a multi-threads pipeline, and the keyframe selection strategy is adopted to select those essential frames and bridge the odometry to back-end optimization. Specifically, the real-time performance of the visual SLAM algorithm is determined by Visual Odometry, and the later optimized map provides a reference for odometry for more accurate results.

ORB-SLAM [22] is one of the most famous and classic visual SLAM algorithms; it further parallelizes the visual SLAM algorithm as three threads: tracking, local mapping, and loop closing. The local mapping thread provides intermediate results between initial tracking and final global optimization to build local data associations that efficiently optimize the tracking reference to enhance quality. The system is efficient, consistent, and reliable, using the same fast ORB features for all algorithm components. The new frame is tracked in the tracking thread by extracting the ORB features that match the local map. A fixed window is managed in the local mapping thread, and keyframes and map points within this window are optimized as the local map. In the loop closing thread, the loop is detected by DboW2 [68] and the global pose graph BA is performed to eliminate

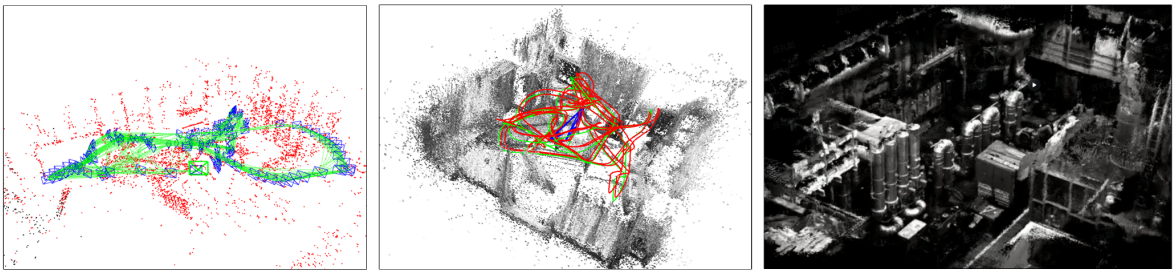
the accumulated drift. This algorithm further parallelizes the pipeline to create short-term, mid-term, and long-term associations, achieving state-of-the-art outdoor and indoor scene performance. ORB-SLAM2 [23] is an extension of the previous version; it takes multiple types of sensors as algorithm input, including monocular, stereo, and RGB-D cameras, and adds a new thread to perform global BA that jointly optimizes all keyframes and map points for more accurate results.

In conclusion, the multiple-thread visual SLAM pipeline is currently mainstream in this area. By splitting tracking and mapping, the real-time performance of visual SLAM systems is determined by the tracking thread. Thus, the above BA-based systems sacrifice accuracy in the tracking thread to ensure its efficiency. For instance, Refs. [22,67] use a fast and fairly robust feature extractor, Ref. [28] rather than using a semi-dense formulation to boost the tracking speed. Additionally, in filtering methods, real-time performance can easily be acquired, especially by redesigning the optimizable variables to reduce the propagation scale. However, this incremental propagation still suffers severely from accumulated drift. To this end, combining filtering methods and BA methods organically should be promising. To use filtering methods in the tracking thread, while in the mapping thread, BA methods refine the accuracy by integrating history information. In other words, filtering methods are more suitable as an odometry algorithm than a SLAM system. Ultimately, the real-time performance of visual SLAM algorithms seems to be enough for current desktop devices or even embedded devices by following this multiple-thread pipeline. However, it is necessary to further improve efficiency and save resources so that enabling more algorithms can be implemented. Performance is always compromised for efficiency, such as by simplifying the feature extractor or reducing the scale of optimization. Therefore, how to improve efficiency without sacrificing performance is crucial. Opinions for further improving efficiency from an engineering perspective are listed as follows :

- GPU boosting. A GPU has the capability of highly parallel computing and can be widely applied in learning-based methods. A common usage of GPUs is to boost feature extraction, since a GPU is a good computing matrix; however, in a visual SLAM system, there are other products that can be parallel-computed, such as map generation and solving BA equations. By utilizing the parallel performance of the CPU and the GPU, the efficiency of visual SLAM can be further improved.
- Data management. Data query is one of the most frequent operations; every new frame inputs to the system, finding correspondences to the reference frame, the local map, and the keyframe database. Therefore, how to use the appropriate data structure to manage the data stored in the system to achieve efficient queries is crucial. Especially in large-scale long-term visual SLAM systems.

3.2. Texture-Less Environment

White walls, space, and long tunnels, texture-less environments, severely undermine feature-based visual SLAM algorithm performance. The direct method performs more robustly in those environments and is accurate because of the utilization of all image pixels. It tracks the image by minimizing photometric error at the assumption of intensity invariance, subsequently saving the computational cost of feature extraction. This method can be categorized into three types: dense, semi-dense, and sparse (see Figure 4). The dense method is used in indoor scenes, with depth images provided by an RGB-D camera to construct the dense surface. The semi-dense approach builds the depth map of the vicinity of gradient pixels with the monocular camera and some prior geometry. However, both the dense and the semi-dense methods fail to jointly optimize pose and structure, leading to less accuracy. The sparse method constructs a sparse map of gradient pixels or patches, which can be jointly optimized with poses to achieve more accurate results.



(a) sparse (b) semi-dense (c) dense

Figure 4. Different type of maps from EuRoC [41] dataset.

3.2.1. Dense Direct Formulation

For these dense methods, RGB-D cameras are usually required to produce dense depth images. KinectFusion [33] is presented for indoor real-time dense surface reconstruction; this algorithm tracks the camera pose by frame-to-model alignment, which ray-casts a global scene model to compute surface prediction and align the live frame through the multi-scale ICP method. Leveraging the GPU parallel computation, this algorithm integrates every frame measurement into the scene model, represented by the Truncated Signed Distance Function (TSDF). DTAM [69] uses a keyframe-based framework to construct a dense depth map by minimizing the global photometric error, requiring only an RGB sensor and commodity GPU hardware. In [32], the RGB-D camera is tracked by minimizing both photometric (intensity) and geometric (depth) errors. Also, a keyframe-based framework is used to construct the dense map. This algorithm does not need GPU enhancement but runs on the CPU device in real time. In [70], semi-dense monocular Visual Odometry is proposed, which continuously estimates a semi-dense inverse depth map of receiving frames. This algorithm represents the pixel inverse depth as a Gaussian probability distribution and propagates it frame-to-frame, constructing the vicinity of large intensity gradients of pixels. Additionally, it shows comparable performance against the dense method without depth cameras and GPU acceleration. LSD-SLAM [28] completes this odometry algorithm to the SLAM algorithm by maintaining a global map that contains a pose graph of keyframes with associated probabilistic semi-dense depth maps. The accumulated drift and scale drift are both reduced for larger-scale estimation.

3.2.2. Sparse Direct Formulation

Since joint optimization of the pose and dense structure in real time is unaffordable, the map points are created initially and fixed with the associated frame, resulting in limited performance. SVO [71] uses the semi-direct method to track camera motion. It initially estimates the camera pose through sparse model-based image alignment, which minimizes the photometric error between pixel correspondences. Then, pose and structure joint optimization is performed through feature alignment by minimizing the reprojection error. In a separate thread, keyframe decision, feature extraction, and depth filter are used to construct the keyframe-based sparse map as a tracking reference. This algorithm utilizes the advantages of both direct and feature-based methods to achieve fast motion tracking. The latter extension [38] supports multiple camera configurations, edge tracking, and other camera models. With a full photometric calibration, including exposure time, lens vignetting, and non-linear response functions, DSO [29] proposes a sparse and direct formulation for Visual Odometry and shows superior performance against dense or direct methods. It minimizes the photometric error with the new formulation modeling the photometric parameters and jointly optimizes camera poses, affine brightness parameters, inverse depth values, and camera intrinsics. It shows the state-of-the-art performance and the great potential of direct sparse odometry.

3.2.3. Structure Feature and Multi-Camera

While direct-based SLAM algorithms effectively solve texture-less environments, their limitations remain, as illustrated above. Structure feature-based algorithms extract points, lines, and planes in the environment to handle texture-less environments where the point features cannot be extracted. PL-SLAM [72,73] is built upon ORB-SLAM that extracts point and line features to track camera motion and perform optimization jointly. By formulating the representation of the line feature and its reprojection error term, those line features can be easily integrated with the original point feature-based algorithm. A new initialization approach is proposed based on only line correspondences that can estimate an initial map from three consecutive frames. Many excellent point and line feature-based algorithms that use the line features to improve robustness and accuracy, especially in texture-less environments, are presented [73–78]. In addition to the algorithm enhancement, texture-less problems can be solved intuitively by enlarging the camera's FOV [79], fish-eye, omnidirectional [80,81], and multi-camera configurations [38–40]; they can receive more information about environments. To fully utilize those configurations, suitable camera models are considered, and relative strategies are proposed instead to migrate those configurations to existing algorithms.

In the context of a texture-less environment, through the comprehensive reviewing, two useful strategies to handle it can be summarized as follows:

- Fully utilizing current information, in such a situation, on how to extract useful information from images is crucial. Direct methods are proposed not only for discarding feature extraction but also to exploit the intensity information of images, making it possible to build correspondences with the current system. That is why direct methods also outperform feature-based methods. Unfortunately, intensity information is unstable compared with features and can be severely interfered with by photometric noise. Structure feature extracts useful information from another perspective. Line and plane are the common geometry similar to points and benefit for building correspondences. However, introducing these relatively complex geometries to the system undermines its efficiency and complexity. What is more, because of the limited FOV of the camera, the system will possibly face a structure-degeneration problem. On the one hand, direct methods need to reduce the influence of photometric noise such as photometric calibration and exposure control, or they need to combine with the feature extractor to provide more stable information, subsequently improving robustness. On the other hand, structure feature-based methods need to simplify the geometry expression and reduce system complexity to improve efficiency.
- Gathering more information, a texture-less environment usually means partial texture deficiency, because it is hard to find somewhere totally without texture. Therefore, simply enlarging the FOV of the sensor is a useful solution to gather more information and support to build correspondence. Subsequently, more data are inputted into the SLAM system, which will cause inefficiency and minimal improvement while in a rich texture environment. Therefore, reducing data redundancy, improving system efficiency, and building data connection (for multi-camera) is essential.

3.3. Dynamic Environment

We emphasize the static environment in the traditional visual SLAM problem. However, a real-world environment is always complex and dynamic. The moving objects presented in the image can interfere with the pose estimation and cause it to fail. Therefore, how to solve the SLAM problem in a dynamic environment has attracted attention in recent years, and is also the foundation of many applications. To solve this problem, several SLAM algorithms are presented and can be categorized into two types. The first type detects and removes the dynamic points or objects in front-end tracking; this method sees dynamic information as outliers, and it processes only static information to simplify the dynamic problem as a static problem, which is an easy and efficient way but fails to take advantage of dynamic information. The second type tracks the moving object while performing

SLAM; this method tracks self-motion by constructing a map with a stationary background and moving objects that utilize the dynamic information and achieve more accuracy than removing them. Furthermore, object-oriented SLAM extracts the semantic information of the environment, including static objects and dynamic objects; joint optimization can be performed by building a consistent object-level map of the environment.

3.3.1. Discarding Dynamic Information

RANdom SAmple Consensus (RANSAC) [82] is a popular method to remove outliers and improve system robustness. It randomly samples the data to fit the model containing the most significant number of inliers. PTAM, ORB-SLAM, and many visual SLAM algorithms have used this method to remove outliers, which keeps algorithms stable in slightly dynamic environments, which may fail when a large part of the image is dynamic. A prior-based adaptive RANSAC algorithm to handle the scene with many dynamic points was proposed by [83]. This algorithm is similar to the standard version but considers the distribution of inliers to fit the model accurately. While the above algorithms use RANSAC as the main scheme for outlier ejection, Ref. [84] proposed a depth edge-based RGB-D SLAM system. By weighting the points and edges to determine whether it is static or dynamic by creating the keyframe with a static feature, the frame-to-keyframe registration is performed for recovering the motion. A dense scene flow representation of the environment, to detect moving objects, was used by [85]. This algorithm performs coarse-to-fine estimation, first estimating the state in a regular way of odometry and later discarding the outliers to obtain more accurate results. For removing the object-level dynamic outlier, Ref. [86] used a Convolution Neural Network (CNN) to perform image segmentation with a priori dynamic objects. This algorithm builds upon the ORB-SLAM2 framework and segments the dynamic object using the Mask R-CNN module; furthermore, it can not only maintain a map with static points but also synthesize the frame without dynamic object occlusion by using a background inpainting module. These algorithms use the simplest way to perform SLAM in a dynamic world to remove the dynamic outlier, which is efficient and valuable. Still, if we utilized the dynamic information instead of discarding it, we could create a high-level understanding of the surrounding environment and even improve tracking accuracy.

3.3.2. Utilizing Dynamic Information

For real-world applications of autonomous robots, a solution of Simultaneous Localization And Mapping (SLAM) and Moving Object Tracking (MOT) is desired, providing the fundamental function for high-level tasks such as autonomous driving and a higher understanding of the environment. A new discipline for this problem in theoretical and practical perspectives was established by [87]. Theoretically, it proposes a mathematical model to solve the SLAM and MOT problems jointly and builds a solid foundation. From a practical standpoint, it develops an algorithm to model perception, motion, and data association. SLAM++ [88] uses the ICP registration to track live images and detect 3D objects by leveraging prior knowledge through tracking 6DoF objects. An efficient pose graph optimization is performed with camera and object pose. DynamicFusion [89] is presented, which can track the non-rigid dynamic scene motion in real time using a single depth camera. This algorithm warps the scene geometry into the live frame to recover the scene motion without prior information. MaskFusion [90] is presented to track the multiple rigid objects in the scene by image-based instance-level semantic segmentation; this algorithm uses a mask network to update the new frame and then perform the motion tracking and object-level mapping. However, those two algorithms are designed for indoor scenes and still fail to utilize dynamic information to enhance the system's performance. ClusterSLAM [91] proposes a back end for a stereo visual SLAM system that uses static and dynamic landmarks. By clustering the motions of dynamic rigid components, a decoupled factor graph optimization can be performed to estimate camera egomotion, static landmarks, and dynamic rigid motion. While the ClusterSLAM is only a back end that

heavily relies on landmark tracking and association quality, ClusterVO [92] is proposed to contain a complete pipeline for either camera or moving object estimation. This algorithm extracts the ORB features and semantic bounding boxes and creates multi-level probabilistic association. For clustering the landmarks into rigid moving objects, the heterogeneous CRF module is used and, finally, state estimation is performed with sliding windows BA optimization. DynaSLAM2 [93] and VDO-SLAM [94] integrate the camera poses, static and dynamic points, and object motion into a BA factor graph optimization and utilize ynamic information, result in excellent performance.

In conclusion, discarding dynamic information is a simple but useful strategy to deal with dynamic problems and is especially suitable for indoor low dynamic scenes, since moving objects are random and relatively uncommon, simplifying problems from dynamic to static to maintain an efficient system. Utilizing dynamic information is more complex and suitable for outdoor scenes because pedestrians or cars usually have regular movements. However, whether discarding or utilizing dynamic information, localization is supported mainly by static information. Dynamic points filtering, semantic segmentation culling, object tracking, and scene flow tracking undermine and eliminate the disturbance of dynamic information. To this end, while static information is not enough, dynamic problems become texture-less problems. To deal with a highly dynamic environment, combining the direct method, structure feature, and FOV expansion is a practical scheme. Additionally, to enhance those SLAM systems coupled with a deep learning module such as semantic segmentation, the generalization, efficiency, and complexity of the model must be considered.

In this section, we first introduce the development of the visual SLAM algorithm structure, from filtering-based to optimization-based, from single-thread to multi-thread. This modern pipeline decouples localization and mapping from hard real-time constraints and changes the real-time requirement of the SLAM algorithm to the odometry algorithm. This parallelizing pipeline allows researchers to study and modify, helping engineering applications and academic research. We discuss two real-world problems and how the SLAM algorithm deals with a texture-less and dynamic environment. In texture-less environments, we emphasize direct-based algorithms, one of the two main branches of visual SLAM. Dense, semi-dense, and sparse methods show the development of this method. Subsequently, other methods that structure feature extraction and FOV expansion are briefly discussed. Two strategies are adopted in dynamic environments: discarding or tracking the dynamic points. The former is a simple scheme to handle dynamic environments, turning a dynamic problem into a standard static problem. The latter utilizes dynamic information to improve performance, incorporating optical flow, MOT, and semantic segmentation techniques. For both academic and engineering purposes, we list several open source visual SLAM algorithms and summarize their sensor inputs and features in Table 2.

Table 2. Open source visual SLAM algorithms.

Odometry Method	Algorithm	Sensor	Feature
Feature-based	Mono-SLAM [58]	Monocular	EKF-based
	PTAM [67]	Monocular	parallel tracking and mapping
	ORB-SLAM [22]	Monocular	multi-threads
	DynamicFusion [89]	RGB-D	non-rigid dynamic scene motion tracking
	ORB-SLAM2 [23]	Monocular, stereo, RGB-D	multi-configurations
	PL-SLAM [72,73]	Monocular, stereo	point, line feature extraction
	Dyna-SLAM [86]	Monocular, stereo, RGB-D	segment dynamic objects
	MaskFusion [90]	RGB-D	tracks multiple objects
	VDO-SLAM [94]	RGB-D	joint optimization including camera poses, static, dynamic points, and object motion
	PLP-SLAM [78]	Monocular, stereo, RGB-D	point, line, plane feature extraction

Table 2. *Cont.*

Odometry Method	Algorithm	Sensor	Feature
Direct	KinectFusion [33]	RGB-D	dense surface reconstruction
	DTAM [69]	Monocular	monocular dense tracking and mapping
	DVO-SLAM [32]	RGB-D	tracks motion by minimizing both photometric and geometric errors
	LSD-SLAM [28]	Monocular	large-scale estimation
	SVO [38,71] *	Monocular, Multiple camera	hybrid odometry
	DSO [29]	Monocular	sparse direct formulation
	BAD-SLAM [34]	RGB-D	fast direct BA formulation

* SVO is a semi-direct odometry, using both photometric and geometric error.

4. Visual–Inertial Fusion for UAV Localization

The pure visual SLAM algorithm obtains information from the external surrounding environments and fails to sense self-motion. Environmental conditions (over-exposure, dusty conditions, and dark regions) can directly lead to deadly error in the algorithm. Therefore, IMU is needed to significantly reduce the influence of environmental conditions, especially in UAV applications [95–97], where robust localization is required to prevent accidents, such as losing control or dropping from the sky. Since self-motion can be attained and integrated into the visual SLAM algorithm, there are several advantages to integrating visual and inertial measurements:

- Inertial measurements provide extra constraints for pure visual back-end optimization and improve its accuracy and robustness.
- The real-world scale can be recovered for monocular SLAM algorithms to solve the scale ambiguity problem.
- The high-frame-rate inertial measurements can fast propagate the odometry information for autonomous robot agents.
- While visual tracking fails to maintain the odometry, inertial measurements could provide a temporary prediction for keeping the system working.

At the same time, integrating inertial measurement brings new challenges for the visual algorithm. On one hand, the high-frame-rate inertial measurement needs to be processed appropriately to fit the low-frame-rate visual system. On the other hand, the accumulated bias and error of IMU measurement need to be fixed by utilizing the visual information. For UAV applications, visual–inertial solutions have gained plenty of interest, and this sensor configuration significantly improves the algorithm performance in a relatively simple way against complex structure design, algorithm optimization, and mathematical formulation.

4.1. EKF-Based Visual–Inertial SLAM

Similar to pure visual SLAM, this technique also starts with filtering-based algorithms. In [98], an EKF-based visual inertial odometry uses static features to constrain inertial propagation. The algorithm is computationally efficient and can precisely estimate large-scale real-world environments. ROVIO [99] is proposed to use the photometric error of the multi-level patch as an innovation term in EKF propagation. By parametrizing features, the filtering operations can be applied. Overall, this algorithm propagates robot-centric rotation, translation, velocity, the transformation of IMU and camera, IMU bias, and feature parameters, and employs QR-decomposition to maintain computational efficiency. These EKF-based visual–inertial algorithms have experimentally demonstrated robustness and accuracy by integrating inertial and visual measurements. However, the optimization-based system could perform better in terms of robustness and accuracy by constructing a factor graph for joint optimization.

4.2. BA-Based Visual–Inertial SLAM

In [100], a tightly coupled visual–inertial SLAM was proposed, introducing the IMU error term integrated with feature reprojection error for joint optimization. A marginalization scheme was employed to maintain the visual constraints of keyframes for bounding computation complexity. ORB-SLAM-VI [101] proposes zero-drift localization by re-using the map; this algorithm performs optimization within a local window but considers a fixed window connected by a co-visibility graph and loop closure with a pose graph. In addition to this, a novel IMU initialization method is proposed that computes scale, gravity direction, velocity, and biases.

VINS-MONO [102] is a tightly coupled monocular inertial system. In [102], they propose a robust initialization procedure that performs vision-only initialization and then aligns metric IMU pre-integration with the visual-only result to recover scale, gravity, velocity, and biases. For front-end tracking, the system tracks the existing features by the KLT sparse optical flow algorithm and detects new features to maintain a minimum number of features in the current frame; simultaneously, IMU measurements within two frames are pre-integrated. If the system is initialized, a tightly coupled optimization includes pre-integration terms, features, and poses in a sliding window. Furthermore, they adopt the DBoW2 to detect and close the loop by 4-DOF global pose graph optimization, since IMU can provide absolute pitch and roll observation.

ORB-SLAM3 [24] further extended ORB-SLAM—which supports both visual and visual–inertial sensor configurations, including monocular, monocular inertial, stereo inertial, etc.—and introduced Atlas to save a set of disconnected maps that can be used for loop detection and relocalization, and which merge smoothly with the current connected map. This algorithm is based on ORB-SLAM2 and integrates the IMU measurements; it initializes the system in three steps: visual-only, inertial-only, and visual–inertial joint initialization. In the tracking thread, the algorithm continuously pre-integrates IMU measurements and estimates the pose by feature extraction and matching. When the tracking is lost, the system will attempt to predict motion by IMU measurements, and if this does not work, the relocalization mode will be executed. In the mapping thread, the IMU constraints will be added to the graph for joint optimization and will perform IMU scale refinement. In the loop closure thread, while the system detects a loop in ATLAS, two maps will be merged into one map, and essential graph optimization will be performed. This algorithm is complete, supports almost all visual sensor configurations and different camera models, and contains short-term, mid-term, and long-term data associations; therefore, it shows excellent accuracy and robustness.

VI-DSO [103] and DM-VIO [104] are monocular visual–inertial odometry, which minimize the photometric errors of sparse pixels with high-intensity gradient and IMU measurement errors. VI-DSO introduces a novel marginalization procedure called dynamic marginalization that maintains several marginalization priors to adapt the scale estimation dynamically. Preventing the scale is fixed by the marginalization prior, while it needs to be better estimated. DM-VIO proposes delayed marginalization to solve marginalization that is hard to reverse. This approach can inject the IMU information after initialization to the pure visual prior and replace the prior. At the same time, the scale estimate changes in the same way as VI-DSO. Additionally, a weighted photometric BA is proposed to adjust the weight of visual residuals dynamically. Those algorithms improve the marginalization procedure to better compute the priors in BA optimization. DM-VIO exceeds the state-of-the-art visual–inertial stereo algorithms and has shown its effectiveness. Table 3 summarizes the above algorithms with three basic components.

In conclusion, this technique builds upon visual SLAM and achieves excellent performance, in terms of robustness and accuracy, by integrating IMU measurements. Meanwhile, in UAV applications such as unknown-space perception, industrial-defect inspection, and military operations, this technique provides a suitable solution for localization. Specifically, for navigation, robust localization enables the UAV to fly in challenging environments with low illumination, over-exposure, and poor texture. For exploration and reconstruction,

accurate localization provides a solid foundation to improve the quality and consistency of reconstruction. For flying control, more body states such as velocity and accelerated velocity and high-rate odometry information propagated by IMU support aggressive control to utilize the flexibility of UAV.

Table 3. Open source visual–inertial SLAM algorithms.

Algorithm	Odometry Method	Optimization	Loop Closure
MSCKF [98]	feature-based	EKF-based	-
OKVIS [100]	feature-based	Local BA	-
ROVIO [99]	direct	EKF-based	-
ORB-SLAM-VI [101]	feature-based	Local BA	PGBA
VINS-Mono [102]	feature-based	Local BA	PGBA
VI-DSO [103]	direct	Local BA	-
ORB-SLAM3 [24]	feature-based	Local BA, GBA	PGBA
DM-VIO [104]	direct	Local BA	-

5. Learning-Based Enhancement for UAV Perception

In UAV applications, obstacle avoidance, path planning, and real-time reconstruction, a dense or semi-dense map will be required. However, for robustness, accuracy, and efficiency of localization, visual SLAM algorithms usually retain a sparse map for joint optimization. To this end, another denser map is constructed by obtaining depth images and associated odometry. Depth images are projected using a camera model to produce a dense point cloud, which is aligned by the odometry of the SLAM algorithm [105,106]. This approach usually requires depth sensors, such as Realsense D435i, Realsense D455, and Kinect v2, to provide depth images. This approach is limited by the performance of depth sensors like RGB-D cameras, which will be disrupted by sunlight and will fail to measure the depth of outdoor long-range scenes. Therefore, gathering the depth from original visual SLAM inputs gains excellent interest, reducing the sensors' cost and improving the synchronization between depth image and odometry, since the depth sensor probably needs to be better synchronized with SLAM algorithm inputs. With the advancement of GPUs, Jetson developer kits such as Jetson Xavier NX, Jetson TX2, and Jetson Orin integrate GPUs into embedded devices to enable the implementation of learning-based modules. These low-cost devices are suitable for autonomous mobile robots such as UAVs and have been successfully applied in commodity drones such as Skydio. There are some advantages of learning-based enhancement for UAV perception:

- Learning-based methods are good for extracting texture information in images, and usually perform better than traditional methods.
- Learning-based perception uses the same inputs as in localization, improving consistency and, at the same time, saving the cost of sensors.
- With the rapid development of GPUs and Artificial Intelligence, learning-based methods have become mainstream.

Similar to embedded CPUs, embedded GPUs also suffer from power limitations that require the implemented learning-based module to be simple and efficient.

5.1. Monocular Depth Estimation

Monocular depth estimation, a technique to estimate the depth of pixels in 2D images, is significantly enhanced by deep learning. CNNs can extract richer and more complex feature representations than traditional approaches, which usually rely on hand-crafted features, scene assumption, and manual parameters adaption, subsequently achieving better results for depth estimation. In [107], two CNN stacks were used for monocular depth estimation: one stack to estimate the global scale depth, and another to refine the local detail. In [108], the same multi-scale architecture as in [107] was adopted to predict depth, surface normals and semantic labels. In [109], a unified deep CNN framework was

used to learn the potential of a continuous Conditional Random Field (CRF), estimating the depth of general scenes without geometric priors and extra information. On the contrary, Ref. [110] did not rely on refinement or CRF, and proposed a full CNN architecture to estimate depth. This architecture was built upon ResNet, and it outperformed previous methods. In [111,112], the sparse pixel depth provided by the visual SLAM algorithm was used to improve the accuracy and reliability of depth estimation. Monocular depth estimation algorithms rely on deep learning-based methods that focus on estimation accuracy but with increasing computational complexity. To this end, FastDepth [113] proposed a lightweight encoder–decoder network to efficiently estimate the monocular depth map, which achieves comparable accuracy with embedded GPU devices in real time. In [114], a proper trade-off was achieved between accuracy and efficiency, assembling two encoder–decoder subnetworks to solve spatial information loss caused by the feature extractor. These efficient networks provide an alternative solution for UAV depth sensing, which gets rid of the limitations of depth sensors. Incorporating accurate localization, a promising dense depth map can be constructed.

5.2. NeRF-Based SLAM

Neural implicit fields (NeRF) [115] are novel representations that reconstruct high-fidelity surfaces and arbitrary view renderings of scenes. Compared to traditional point cloud reconstruction, NeRF-based reconstruction produces realistic illumination and high-quality images, resulting in excellent performance of complex scenes and detailed reconstruction. Visual SLAM incorporated with NeRF for camera tracking and dense reconstruction has recently been investigated and called NeRF-based SLAM. Unlike the traditional dense visual SLAM, this technique overcomes the drawback of failing to jointly optimize the structure and poses since this implicit representation is differentiable. The first NeRF-based SLAM to track an RGB-D camera pose in real time and to jointly optimize poses with a dense map was iMAP [116]. Following the traditional visual SLAM pipeline, tracking and mapping are run in two parallelizing threads to decouple the hard real-time constraints. NICE-SLAM [117] further improves the efficiency of tracking and mapping, incorporating multi-level local information. This algorithm is more scalable and robust than other NeRF-based SLAM algorithms. NeRF-based SLAM is a new trend in visual SLAM research. Therefore, some problems still need solutions. Urgently, the efficiency of algorithms needs to be improved to reduce the hardware requirement, which is usually a desktop setup. More recently, 3D Gaussian splatting [118] has been used for real-time radiance field rendering. By improving efficiency, these algorithms can be applied in UAVs, which will be a milestone for UAV reconstruction. In addition, current NeRF-based SLAM algorithms only support small-scale indoor scenes, with a solution still needed for large-scale outdoor scenes.

Regular pipelines of the above three schemes are depicted in Figure 5. In conclusion, along with the development of GPU-based hardware, the accurate and excellent performance of learning-based techniques for UAV perception have shown us new solutions. Lower sensor complexity and better performance make UAVs more simple and powerful. However, due to the limited computational source of embedded devices, the main problem or research direction for these learning-based modules to be applied in UAVs is the efficiency of algorithms.

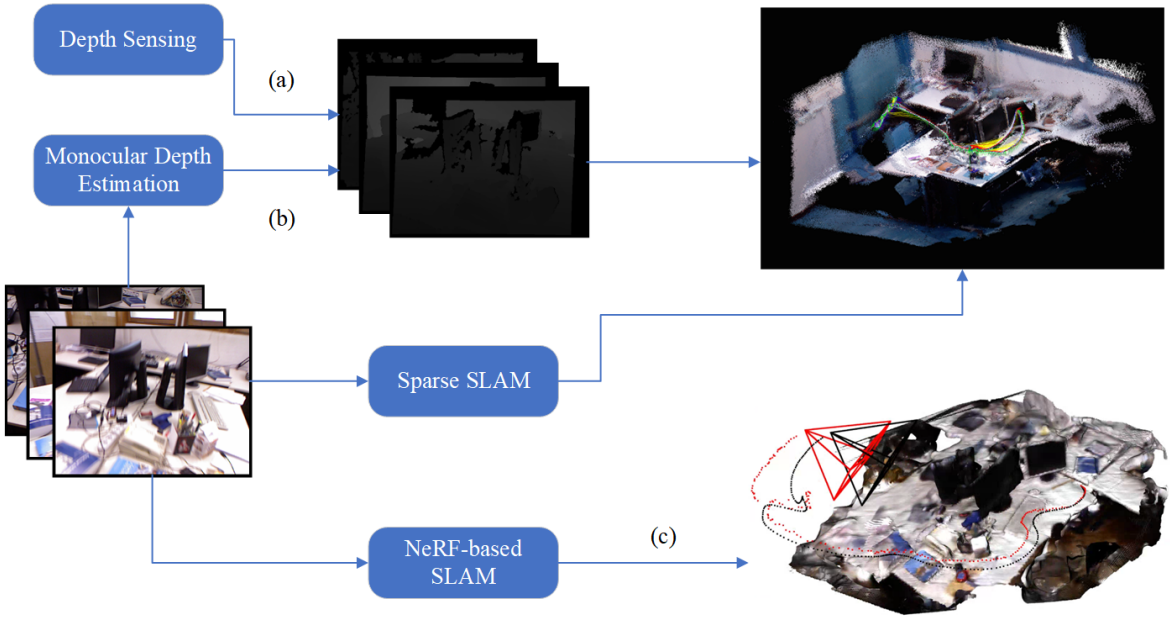


Figure 5. Three schemes to build a dense map (experiment dataset [119]): (a) and (b) use different ways to produce depth image, the former measuring depth by depth sensors, while the latter estimates depth by deep learning modules and reconstructs a dense map by retrieving the pose of the depth images; (c) a NeRF-based SLAM produces a NeRF representation of environments. Other dense SLAM algorithms usually result in poor performance because of the lack of joint optimization.

6. Conclusions

This paper reviewed the development of visual SLAM in three aspects: real-time performance, texture-less environments, and dynamic environments. Introducing state-of-the-art and recently presented algorithms, we illustrated the mainstream of pipeline design and the two main applied environment problems. Furthermore, localization and perception for UAV applications were discussed, based on visual SLAM. For localization, we emphasized the widely applied visual–inertial SLAM to show how inertial measurements improve UAV localization and subsequently improve other tasks such as exploration. For perception, we demonstrated the capabilities of the newly presented learning-based methods. GPU integrated into embedded devices can make implementing learning-based modules for UAV perception possible.

Through decades of development, visual SLAM has become more complete and powerful, providing outstanding localization and mapping for various robotic applications. Recent rapidly developed learning-based approaches have continued to improve this technique, to transform traditional hand-crafted algorithms into data-driven algorithms, replacing conventional feature extraction, odometry, and even the whole system. However, while these data-driven algorithms perform better than hand-crafted ones, their generalization ability, interpretability, and efficiency are still open to question. Meanwhile, traditional visual SLAM modules can be improved for more challenging environments, hardware, and motion. To provide a strong foundation for UAV applications, multi-sensor fusion is an efficient scheme for localization and perception. Among these configurations, visual–inertial fusion is undoubtedly a simple and effective way to improve localization. Multi-camera fusion also improves perception and localization by enlarging the FOV. For multiple data inputs, efficiently processing them could be a crucial problem that filters redundant or wrong information and uses essential information to enhance performance. Additionally, the later keyframe decision, BA optimization, and map management need to be considered to adjust multiple inputs so that they can be fully utilized.

To handle complex scenes and missions, autonomous UAVs are required to be more reliable and intelligent. Therefore, robust localization and intelligentized perception are

essential trends in autonomous UAV applications. Further explanations are discussed as follows:

- Robust localization: UAV localization may not be so accurate but is robust, especially for applications that require the UAV to cross various scenes, such as unknown space exploration. As in military investigation, dusty disturbance, complex environments, and the requirement for fast movement severely interfere with the data inputs. In cave or tunnel exploration, partial darkness, and reduced environmental texture also weaken information support for localization. These potential factors bring challenges to UAV localization, and only robust UAV localization can deal with various challenging environments, meeting the needs of the day-by-day growing complexity of applications.
- Intelligentized perception: A regular dense point cloud map or grid map is built for point-driven navigation; however, it is not adequate to support more and more intelligent missions, such as object searching in an unknown space. For instance, in disaster rescue, the regular map for navigation only supports the UAV to mechanically search victims, and it is inefficient. Intelligentized perception means a high-level understanding of surrounding environments: with this understanding, the UAV can infer potential victims by obtaining environmental clues, such as blood or a piece of clothing. Moreover, manual intervention can be significantly reduced and UAVs can take charge of strategy decisions, parameter adjustment, and risk prevention.

Additionally, there is the popular concept of lifelong SLAM [120] with continuous localization and mapping in the long term. Robust localization meets the needs of long-term localization in complex and changing environments. In addition, intelligentized perception helps the system to understand changes in scenes and objects for long-term mapping. Moreover, this understanding supports the system in mission planning and decisions.

This paper summarizes the research, including visual SLAM, visual–inertial SLAM, and learning-based SLAM, from different aspects, to comprehensively understand this technique. Furthermore, we have discussed the problems, advantages, and future trends of these relative approaches. With the advent of Artificial Intelligence, it is worth reviewing these conventional approaches, to ascertain potential and understand the foundation of this technique.

Author Contributions: Conceptualization, L.Z. and W.Y.; formal analysis, L.Z., L.X. and W.Y.; investigation, L.Z.; resources, L.Z. and C.T.; data curation, L.Z.; writing—original draft preparation, L.Z., L.X. and C.T.; writing—review and editing, L.Z., X.Z., L.X. and W.Y.; visualization, X.Z. and C.T.; supervision, W.Y.; project administration, W.Y.; funding acquisition, W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key Research and Development Program of China (2022YFB3904602).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: Our thanks to Autonomous Aerial Mobile Sensing Group (ARMS) at the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) for knowledge and equipment support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Al-Kaff, A.; Martin, D.; Garcia, F.; de la Escalera, A.; Armingol, J.M. Survey of computer vision algorithms and applications for unmanned aerial vehicles. *Expert Syst. Appl.* **2018**, *92*, 447–463. [CrossRef]
2. Lu, Y.; Xue, Z.; Xia, G.S.; Zhang, L. A survey on vision-based UAV navigation. *Geo-Spat. Inf. Sci.* **2018**, *21*, 21–32. [CrossRef]

3. Kanellakis, C.; Nikolakopoulos, G. Survey on computer vision for UAVs: Current developments and trends. *J. Intell. Robot. Syst.* **2017**, *87*, 141–168. [CrossRef]
4. Aulinas, J.; Petillot, Y.; Salvi, J.; Lladó, X. The SLAM problem: A survey. *Artif. Intell. Res. Dev.* **2008**, *184*, 363–371.
5. Takleh, T.T.O.; Bakar, N.A.; Rahman, S.A.; Hamzah, R.; Aziz, Z. A brief survey on SLAM methods in autonomous vehicle. *Int. J. Eng. Technol.* **2018**, *7*, 38–43. [CrossRef]
6. Song, J.B.; Hwang, S.Y. Past and state-of-the-art SLAM technologies. *J. Inst. Control Robot. Syst.* **2014**, *20*, 372–379. [CrossRef]
7. Taketomi, T.; Uchiyama, H.; Ikeda, S. Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Trans. Comput. Vis. Appl.* **2017**, *9*, 1–11. [CrossRef]
8. Macario Barros, A.; Michel, M.; Moline, Y.; Corre, G.; Carrel, F. A comprehensive survey of visual slam algorithms. *Robotics* **2022**, *11*, 24. [CrossRef]
9. Tourani, A.; Bavle, H.; Sanchez-Lopez, J.L.; Voos, H. Visual SLAM: What are the current trends and what to expect? *Sensors* **2022**, *22*, 9297. [CrossRef]
10. Kazerouni, I.A.; Fitzgerald, L.; Dooly, G.; Toal, D. A survey of state-of-the-art on visual SLAM. *Expert Syst. Appl.* **2022**, *205*, 117734. [CrossRef]
11. Servières, M.; Renaudin, V.; Dupuis, A.; Antigny, N. Visual and visual-inertial slam: State of the art, classification, and experimental benchmarking. *J. Sens.* **2021**, *2021*, 2054828. [CrossRef]
12. Harris, C.G.; Pike, J. 3D positional integration from image sequences. *Image Vis. Comput.* **1988**, *6*, 87–90. [CrossRef]
13. Nistér, D.; Naroditsky, O.; Bergen, J. Visual odometry. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004, Washington, DC, USA, 27 June–2 July 2004; Volume 1, pp. 1–652.
14. Zhang, Z. Determining the epipolar geometry and its uncertainty: A review. *Int. J. Comput. Vis.* **1998**, *27*, 161–195. [CrossRef]
15. Wu, Y.; Hu, Z. PnP problem revisited. *J. Math. Imaging Vis.* **2006**, *24*, 131–141. [CrossRef]
16. Zheng, Y.; Kuang, Y.; Sugimoto, S.; Astrom, K.; Okutomi, M. Revisiting the pnp problem: A fast, general and optimal solution. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 2344–2351.
17. Hesch, J.A.; Roumeliotis, S.I. A direct least-squares (DLS) method for PnP. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 383–390.
18. Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In Proceedings of the Proceedings Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 28 May–1 June 2001; pp. 145–152.
19. Sharp, G.C.; Lee, S.W.; Wehe, D.K. ICP registration using invariant features. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 90–102. [CrossRef]
20. Wang, S.; Clark, R.; Wen, H.; Trigoni, N. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2043–2050.
21. Triggs, B.; McLauchlan, P.F.; Hartley, R.I.; Fitzgibbon, A.W. Bundle adjustment—A modern synthesis. In *Proceedings of the Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms, Corfu Greece, 21–22 September 1999 Proceedings*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 298–372.
22. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [CrossRef]
23. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
24. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.; Tardós, J.D. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]
25. Yousif, K.; Bab-Hadiashar, A.; Hoseinnezhad, R. An overview to visual odometry and visual SLAM: Applications to mobile robotics. *Intell. Ind. Syst.* **2015**, *1*, 289–311. [CrossRef]
26. Aqel, M.O.; Marhaban, M.H.; Saripan, M.I.; Ismail, N.B. Review of visual odometry: Types, approaches, challenges, and applications. *SpringerPlus* **2016**, *5*, 1897. [CrossRef]
27. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
28. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. In *Computer Vision—ECCV 2014, Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014*; Springer: Cham, Switzerland, 2014; pp. 834–849.
29. Engel, J.; Koltun, V.; Cremers, D. Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 611–625. [CrossRef]
30. Wang, R.; Schworer, M.; Cremers, D. Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3903–3911.
31. Engel, J.; Stückler, J.; Cremers, D. Large-scale direct SLAM with stereo cameras. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 1935–1942.
32. Kerl, C.; Sturm, J.; Cremers, D. Dense visual SLAM for RGB-D cameras. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 2100–2106.

33. Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A.; et al. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, 16–19 October 2011; pp. 559–568.
34. Schops, T.; Sattler, T.; Pollefeys, M. Bad slam: Bundle adjusted direct rgb-d slam. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 134–144.
35. Weikersdorfer, D.; Hoffmann, R.; Conradt, J. Simultaneous localization and mapping for event-based vision systems. In *Computer Vision Systems, Proceedings of the 9th International Conference, ICVS 2013, St. Petersburg, Russia, 16–18 July 2013*; Proceedings 9; Springer: Berlin/Heidelberg, Germany, 2013; pp. 133–142.
36. Kim, H.; Leutenegger, S.; Davison, A.J. Real-time 3D reconstruction and 6-DoF tracking with an event camera. In *Computer Vision–ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part VI 14; Springer: Cham, Switzerland, 2016; pp. 349–364.
37. Rebecq, H.; Horstschäfer, T.; Gallego, G.; Scaramuzza, D. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robot. Autom. Lett.* **2016**, *2*, 593–600. [CrossRef]
38. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Trans. Robot.* **2016**, *33*, 249–265. [CrossRef]
39. Harmat, A.; Sharf, I.; Trentini, M. Parallel tracking and mapping with multiple cameras on an unmanned aerial vehicle. In *Intelligent Robotics and Applications, Proceedings of the 5th International Conference, ICIRA 2012, Montreal, QC, Canada, 3–5 October 2012*; Proceedings, Part I 5; Springer: Berlin/Heidelberg, Germany, 2012; pp. 421–432.
40. Kuo, J.; Muglikar, M.; Zhang, Z.; Scaramuzza, D. Redesigning SLAM for arbitrary multi-camera systems. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 2116–2122.
41. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res.* **2016**, *35*, 1157–1163. [CrossRef]
42. Harris, C.; Stephens, M. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*; Citeseer: Princeton, NJ, USA, 1988; Volume 15; pp. 10–5244.
43. Shi, J. Good features to track. In Proceedings of the 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600.
44. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
45. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006, Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006*; Proceedings, Part I 9; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
46. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
47. Rosten, E.; Drummond, T. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006, Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006*; Proceedings, Part I 9; Springer: Berlin/Heidelberg, Germany, 2006; pp. 430–443.
48. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010, Proceedings of the 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010*; Proceedings, Part IV 11; Springer: Berlin/Heidelberg, Germany, 2010; pp. 778–792.
49. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary robust invariant scalable keypoints. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2548–2555.
50. Mair, E.; Hager, G.D.; Burschka, D.; Suppa, M.; Hirzinger, G. Adaptive and generic corner detection based on the accelerated segment test. In *Computer Vision–ECCV 2010, Proceedings of the 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010*; Proceedings, Part II 11; Springer: Berlin/Heidelberg, Germany, 2010; pp. 183–196.
51. Alcantarilla, P.F.; Bartoli, A.; Davison, A.J. KAZE features. In *Computer Vision–ECCV 2012, Proceedings of the 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012*; Proceedings, Part VI 12; Springer: Berlin/Heidelberg, Germany, 2012; pp. 214–227.
52. Alcantarilla, P.F.; Solutions, T. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell* **2011**, *34*, 1281–1298.
53. Tareen, S.A.K.; Raza, R.H. Potential of SIFT, SURF, KAZE, AKAZE, ORB, BRISK, AGAST, and 7 More Algorithms for Matching Extremely Variant Image Pairs. In Proceedings of the 2023 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 17–18 March 2023; pp. 1–6.
54. Tareen, S.A.K.; Saleem, Z. A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 3–4 March 2018; pp. 1–10.
55. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. *IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation*; Technical Report; In Robotics: Science and Systems XI; Sapienza University of Rome: Rome, Italy, 13–17 July 2015.
56. Schonberger, J.L.; Frahm, J.M. Structure-from-motion revisited. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4104–4113.
57. Davison. Real-time simultaneous localisation and mapping with a single camera. In Proceedings of the Proceedings Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; pp. 1403–1410.

58. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067. [CrossRef]
59. Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. FastSLAM: A factored solution to the simultaneous localization and mapping problem. *AAAI/IAAI* **2002**, *593598*, 593–598.
60. Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In Proceedings of the IJCAI 2003, Acapulco, Mexico, 9–15 August 2003; Volume 3, pp. 1151–1156.
61. Demim, F.; Nemra, A.; Louadj, K. Robust SVSF-SLAM for unmanned vehicle in unknown environment. *IFAC-PapersOnLine* **2016**, *49*, 386–394. [CrossRef]
62. Ahmed, A.; Abdelkrim, N.; Mustapha, H. Smooth variable structure filter VSLAM. *IFAC-PapersOnLine* **2016**, *49*, 205–211. [CrossRef]
63. Demim, F.; Boucheloukh, A.; Nemra, A.; Louadj, K.; Hamerlain, M.; Bazoula, A.; Mehal, Z. A new adaptive smooth variable structure filter SLAM algorithm for unmanned vehicle. In Proceedings of the 2017 6th International Conference on Systems and Control (ICSC), Batna, Algeria, 7–9 May 2017; pp. 6–13.
64. Demim, F.; Nemra, A.; Boucheloukh, A.; Louadj, K.; Hamerlain, M.; Bazoula, A. Robust SVSF-SLAM algorithm for unmanned vehicle in dynamic environment. In Proceedings of the 2018 International Conference on Signal, Image, Vision and Their Applications (SIVA), Guelma, Algeria, 26–27 November 2018; pp. 1–5.
65. Elhaouari, K.; Allam, A.; Larbes, C. Robust IMU-Monocular-SLAM For Micro Aerial Vehicle Navigation Using Smooth Variable Structure Filter. *Int. J. Comput. Digit. Syst.* **2023**, *14*, 1063–1072.
66. Strasdat, H.; Montiel, J.M.; Davison, A.J. Visual SLAM: Why filter? *Image Vis. Comput.* **2012**, *30*, 65–77. [CrossRef]
67. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 225–234.
68. Gálvez-López, D.; Tardos, J.D. Bags of binary words for fast place recognition in image sequences. *IEEE Trans. Robot.* **2012**, *28*, 1188–1197. [CrossRef]
69. Newcombe, R.A.; Lovegrove, S.J.; Davison, A.J. DTAM: Dense tracking and mapping in real-time. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2320–2327.
70. Engel, J.; Sturm, J.; Cremers, D. Semi-dense visual odometry for a monocular camera. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 1449–1456.
71. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 15–22.
72. Pumarola, A.; Vakhitov, A.; Agudo, A.; Sanfeliu, A.; Moreno-Noguer, F. PL-SLAM: Real-time monocular visual SLAM with points and lines. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4503–4508.
73. Gomez-Ojeda, R.; Moreno, F.A.; Zuniga-Noël, D.; Scaramuzza, D.; Gonzalez-Jimenez, J. PL-SLAM: A stereo SLAM system through the combination of points and line segments. *IEEE Trans. Robot.* **2019**, *35*, 734–746. [CrossRef]
74. Fu, Q.; Yu, H.; Lai, L.; Wang, J.; Peng, X.; Sun, W.; Sun, M. A robust RGB-D SLAM system with points and lines for low texture indoor environments. *IEEE Sens. J.* **2019**, *19*, 9908–9920. [CrossRef]
75. Yang, S.; Scherer, S. Direct monocular odometry using points and lines. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3871–3877.
76. Gomez-Ojeda, R.; Briales, J.; Gonzalez-Jimenez, J. PL-SVO: Semi-direct monocular visual odometry by combining points and line segments. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 4211–4216.
77. Zuo, X.; Xie, X.; Liu, Y.; Huang, G. Robust visual SLAM with point and line features. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1775–1782.
78. Shu, F.; Wang, J.; Pagani, A.; Stricker, D. Structure plp-slam: Efficient sparse mapping and localization using point, line and plane for monocular, rgb-d and stereo cameras. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 2105–2112.
79. Zhang, Z.; Rebecq, H.; Forster, C.; Scaramuzza, D. Benefit of large field-of-view cameras for visual odometry. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 801–808.
80. Huang, H.; Yeung, S.K. 360vo: Visual odometry using a single 360 camera. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 5594–5600.
81. Matsuki, H.; Von Stumberg, L.; Usenko, V.; Stückler, J.; Cremers, D. Omnidirectional DSO: Direct sparse odometry with fisheye cameras. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3693–3700. [CrossRef]
82. Derpanis, K.G. Overview of the RANSAC Algorithm. *Image Rochester NY* **2010**, *4*, 2–3.
83. Tan, W.; Liu, H.; Dong, Z.; Zhang, G.; Bao, H. Robust monocular SLAM in dynamic environments. In Proceedings of the 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Adelaide, SA, Australia, 1–4 October 2013; pp. 209–218.
84. Li, S.; Lee, D. RGB-D SLAM in dynamic environments using static point weighting. *IEEE Robot. Autom. Lett.* **2017**, *2*, 2263–2270. [CrossRef]

85. Alcantarilla, P.F.; Yebes, J.J.; Almazán, J.; Bergasa, L.M. On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 1290–1297.
86. Bescos, B.; Fácil, J.M.; Civera, J.; Neira, J. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robot. Autom. Lett.* **2018**, *3*, 4076–4083. [CrossRef]
87. Wang, C.C.; Thorpe, C.; Thrun, S.; Hebert, M.; Durrant-Whyte, H. Simultaneous localization, mapping and moving object tracking. *Int. J. Robot. Res.* **2007**, *26*, 889–916. [CrossRef]
88. Salas-Moreno, R.F.; Newcombe, R.A.; Strasdat, H.; Kelly, P.H.; Davison, A.J. Slam++: Simultaneous localisation and mapping at the level of objects. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1352–1359.
89. Newcombe, R.A.; Fox, D.; Seitz, S.M. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 343–352.
90. Runz, M.; Buffier, M.; Agapito, L. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In Proceedings of the 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Munich, Germany, 16–20 October 2018; pp. 10–20.
91. Huang, J.; Yang, S.; Zhao, Z.; Lai, Y.K.; Hu, S.M. Clusterslam: A slam backend for simultaneous rigid body clustering and motion estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5875–5884.
92. Huang, J.; Yang, S.; Mu, T.J.; Hu, S.M. ClusterVO: Clustering moving instances and estimating visual odometry for self and surroundings. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2168–2177.
93. Bescos, B.; Campos, C.; Tardós, J.D.; Neira, J. DynaSLAM II: Tightly-coupled multi-object tracking and SLAM. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5191–5198. [CrossRef]
94. Zhang, J.; Henein, M.; Mahony, R.; Ila, V. VDO-SLAM: A visual dynamic object-aware SLAM system. *arXiv* **2020**, arXiv:2005.11052.
95. Li, D.; Yang, W.; Shi, X.; Guo, D.; Long, Q.; Qiao, F.; Wei, Q. A visual-inertial localization method for unmanned aerial vehicle in underground tunnel dynamic environments. *IEEE Access* **2020**, *8*, 76809–76822. [CrossRef]
96. Kelly, J.; Saripalli, S.; Sukhatme, G.S. Combined visual and inertial navigation for an unmanned aerial vehicle. In *Field and Service Robotics: Results of the 6th International Conference*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 255–264.
97. Lin, Y.; Gao, F.; Qin, T.; Gao, W.; Liu, T.; Wu, W.; Yang, Z.; Shen, S. Autonomous aerial navigation using monocular visual-inertial fusion. *J. Field Robot.* **2018**, *35*, 23–51. [CrossRef]
98. Mourikis, A.I.; Roumeliotis, S.I. A multi-state constraint Kalman filter for vision-aided inertial navigation. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007; pp. 3565–3572.
99. Bloesch, M.; Omari, S.; Hutter, M.; Siegwart, R. Robust visual inertial odometry using a direct EKF-based approach. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 298–304.
100. Leutenegger, S.; Furgale, P.; Rabaud, V.; Chli, M.; Konolige, K.; Siegwart, R. Keyframe-based visual-inertial slam using nonlinear optimization. In Proceedings of the Robotis Science and Systems (RSS) 2013, Berlin, Germany, 24–28 June 2013.
101. Mur-Artal, R.; Tardós, J.D. Visual-inertial monocular SLAM with map reuse. *IEEE Robot. Autom. Lett.* **2017**, *2*, 796–803. [CrossRef]
102. Qin, T.; Li, P.; Shen, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [CrossRef]
103. Von Stumberg, L.; Usenko, V.; Cremers, D. Direct sparse visual-inertial odometry using dynamic marginalization. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 2510–2517.
104. Von Stumberg, L.; Cremers, D. Dm-vio: Delayed marginalization visual-inertial odometry. *IEEE Robot. Autom. Lett.* **2022**, *7*, 1408–1415. [CrossRef]
105. Henry, P.; Krainin, M.; Herbst, E.; Ren, X.; Fox, D. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *Int. J. Robot. Res.* **2012**, *31*, 647–663. [CrossRef]
106. Huang, A.S.; Bachrach, A.; Henry, P.; Krainin, M.; Maturana, D.; Fox, D.; Roy, N. Visual odometry and mapping for autonomous flight using an RGB-D camera. In *Robotics Research: The 15th International Symposium ISRR*; Springer: Cham, Switzerland, 2017; pp. 235–252.
107. Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. *Adv. Neural Inf. Process. Syst.* **2014**, 2366–2374.
108. Eigen, D.; Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2650–2658.
109. Liu, F.; Shen, C.; Lin, G. Deep convolutional neural fields for depth estimation from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5162–5170.
110. Laina, I.; Rupprecht, C.; Belagiannis, V.; Tombari, F.; Navab, N. Deeper depth prediction with fully convolutional residual networks. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 239–248.

111. Ma, F.; Karaman, S. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 4796–4803.
112. Ma, F.; Cavalheiro, G.V.; Karaman, S. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 3288–3295.
113. Wofk, D.; Ma, F.; Yang, T.J.; Karaman, S.; Sze, V. Fastdepth: Fast monocular depth estimation on embedded systems. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 6101–6108.
114. Dong, X.; Garratt, M.A.; Anavatti, S.G.; Abbass, H.A. Mobilexnet: An efficient convolutional neural network for monocular depth estimation. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 20134–20147. [CrossRef]
115. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* **2021**, *65*, 99–106. [CrossRef]
116. Sucar, E.; Liu, S.; Ortiz, J.; Davison, A.J. iMAP: Implicit mapping and positioning in real-time. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 6229–6238.
117. Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M.R.; Pollefeys, M. Nice-slam: Neural implicit scalable encoding for slam. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12786–12796.
118. Kerbl, B.; Kopanas, G.; Leimkühler, T.; Drettakis, G. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* **2023**, *42*, 1–14. [CrossRef]
119. Sturm, J.; Burgard, W.; Cremers, D. Evaluating egomotion and structure-from-motion approaches using the TUM RGB-D benchmark. In Proceedings of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS), Vilamoura-Algarve, Portugal, 7–12 October 2012; Volume 13.
120. Zhao, M.; Guo, X.; Song, L.; Qin, B.; Shi, X.; Lee, G.H.; Sun, G. A general framework for lifelong localization and mapping in changing environment. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 3305–3312.