robotics

MDPI

Review

A Review of Visual-Inertial Simultaneous Localization and Mapping from Filtering-Based and Optimization-Based Perspectives

Chang Chen [1,2], Hua Zhu [1,2,*], Menggang Li [1,2], and Shaoze You [1,2]

[1] University School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou 221700, China; cumtchenchang@163.com or chenchang@cumt.edu.cn (C.C.); sallylmg@cumt.edu.cn (M.L.); youshaoze@cumt.edu.cn (S.Y.)

[2] Jiangsu Collaborative Innovation Center of Intelligent Mining Equipment, Xuzhou 221700, China

* Correspondence: zhuhua83591917@163.com; Tel.: +86-0516-8359-1917

Received: 6 July 2018; Accepted: 10 August 2018; Published: 15 August 2018

Abstract: Visual-inertial simultaneous localization and mapping (VI-SLAM) is popular research topic in robotics. Because of its advantages in terms of robustness, VI-SLAM enjoys wide applications in the field of localization and mapping, including in mobile robotics, self-driving cars, unmanned aerial vehicles, and autonomous underwater vehicles. This study provides a comprehensive survey on VI-SLAM. Following a short introduction, this study is the first to review VI-SLAM techniques from filtering-based and optimization-based perspectives. It summarizes state-of-the-art studies over the last 10 years based on the back-end approach, camera type, and sensor fusion type. Key VI-SLAM technologies are also introduced such as feature extraction and tracking, core theory, and loop closure. The performance of representative VI-SLAM methods and famous VI-SLAM datasets are also surveyed. Finally, this study contributes to the comparison of filtering-based and optimization-based methods through experiments. A comparative study of VI-SLAM methods helps understand the differences in their operating principles. Optimization-based methods achieve excellent localization accuracy and lower memory utilization, while filtering-based methods have advantages in terms of computing resources. Furthermore, this study proposes future development trends and research directions for VI-SLAM. It provides a detailed survey of VI-SLAM techniques and can serve as a brief guide to newcomers in the field of SLAM and experienced researchers looking for possible directions for future work.

Keywords: sensor fusion; robotics; SLAM; navigation; computer vision; localization

1. Introduction

Simultaneous localization and mapping (SLAM) technology was first proposed by Smith [1,2], which was applied in robotics with the goal of building a real-time map of surroundings based on sensor data in an unknown environment as the sensor positioned itself. Over the years, new methods have appeared using different sensors such as sonar [3], lidar [4], and cameras [5]. These methods created new data representations and consequently new maps. Durrant-Whyte and Bailey [6,7] systematically reviewed SLAM technologies. Due to recent advances in CPU and GPU technologies, visual SLAM methods have seen increased interest because of the rich visual information available from low-cost cameras compared to other sensors. There are many excellent visual SLAM methods that have improved the development of SLAM technologies, such as MonoSLAM [5], PTAM [8], RatSLAM [9],DTAM [10], KinectFusion [11], and ORB-SLAM [12]. SLAM technology has undergone three major iterations over the last 30 years [13]. Today, SLAM technology is thriving and robust; real-time, high-precision SLAM technology is urgently needed in robotics.

Visual-inertial simultaneous localization and mapping (VI-SLAM) that fuses camera and IMU data for localization and environmental perception has become increasingly popular for several reasons. First, the technology is used in robotics, especially in extensive research and applications involving the autonomous navigation of micro aerial vehicles (MAV). Second, augmented reality (AR) and virtual reality (VR) are growing rapidly. Third, unmanned technology and artificial intelligence has expanded tremendously.

VI-SLAM is generally divided into two approaches: filtering-based and optimization-based. Maplab [14,15] and VINS-mono [16–18] are typical of these two methods, and both are open source. Maplab is a filtering-based VI-SLAM system that also provides the research community with a collection of multi-session mapping tools including map merging, loop closure, and visual-inertial optimization. VINS-mono is a real-time optimization-based VI-SLAM system that uses a sliding window to provide high-precision odometry. Furthermore, it features efficient IMU pre-integration with bias correction, automatic estimator initialization, online extrinsic calibration, failure detection, and loop detection.

Much research has been conducted on SLAM over the last few decades, including reviews and tutorials. A classic review was [6,7]; however, they do not reflect the more recent and emerging SLAM technology. Most reviews [19–23] have also focused solely on visual SLAM or visual odometry without addressing VI-SLAM technology. This study, therefore, provides an overview of VI-SLAM technology from filtering-based and optimization-based perspectives. Feature extraction and tracking, core theory, and loop closure are proposed, which are key technologies in VI-SLAM methods. This work also summarizes research over the previous 10 years and famous VI-SLAM datasets and compares filtering-based and optimization-based methods through experiments. Finally, potential development trends and forthcoming research directions are introduced.

2. Filtering-Based Methods

VI-SLAM approaches can also be further categorized into either loosely or tightly coupled according to sensor fusion type. State-of-the-art VI-SLAM studies over the last 10 years are listed in Table 1. This study divides VI-SLAM methods into filtering-based and optimization-based approaches, mainly according to their back-end optimization type. The loosely coupled method [24,25] usually only fuses the IMU to estimate the orientation and possible the change in position, but not the full pose. In contrast, the tightly coupled method [26,27] fuses the state of the camera and IMU together into a motion and observation equation, and then performs state estimation. Tightly coupled methods presently constitute the main research focus, thanks to advances in computer technology.

Table 1. State-of-the-art visual-inertial simultaneous localization and mapping (VI-SLAM) methods.

Year	Paper	Back-End Approach	Camera Type	Fusion Type	Application
2007	MSCKF [28]	filtering-based	monocular	tightly coupled	
2007	[29]	filtering-based	monocular	tightly coupled	
2010	[30]	filtering-based	stereo	loosely coupled	
2011	[31]	filtering-based	monocular	tightly coupled	vehicle
2011	[32]	filtering-based	monocular	tightly coupled	
2011	[24,25]	filtering-based	monocular	loosely coupled	
2011	[33]	filtering-based	monocular	loosely coupled	MAV
2012	[27]	filtering-based	monocular	tightly coupled	vehicle
2012	[34]	filtering-based	monocular	loosely coupled	
2012	[35]	filtering-based	stereo	tightly coupled	
2013	[36]	filtering-based	monocular	tightly coupled	vehicle
2013	[37]	filtering-based	monocular	loosely coupled	
2013	[38]	filtering-based	monocular	loosely coupled	MAV
2013	[39]	filtering-based	monocular	loosely coupled	
2013	[40]	filtering-based	rgb-d	tightly coupled	
2014	[41]	filtering-based	monocular	tightly coupled	mobile phone

2014	[42]	filtering-based	stereo	tightly coupled	
2015	OKVIS [43–45]	optimization-based	monocular	tightly coupled	
2015	SR-ISWF [46]	filtering-based	monocular	tightly coupled	mobile phone
2015	[47]	optimization-based	monocular	tightly coupled	
2015	[48]	optimization-based	Stereo	tightly coupled	MAV
2015	[49]	optimization-based	rgb-d	loosely coupled	Mobile devices
2015	[50]	filtering-based	monocular	tightly coupled	
2015	ROVIO [51]	filtering-based	monocular	tightly coupled	UAV
2015	[52]	optimization-based	monocular	tightly coupled	autonomous vehicle
2015	[53]	filtering-based	stereo	tightly coupled	
2015	[54]	optimization-based	stereo	tightly coupled	
2016	[55]	optimization-based	monocular	tightly coupled	
2016	[56]	optimization-based	stereo	tightly coupled	
2016	[57]	filtering-based	monocular	loosely coupled	robot
2016	[58]	optimization-based	rgb-d	loosely coupled	
2016	[59]	filtering-based	stereo	loosely coupled	
2016	VIORB [60]	optimization-based	monocular	tightly coupled	MAV
2017	[61]	optimization-based	rgb-d	tightly coupled	
2017	[62]	filtering-based	monocular	loosely coupled	AR/VR
2017	[63]	filtering-based	Multi-camera	tightly coupled	MAV
2017	[64]	filtering-based	monocular	tightly coupled	UAV
2017	VINS-mono [16–18]	optimization-based	monocular	tightly coupled	MAV,AR
2017	[65]	optimization-based	monocular	tightly coupled	AR
2017	[66]	optimization-based	monocular	tightly coupled	
2017	[67]	filtering-based	monocular	tightly coupled	MAV
2017	VINet [68]	end-to-end	monocular	/	deep-learning
2017	[69]	optimization-based	event camera	tightly coupled	
2017	S-MSCKF [26]	filtering-based	stereo	tightly coupled	MAV
2017	[70]	optimization-based	monocular	tightly coupled	MAV
2017	[71]	optimization-based	stereo\monocular	tightly coupled	
2017	PIRVS [72]	filtering-based	stereo	tightly coupled	robot
2017	Maplab [14,15]	filtering-based	monocular	tightly coupled	mobile platform
2018	[73]	optimization-based	stereo	tightly coupled	mobile robot
2018	[74]	optimization-based	stereo	tightly coupled	

2.1. Feature Extraction and Tracking

2.1.1. Feature Extraction

Tracking is an important component in VI-SLAM systems, which depends on the tracking camera pixel. VI-SLAM tracking strategies are presented on Table 2.

Table 2. VI-SLAM tracking strategies.

Methods	Strategies		Papers
1	feature extraction	descriptor matching	[28,60]
2	feature extraction	filter-based tracking	[75]
3	feature extraction	optical flow tracking	[26,76]
4	direct pixel processing		[56,77]

Feature detection aims to identify features and determine their position in an image. Features used in VI-SLAM are mainly Harris [78], FAST [79], ORB [80], SIFT [81], and SURF [82]. Feature detection uses descriptors to describe the keypoint neighborhoods. The ways to obtain features in the image are summarized at several points: (1) the pixel point corresponding to the local maximum of the first derivative, (2) the intersection point of two or more edges, (3) the point where the rate change of the gradient value and gradient direction is high, and (4) the point at which the first derivative at the corner point is the largest and the second derivative is zero.

Brito [83] evaluated the application of different state-of-the-art methods for interest point matching, including SURF, SIFT, ORB, BRISK, and FREAK, aiming for the projective reconstruction of three-dimensional scenes. New features have also been incorporated into the SLAM system, such as the planar feature [84,85], line, or edge feature [86–88]. Importantly, Yang [85] translated monocular sequences to the 3D plane map and proposed semantic monocular plane SLAM for low-texture environments.

2.1.2. Feature Tracking

There are four commonly used methods to track pixel in SLAM systems: descriptor matching [28], filter-based tracking [75], optical flow tracking [26], and direct pixel processing [77]. The principle of the descriptor and feature is the same. Filter-based tracking includes the Kalman filter, particle filter, and mean-shift method. These methods model the target area in the current frame, and predict position by finding the most similar area to the model in the next frame. Optical flow is an effective means of estimating the movement state, such as velocity, pose, and displacement during navigation. Optical flow relates to the apparent movement in the image brightness mode and expresses an image change.

Optical flow can also be divided into three methods depending on the type of calculation, namely the difference [89], correlation [90], and phase-based methods [91]. Among these, the block-matching algorithm is most commonly used in SLAM. However, it has shortcomings, such as a lack of sub-pixel accuracy and reduction of the matching degree after image deformation. To solve these problems, an image pyramid is applied simultaneously to increase computing speed [92].

2.2. Dynamic and Observational Models

The filtering-based SLAM method uses linear or nonlinear models in dynamic and observation models. However, the nonlinear model is mainly used in the filtering-based VI-SLAM method, whose dynamic model is expressed as

$$x_t = f(x_{t-1}, u_t) + w_t \tag{1}$$

where u_t is the control vector, w_t is the process noise, and $w_t \sim N(0, Q_t)$, Q_t is the variance. The IMU status is expressed as a 16-dimension vector.

$$x_{\mathrm{I}} = [\,{}^{\mathrm{I}}_{\mathrm{W}}\bar{q}^{\mathrm{T}} \;\; {}^{\mathrm{W}}p_{\mathrm{I}}^{\mathrm{T}} \;\; {}^{\mathrm{W}}v_{\mathrm{I}}^{\mathrm{T}} \;\; b_g^{\mathrm{T}} \;\; b_a^{\mathrm{T}}\,]^{\mathrm{T}} \tag{2}$$

where ${}^{\mathrm{I}}_{\mathrm{W}}\bar{q}^{\mathrm{T}}$ is the quaternion rotated from the world frame to the IMU frame, and ${}^{\mathrm{W}}p_{\mathrm{I}}^{\mathrm{T}}$ and ${}^{\mathrm{W}}v_{\mathrm{I}}^{\mathrm{T}}$ correspond to the rotation and speed of the world coordinate system, respectively. $b_g{}^{\mathrm{T}}$ and $b_a{}^{\mathrm{T}}$ correspond to the gyroscope bias and accelerometer bias, respectively.

The classic filtering-based method framework is shown in Table 3. Propagation and update steps are important to filtering-based methods. The non-linear observation and prediction equation model are expressed as

$$z_t = h(x_t) + n_t \quad x_{t|t-1} = f(x_{t-1}, u_t) \tag{3}$$

The work of filtering-based VIO focuses mainly on the covariance matrix, feature processing, and EKF updates. The propagated covariance matrix is expressed as

$$P_{t|t-1} = F_t P_{t-1} F_t^T + Q_t \quad F_t = \frac{\partial f}{\partial x}|_{x_t, u_t} \tag{4}$$

The update equations are expressed as

$$y_t = z_t - h(x_{t|t-1}) \quad S_t = H_t P_{t|t-1} H_t^T + R_t \quad H_t = \frac{\partial h}{\partial x}|_{x_t} \tag{5}$$

Table 3. Classic filtering-based method framework.

Propagation: For each IMU measurement received, propagate the filter state and covariance
Image registration: Every time a new image is recorded.
augment the state and covariance matrix with a copy of the current camera pose estimate
image processing modules begins operation
Update: When the feature measurements of a given image become available, perform an EKF update

2.3. Filtering-Based VIO and VI-SLAM

MSCKF [28] is a classic VI-SLAM system. It is also a visual inertial navigation system based on the multi-state constraint EKF. It employs a measurement model to express the geometric constraints that arise when a static feature is observed from multiple camera poses. The algorithm extracts and matches the SIFT feature, and maintains 30 camera poses in the filter state.

In addition, Li [27,36] proved that the standard method of computing Jacobian matrixes in filters inevitably resulted in inconsistencies and a loss of accuracy through simulation tests, which showed that the yaw errors of the MSCKF and FLS [93] lay outside the ±3σ bounds indicating inconsistencies. Thus they proposed modifications to the MSCKF algorithm, which ensure the correct observability properties without incurring additional computational costs. Clement [53] compared MSCKF and the sliding window filter (SWF). Its results showed the SWF to be more accurate and less sensitive to tuning parameters than the MSCKF. However, the MSCKF is computationally cheaper, has good consistency properties, and improves in accuracy as more features are tracked. In contrast to feature-based methods, Tanskanen [50] combined the advantages of EKF filters and minimized photometric errors to propose a direct VIO using only CUP. Increasing studies also began to apply VI-SLAM technologies to small devices such as mobile phones and cleaning robots [41,46].

Bloesch [51] proposed a monocular VIO-ROVIO (https://github.com/ethz-asl/rovio), used to directly detect luminosity error to obtain accurate, robust tracking from image matching. The model also uses the FAST corner to recognize candidate feature regions. A multi-layer image pyramid is used to extract multi-layer features with edge features added. The work process of the filter feature is shown in Figure 1.

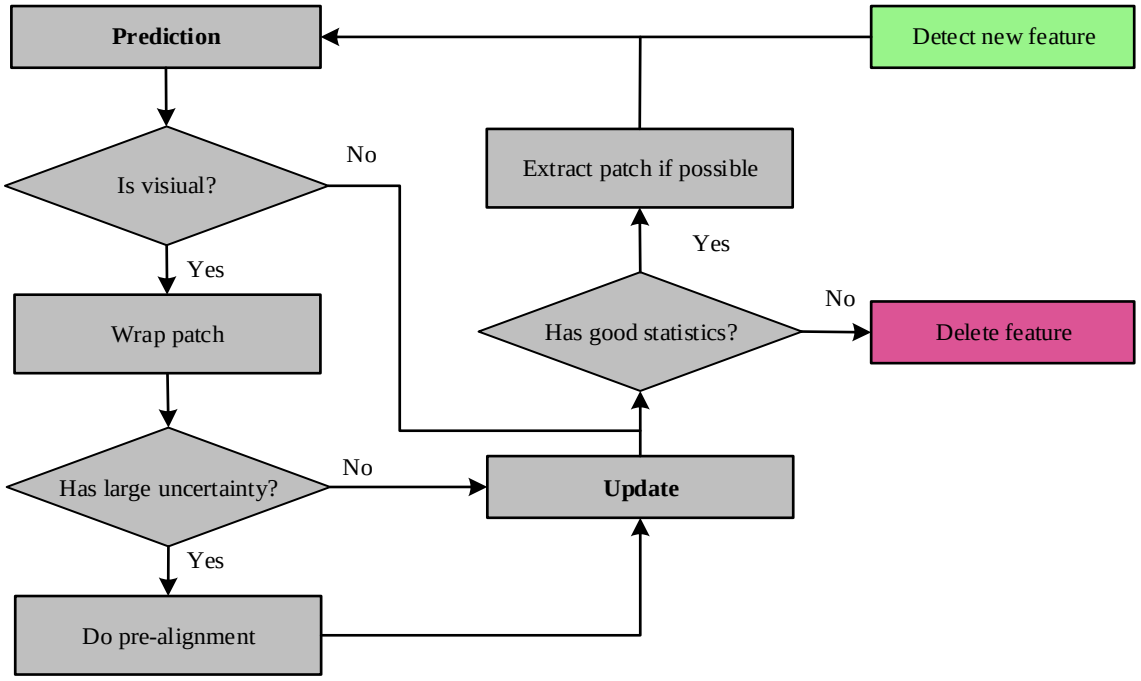

Figure 1. Work process of the filter feature, reproduced from [51].

For image pyramid I_l and multi-layer image block feature with coordinates p and block P_l, the photometric error of block pixel p_j at pyramid l is shown as

$$e_{l,j} = P_l(p_j) - I_l(ps_l + Wp_j) - m \quad (6)$$

where W is the radiation enhancement transformation matrix, and m is the mean intensity error. The average image processing time with 50 features at initialization is 29.72 ms, while the system can run smoothly at 20 Hz. Furthermore, a VIO based on an iterative extended Kalman filter was proposed [63].

S-MSCKF (https://github.com/KumarRobotics/msckf_vio) [26] can be considered a stereo version of MSCKF. The software takes synchronized stereo images and IMU messages and generates a real-time 6DOF camera pose estimation. It uses the FAST corner [79] to increase the speed and tracked features with KLT optical flow [94]. In addition, circular matching can be used to remove outliers generated during feature tracking and stereo matching. It is hard to compare these VI-SLAM methods using only accuracy, due to their different application platforms and sceneries. Therefore, this study surveys representative filtering-based and optimization-based VI-SLAM methods in Appendix A.

Robust and accurate state estimation in robotics remains challenging. If the system can obtain accurate pose estimation based on a prior map, then system adaptability will improve. Therefore, Schneider [15] proposed a VI-SLAM system called Maplab that includes integrated functions of creating, processing and blending multiple maps. The system extensibility is suitable for research, and provided the evaluation method for the selection of system mining components. In addition, Maplab has been found to extract BRISK [95] and FREAK [96] from the image and fuses IMU data for localization and mapping. Separate sections can be combined into a single global map to correct drift for odometry and localization. ROVIOLI [63] is the front-end of Maplab for localization and mapping; the system module and data flow are present in Figure 2. The matching window has been shown to improve efficiency and robustness based on integrated gyroscope measurements. This system easily extends new algorithms in the current framework, such as multithreaded map building, semantic SLAM, and positioning.

Methods combining the advantages of filtering-based and optimization-based approaches have also drawn wide attention. Quan [97] proposed a monocular VI-SLAM using a Kalman filter as an assistant. To enable place recognition and reduce trajectory estimation drift, the authors constructed a factor-graph-based nonlinear optimization in the back-end. A feedback mechanism was used to guarantee estimation accuracy of the front-end and back-end.

The continuous updating and maintenance of maps in a large scale environment is still a challenge. It is particularly essential for platforms that work in repetitive scenarios or use previous maps, such as inspection robots and driverless cars. To update the map according to the dynamic changes and new explored areas, Labbé [98] employed a memory management mechanism into the SLAM system, which identified locations that should remain in fast access memory for online processing from locations.

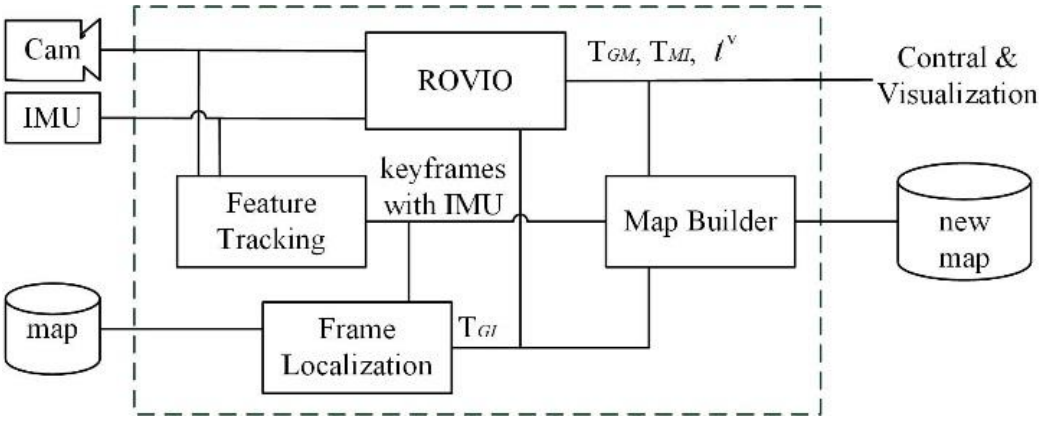

Figure 2. ROVIOLI modules and data flows, reproduced from [15].

3. Optimization-Based Methods

With the development of computer technology, optimization-based VI-SLAM has proliferated rapidly. Optimization-based methods divide the entire SLAM frame into a front-end and back-end according to image processing; the front-end is responsible for map construction, whereas the back-end is responsible for pose optimization. Back-end optimization techniques are usually implemented on g2o [99], ceres-solver [100], and gtsam [101]. Many excellent datasets can be used to study visual-inertial methods, such as EuRoC [102], Canoe [103], Zurich urban MAV [104], TUM VI Benchmark [105], and PennCOSYVIO [106]. Details of the study surveys are provided in Appendix B.

3.1. Loop Closure

Loop closure can detect whether the robot re-enters at the same location; and can determine whether the robot returns to a previously visited location, thus creating a loop in its trajectory. Loop closure also optimizes the entire circuit map and increases system positioning accuracy.

Loop closure methods are mainly classified into odometry-based geometric relationship and appearance-based approaches. The odometry-based geometric relationship approach does not work when the cumulative error is large [107]. The appearance-based approach determines the loop closure relationship to eliminate the cumulative error according to the similarity of two images, and it has been used successfully in VI-SLAM systems [18,31,60].

As shown in Figure 3, the camera data in the VI-SLAM is image-processed to match the spot stored in the map, and a position recognition decision is made after successful matching. The storage map is then updated.

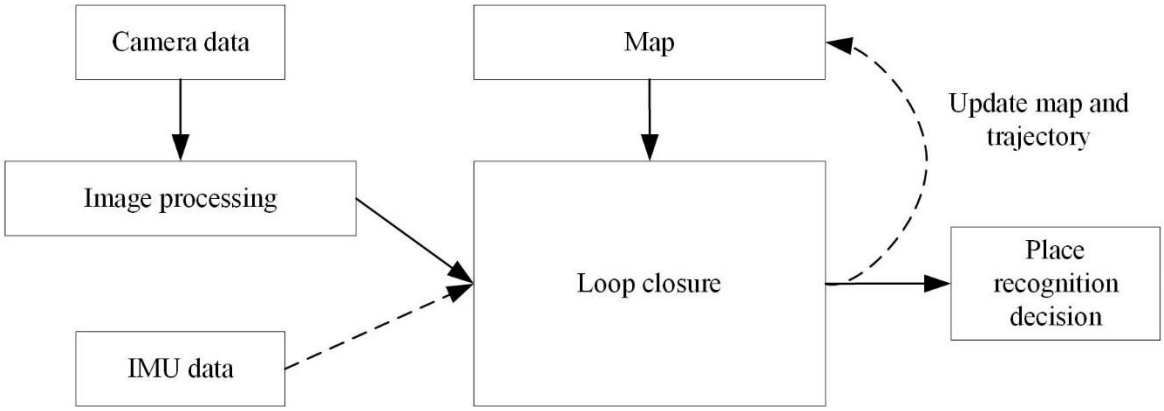

Figure 3. Loop closure schematic, reproduced from [108].

Loop closure is essentially a matter of scene recognition, which is a difficult because of different appearances in various places in the real world. To solve this problem, Galvez-López [109] proposed DBoW2 to obtain a binary bag model with BRIEF and FAST features. Although this algorithm was more efficient and robust in terms of feature extraction compared to those using SIFT or SURF, the BRIEF descriptor lacks rotation and scale invariance, and it can only be used in 2D environments. To address this issue, Mur-Artal [12] used a bag-of-words model of location recognition based on DBoW2 and ORB that included covisibility information.

Loop closure methods based on deep learning continue to emerge [110–112]. Compared with the appearance-based method, they were more robust to environmental changes. However, designing a neural network architecture to run in real-time in a VI-SLAM system remains challenging. In the robotic area coverage problem, the goal is to explore and map a given target area within a reasonable amount of time, which necessitates the use of minimally redundant overlap trajectories for coverage efficiency. However, system estimates will inevitably drift over time in the absence of loop closures. Efficient area coverage and good SLAM navigation performance represent competing objectives. In this case, active SLAM algorithm is needed that accounts for the area

coverage and navigation uncertainty performance to efficiently explore a target area of interest [113]. Thrun [114] found a balance between visiting new places (exploration) and reducing the uncertainty by re-visiting known areas (exploitation), providing a more efficient alternative with respect to random exploration or pure exploitation.

3.2. Optimization-Based VI-SLAM Algorithms

OKVIS (https://github.com/ethz-asl/okvis) [43–45] was an excellent keyframe-based VI-SLAM system; that combined the IMU and reprojection error terms into a cost function to optimize the system. The old keyframes were marginalized to maintain a bounded-sized optimization window, ensuring real-time operation. As a first step to initialization and matching, they propagated the last pose using acquired IMU measurements to obtain a preliminary uncertain estimate of the states. Optimization strategies of optimization-based VI-SLAM algorithms are surveyed in Table 4.

Table 4. Optimization strategies of optimization-based VI-SLAM algorithms.

Methods	Optimization Function	Initialization	Optimization Strategies
OKVIS	reprojection error and IMU temporal error term	using IMU measurements to obtain a preliminary uncertain estimate of the states	Gauss-Newton algorithm Schur complement sliding window
Paper [56]	photometric error and IMU non-linear error terms	initialize the depth map with the propagated depth from the previous keyframe	Levenberg-Marquardt algorithm Schur complement partial marginalization
Paper [55]	photometric error and IMU inertial residual	/	Gauss-Newton algorithm Schur complement
VIORB	reprojection error of all matched points and IMU error term	using vision first, than initializing scale, gravity direction, velocity, and accelerometer and gyroscope biases	Gauss-Newton algorithm local bundle adjustment in local mapping
VINS-mono	reprojection error and IMU residual	using loosely-coupled sensor fusion method get initial values, than aligning metric IMU pre-integration with the visual-only SfM results to recover scale, gravity, velocity, and even bias	Gauss-Newton algorithm Schur complement sliding window two-way marginalization scheme

To avoid repeated constraints caused by the parameterization of relative motion integration, pre-integration was proposed to reduce computation. This method was first described by Lupton [35], where IMU data were changed between two frames by pre-integrating the constraints. The pre-integration principle is illustrated in Figure 4. The pre-integration theory was further developed after Forster [47] applied it to the VI-SLAM framework to reduce bias.

Systems that fused IMU data into the classic visual SLAM also garnered widespread attention. Usenko [56] proposed a stereo direct VIO that combined IMU and stereo LSD-SLAM [115]. They formulated a joint optimization problem to recover the full state containing camera pose, translational velocity, and IMU biases of all frames. Concha [55] devised the first direct tightly coupled VIO algorithm that could run in real-time under a standard CPU, but initialization was not introduced.

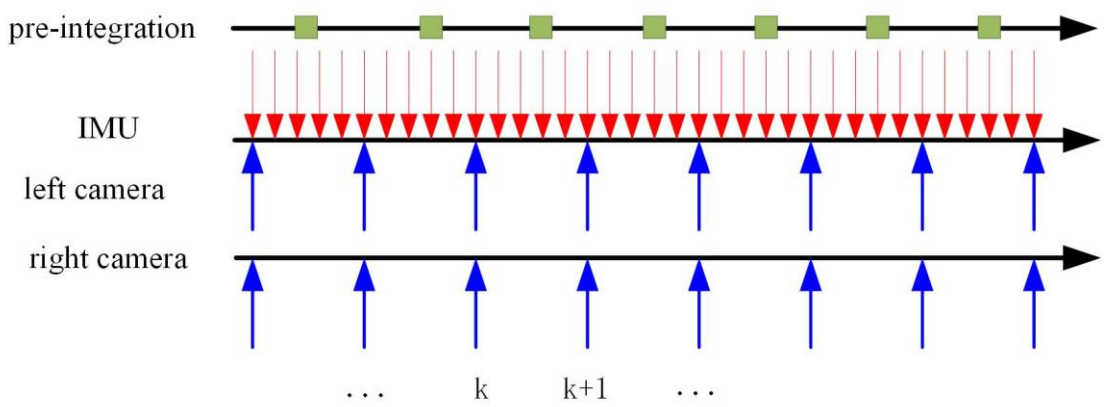

Figure 4. Pre-integration principle, reproduced from [47].

VIORB [60] is a monocular tightly coupled VI-SLAM based on ORB-SLAM and contains an ORB sparse front-end, graph optimization back-end, loop closure, and relocation. This method was first initialized using only monocular vision, and performed a specific initialization of the scale, gravity direction, velocity, and accelerometer and gyroscope biases after a few seconds. VIORB proposed a novel IMU initialization method, which is divided into next four steps: (1) gyroscopes biases estimation, (2) scale and gravity approximation (considering no accelerometer bias), (3) accelerometer biases estimation (scale and gravity direction refinement), and (4) velocity estimation. The local map module uses local BA to optimize the latest N keyframes and all points observed on these N keyframes after a new keyframe is inserted. Local maps are then retrieved based on the time series of the keyframe. The fixed window connects the $N+1$th keyframe and co-visibility graph. The keyframe in the local map is shown in Figure 5. In addition to monocular and IMU fusion methods, SLAM with stereo and RGBD fusion with IMU have also been investigated [54,58].

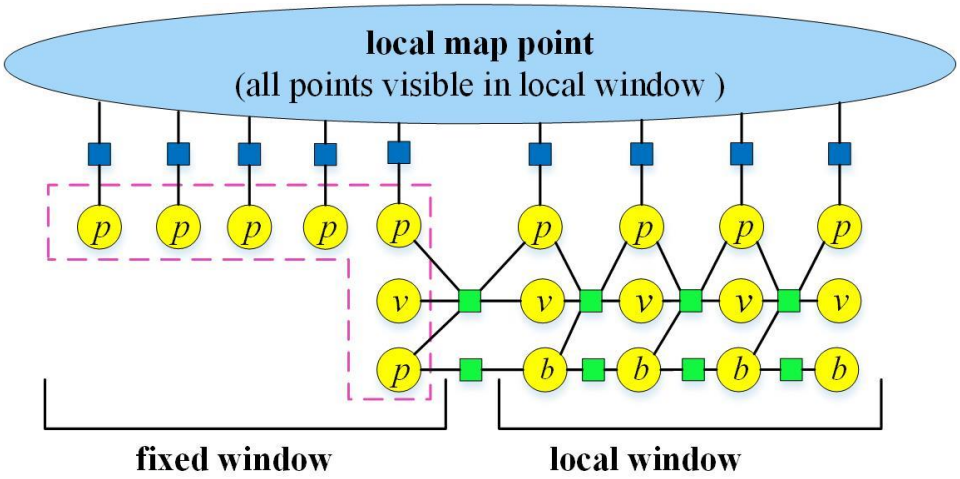

Figure 5. Keyframe in the local map, reproduced from [60].

VINS-mono (https://github.com/HKUST-Aerial-Robotics/VINS-Mono) was a standout VI-SLAM method whose frond-end uses the KLT optical flow [94] to track the Harris corner, while the back-end uses a sliding window for nonlinear optimization. The entire system includes measurement processing, estimation initialization, local bundle adjustment without relocalization, loop closure, and global pose optimization. See Figure 6 for the system framework. The Fisheye camera model is used in the front-end, and an outlier of the fundamental matrix is rejected by the RANSAC method. The calibration error between the camera and IMU is less than 0.02 m, and the rotation error is less than 1° [76]. In addition, this method has been successfully applied to AR [18].

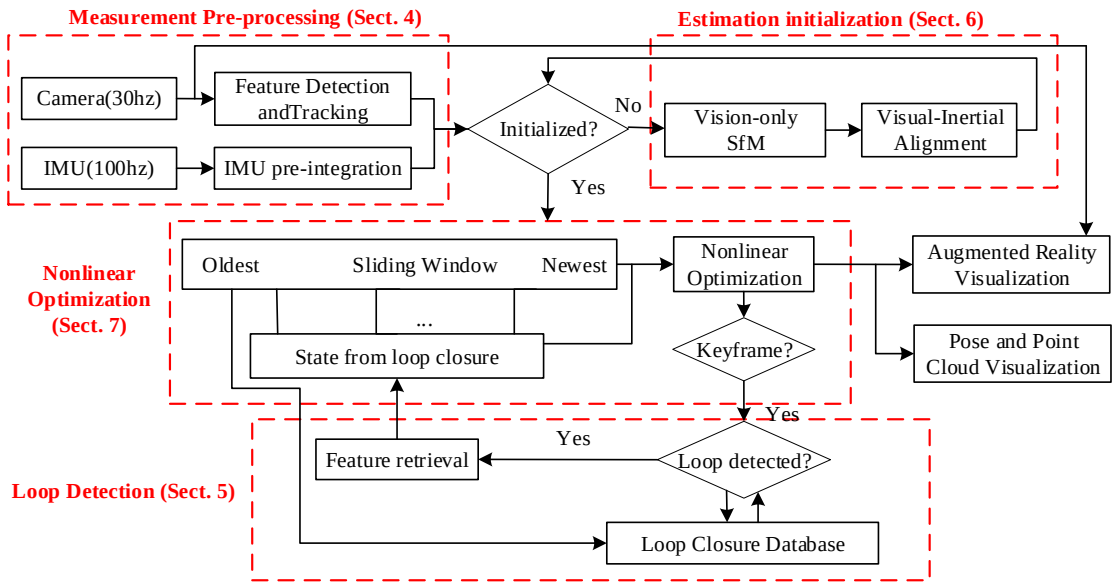

Figure 6. VINS-mono system framework, reproduced from [18].

Additionally, methods integrated with deep learning and new sensors have accompanied the rise of artificial intelligence and computer vision. Clark [68] proposed an end-to-end VIO with good results that combined sensor fusion and depth learning. However, loop closure and mapping were not used in this system. Vidal [69] used event cameras instead of luma frames in VIO to achieve good results in low-light and high-dynamic scenes. CNN-SLAM [116] replaced depth estimation and image matching in LSD-SLAM with CNN-based methods to incorporate semantic information.

4. Comparisons between Filtering-Based and Optimization-Based Methods

4.1. Details

Different VI-SLAM methods are designed for different applications and it is hard to comprehensively evaluate them. To deeply compare filtering-based and optimization-based methods, this section provides the experiments of representative methods on EuRoC datasets using conditions that emulate state estimation for a flying robot. Because VIORB does not have open source code, this study uses an implementation from Jing Wang (https://github.com/jingpang/LearnVIORB).

Experiments are performed on an Intel Core i7-6700×8@3.40GHz computer with 16 Gb RAM. The EuRoC datasets consist of 11 visual inertial sequences recorded onboard a micro-aerial vehicle while it is manually piloted around three different indoor environments. Within each environment, the sequences increase qualitatively in difficulty with increasing sequence number. For example, MH_01 is "easy", while MM_05 is a more challenging sequence in the same environment, introducing things such as faster motions and, poor illumination.

To account for the nondeterministic nature of the multithreading, we run each sequence five times and show the median result for accuracy. In order to compare these methods equally, the mapping thread of VIORB is closed and the camera frequency of all methods is set to 20 Hz.

4.2. Experiments

Experiment results are shown in Tables 5–7. In Table 5, when all eight logical cores are in use, the CPU utilization load is 100%. This study uses the elevation tool evo (https://github.com/MichaelGrupp/evo) to calculate the root mean square error of experiment results according to the ground truth. Notably, VIORB cannot obtain the full trajectory result on the V2_03_difficult dataset. In Table 7, memory utilization is represented as a percentage of the available RAM on the given platform.

Table 5. CPU utilization statistics on VI-SLAM methods (%).

Sequence	ROVIO	S-MSCKF	OKVIS	VINS-Mono	VIORB
MH_01_easy	25.32	33.19	47.32	39.12	30.82
MH_02_easy	26.06	29.01	45.14	39.80	32.34
MH_03_medium	26.53	27.51	49.01	40.48	36.52
MH_04_difficult	25.73	27.91	48.44	40.03	33.07
MH_05_difficult	26.61	29.61	45.74	39.05	36.06
V1_01_easy	27.41	29.59	40.66	41.23	27.82
V1_02_media	27.00	30.61	44.58	35.59	32.44
V1_03_difficult	29.69	30.86	63.30	33.95	31.61
V2_01_easy	27.04	30.83	49.67	37.55	27.55
V2_02_media	26.89	28.29	52.94	36.30	32.07
V2_03_difficult	27.29	27.18	56.74	34.56	32.23
Average	26.87	29.51	49.41	37.97	32.05

Table 6. Root mean square error (RMSE) of VI-SLAM methods (m).

Sequence	ROVIO	S-MSCKF	OKVIS	VINS-Mono	VIORB
MH_01_easy	0.236	0.227	0.164	0.062	0.034
MH_02_easy	0.247	0.231	0.187	0.078	0.049
MH_03_medium	0.427	0.2011	0.274	0.045	0.040
MH_04_difficult	1.170	0.351	0.375	0.134	0.111
MH_05_difficult	0.863	0.213	0.432	0.088	0.269
V1_01_easy	0.216	0.062	0.224	0.045	0.064
V1_02_media	0.210	0.161	0.176	0.045	0.079
V1_03_difficult	0.381	0.281	0.193	0.088	0.212
V2_01_easy	0.298	0.074	0.176	0.057	0.150
V2_02_media	0.232	0.152	0.181	0.114	0.183
V2_03_difficult	0.263	0.366	0.316	0.109	/
Average	0.413	0.211	0.245	0.079	0.119

Table 7. Memory utilization statistics on VI-SLAM methods (%).

Sequence	ROVIO	S-MSCKF	OKVIS	VINS-Mono	VIORB
MH_01_easy	14.86	14.14	11.03	17.09	12.52
MH_02_easy	15.03	14.22	11.22	16.95	12.53
MH_03_medium	15.04	14.22	11.03	17.74	12.48
MH_04_difficult	15.04	14.24	11.10	17.05	12.77
MH_05_difficult	15.29	14.33	11.22	18.61	13.72
V1_01_easy	12.86	14.04	11.28	17.92	12.72
V1_02_media	14.87	13.79	11.63	17.03	12.59
V1_03_difficult	14.30	13.82	11.67	17.80	12.65
V2_01_easy	13.53	14.06	11.69	16.96	12.46
V2_02_media	14.37	14.08	11.81	17.54	12.32
V2_03_difficult	14.82	14.09	12.11	17.26	12.48
Average	14.55	14.04	11.43	17.45	12.65

This section experiments representative optimization-based and filtering-based methods, which are all proposed in recent years. As shown in Table 5, the CPU utilization of ROVIO is the lowest among five methods, and filtering-based methods are better than optimization-based methods. The camera type of ROVIO, VINS-mono, and VIORB is monocular, while the camera type of S-MSCKF and OKVIS is stereo. The stereo VI-SLAM methods use more computing resources than monocular VI-SLAM methods, whether filtering-based or optimization-based. Importantly, filtering-based

methods have advantages over optimization-based methods on CPU utilization. As shown in Table 6, VINS-mono obtains the best accuracy with a 0.079 m average root mean square error. OKVIS and VIORB have advantages in terms of memory utilization (according to Table 7), which implies that they are robust for system management. Optimization-based methods have more potential than filtering-based methods in terms of localization accuracy and memory utilization. In summary, optimization-based methods achieve excellent localization accuracy and lower memory utilization, while filtering-based methods have advantages in terms of computing resource. How to find the right balance between competing requirements and accuracy can be challenging.

5. Development Trends

5.1. SLAM with Deep Learning

At present, the semantic level of the image features used in the SLAM scheme is too low, rendering feature distinguishability weak; the point cloud map constructed by the current method does not distinguish between different objects. Deep learning will develop SLAM technology, which can be used to build semantic maps to advance human computer interaction. Rambach [117] proposed a deep learning approach to visual-inertial camera pose estimation through a trained short-term memory model. Shamwell [118] presented an unsupervised deep neural network approach to the fusion of RGB-D imagery with inertial measurements for absolute trajectory estimation.

Although the study of semantic issues in SLAM is still in a nascent stage, combining semantics with SLAM will enable robots to obtain poses more effectively by building consistent maps using semantic concepts of categories, relationships, and environmental attributes. In addition, a new map of the SLAM system can effectively store and display information, such as SkiMap [119] and Road-SLAM [120]. The continuous updating and maintenance of maps still presents an obstacle in the field.

5.2. Hardware Integration and Multi-Sensor Fusion

The lightweight and miniaturization characteristics of the SLAM system allow it to run well on small devices, such as embedded systems or cell phones. Excellent results were achieved in Microsoft Hololens, Intel RealSense, and Google Tango [121]. Customized hardware for the VI-SLAM can realize the function of robots, and AR/VR devices are applied to sports, navigation, teaching, and entertainment. Therefore, a strong demand exists for SLAM miniaturization and weight reduction, prefacing the future of embedded SLAM [122].

A single sensor cannot adequately sense environmental information, and state estimation is highly uncertain. Multi-sensor fusion can solve these problems and improve the accuracy of system positioning and environment mapping.VI-SLAM technology is an example of multi-sensor fusion. Research and applications involving multi-sensor fusion in SLAM are expected to grow, as evidenced by [123,124].

5.3. Active SLAM on Robots

A pertinent SLAM issue represents a passive estimation problem in robotics. However, the main purpose of controlling the robot motion problem is to control the robot to minimize uncertainty of robotic map representation and positioning. In a conventional approach, SLAM is passive and typically performed on preplanned or human-controlled trajectories. A fully autonomous robot must plan a motion given a high-level command, such as, a task-level command from a human supervisor to explore a given area. In this example, the robot should plan accordingly to accomplish the given task and should not require detailed input by a human supervisor [113]. Active SLAM [125] has therefore attracted gradual attention. The active SLAM algorithm has demonstrated good effects in terms of enabling the robot to identify possible locations, calculate each vantage point visited, and select the most efficient action plan. SLAM technology should thus incorporate technologies such as path planning [126], mission planning [127], and object recognition [128]. References [129,130]

contributed to active SLAM and combine it to make robots more intelligent and practical. In addition, integrating the advantages of different branches of SLAM technology (such as, filtering and optimization-based approaches and loosely and tightly coupled methods) would greatly improve system robustness and accuracy.

5.4. Applications on Complex Dynamic Environments

The SLAM algorithm generally assumes a static environment. However, the actual working environment of the mobile robot often involves changes in the spatial positions of pedestrians and vehicles over time. These dynamic features can provide useful information about environmental changes. Identification of static and dynamic features in the environment and locating and mapping the robot effectively are important. Saarinen [131] made contributions to enabling long-term operation of autonomous vehicles in industrial dynamic environments and proposed a novel 3D normal distribution transform occupancy maps. Additionally (to ensure more effective practical application), seasonal weather changes in unstructured terrain require a more robust SLAM system to handle complex dynamic environments. Multi-robot collaboration SLAM [132] possesses advantages of high accuracy and efficiency, and it is emerging as a common research area.

6. Conclusions

VI-SLAM technology is a popular and complicated research issue in the field of robotics and computer vision. This study provided an overview of VI-SLAM technology and summarized methods over the last 10 years. State-of-the-art VI-SLAM methods are introduced from filtering and optimization-based perspectives. The respective frameworks, key technologies, and advantages of these methods are presented. In addition, central technologies in VI-SLAM are systematically proposed, including feature extraction and tracking, pre-integration, and loop closure. This study surveys the performance of representative VI-SLAM methods and famous VI-SLAM datasets. Comparisons are made between filtering-based and optimization-based methods through experiments, which indicate filtering-based methods have advantages in terms of computing resources, while optimization-based methods achieve excellent localization accuracy and lower memory utilization. This study also predicted upcoming development trends and research directions for SLAM that have the potential to make the technology substantial.

Author Contributions: C.C. designed the architecture and finalized the paper. H.Z. conceived the idea. M.L. and S.Y. did the proof reading.

Funding: This research was supported by grant of the National Key Research and Development Program of China (No. 2018YFC0808003), the National 863 Program of China (No. 2012AA041504) and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

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

Appendix A

This study presents performance of VI-SLAM methods including MSCKF, ROVIO, S-MSCKF, OKVIS, VINS-mono, and VIORB. The performance of these methods is shown in Table A1. The camera type of MSCKF, ROVIO, VINS-mono, and VIORB is monocular. The camera type of S-MSCKF and OKVIS is stereo. Reference [15] proposed an analysis of tightly-coupled monocular, binocular, and stereo visual-inertial odometry. Notably, the drift rate of ROVIO is calculated according to Figure 3 in [51]. VIORB obtained a 0.075 m root mean square error, with a scale error typically below 1%. This method was able to close loops and reuse its map to achieve zero-drift localization in already mapped areas.

Table A1. Performance of representative VI-SLAM methods.

Methods	MSCFK	ROVIO	S-MSCKF	OKVIS	VINS-Mono	VIORB
Platform	**Vehicle**	**UAV**	**MAV**	**Car/Helmet**	**MAV**	**MAV**
Image	640×480 @14Hz	752×480 @20Hz	752×480 @20Hz	752×480 @20Hz	752×480 @20Hz	752×480 @20Hz
Environment	outdoor	indoor	indoor/outdoor	outdoor	indoor/outdoor	indoor
IMU	@100Hz	@200Hz	@200Hz	@200Hz	@100Hz	@200Hz
Drift rate	0.31%	≈1.8%	<0.5%	<0.1%	0.88%	≈0

Appendix B

This study provides more details about SLAM datasets. The comparison of datasets with vision and IMU data is shown in Table A2.

Table A2. Comparison of datasets with vision and IMU data.

Datasets	EuRoC Datasets	PennCOSYVIO	Zurich Urban MAV Dataset	TUM VI Benchmark	Canoe Dataset
Carrier	**MAV**	**Handheld**	**MAV**	**Handheld**	**Canoe**
Cameras	1 stereo gray 2×752×480 (global shutter) @20Hz	4 RGB 1920×1080 @30Hz (rolling shutter), 1 stereo gray 2×752×480 @20Hz, 1 fisheye gray 640×480 @30Hz	1 RGB 1920×1080 @30Hz (rolling shutter)	1 stereo gray 2×1024×1024 (global shutter) @20Hz	1 stereo RGB 2×1600×1200 (rectified 2×600×800) @20Hz
IMUs	ADIS16488 3-axis acc/gyro @200Hz	ADIS16488 3-axis acc/gyro @200Hz, Tango 3-axis acc @128Hz/3-axis gyro @100Hz	3-axis acc/gyro @10Hz	BMI160 3-axis acc/gyro @200Hz	ADIS-16488 3-axis acc/gyro @200Hz,
Environment	indoors	indoor/outdoors	outdoors	indoors/outdoors	Sangamon River
Ground truth	Leica Multistation/Vicon system	fiducial markers	Pix4D	motion capture pose	GPS
Stats/props	11 sequences, 0.9 km	4 sequences, 0.6 km	1 sequence, 2 km	28 sequences, 20 km	28 sequences, 2.7 km

References

1. Smith, R.C.; Cheeseman, P. On the Representation and Estimation of Spatial Uncertainly. *Int. J. Robot. Res.* **1986**, *5*, 56–68.
2. Smith, R.; Self, M.; Cheeseman, P. Estimating Uncertain Spatial Relationships in Robotics. *Mach. Intell. Pattern Recognit.* **1988**, *5*, 435–461.
3. Kleeman, L. Advanced sonar and odometry error modeling for simultaneous localisation and map building. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–8 November 2013; pp. 699–704.
4. Kohlbrecher, S.; Stryk, O.V.; Meyer, J.; Klingauf, U. A flexible and scalable SLAM system with full 3D motion estimation. In Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics, Kyoto, Japan, 1–2 November 2011; pp. 155–160.
5. 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.
6. Durrant-Whyte, H.; Bailey, T. Simultaneous Localization and Mapping: Part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110.
7. Bailey, T.; Durrantwhyte, H. Simultaneous localisation and mapping (slam) part 2: State of the art. *IEEE Robot. Autom. Mag.* **2006**, *13*, 108–117.
8. Klein, G.; Murray, D. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality, Cambridge, UK, 15–18 September 2008; pp. 1–10.

9. Milford, M.J.; Wyeth, G.F.; Prasser, D. RatSLAM: A hippocampal model for simultaneous localization and mapping. In Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–5 June, 2004; pp. 403–408.
10. Newcombe, R.A.; Lovegrove, S.J.; Davison, A.J. DTAM: Dense tracking and mapping in real-time. In Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2320–2327.
11. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohi, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. KinectFusion: Real-time dense surface mapping and tracking. In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality, Atlanta, GA, USA, 5–8 November 2012; pp. 127–136.
12. 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.
13. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.D.; Leonard, J.J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Trans. Robot.* **2016**, 32, 1309–1332.
14. Lynen, S.; Sattler, T.; Bosse, M.; Hesch, J.; Pollefeys, M.; Siegwart, R. Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization. In Proceedings of the Robotics: Science and Systems, Rome, Italy, 13–17 July 2015.
15. Schneider, T.; Dymczyk, M.; Fehr, M.; Egger, K.; Lynen, S.; Gilitschenski, I.; Siegwart, R. maplab: An Open Framework for Research in Visual-inertial Mapping and Localization. *IEEE Robot. Autom. Lett.* **2017, 3,** 1418–1425.
16. Qin, T.; Li, P.; Shen, S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *arXiv* **2017**, arXiv:1708.03852.
17. 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.* **2017**, *35*, 23–51.
18. Li, P.; Qin, T.; Hu, B.; Zhu, F.; Shen, S. Monocular Visual-Inertial State Estimation for Mobile Augmented Reality. In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality, Natnes, France, 9–13 October 2017; pp. 11–21.
19. Scaramuzza, D.; Fraundorfer, F. Visual Odometry [Tutorial]. *IEEE Robot. Autom. Mag.* **2011**, *18*, 80–92.
20. Fraundorfer, F.; Scaramuzza, D. Visual Odometry: Part II: Matching, Robustness, Optimization, and Applications. *IEEE Robot. Autom. Mag.* **2012**, *19*, 78–90.
21. Fuentes-Pacheco, J.; Ruiz-Ascencio, J.; Rendón-Mancha, J.M. Visual simultaneous localization and mapping: A survey. *Artif. Intell. Rev.* **2015**, *43*, 55–81.
22. 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.
23. Paul, M.K.; Wu, K.; Hesch, J.A.; Nerurkar, E.D.; Roumeliotis, S.I. A comparative analysis of tightly-coupled monocular, binocular, and stereo VINS. In Proceedings of the IEEE International Conference on Robotics and Automation, Marina Bay, Singapore, 29 May–3 June 2017; pp. 165–172.
24. Weiss, S.; Siegwart, R. Real-time metric state estimation for modular vision-inertial systems. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 4531–4537.
25. Weiss, S.; Scaramuzza, D.; Siegwart, R. Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments. *J. Field Robot.* **2011**, *28*, 854–874.
26. Sun, K.; Mohta, K.; Pfrommer, B.; Watterson, M.; Liu, S.; Mulgaonkar, Y.; Taylor, C.J.; Kumar, V. Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight. *IEEE Robot. Autom. Lett.* **2018**, *3*, 965–972.
27. Li, M.; Mourikis, A.I. Improving the accuracy of EKF-based visual-inertial odometry. In Proceedings of the IEEE International Conference on Robotics and Automation, St. Paul, USA, 14–18 May 2012; pp. 828–835.
28. Mourikis, A.I.; Roumeliotis, S.I. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In Proceedings of the IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 3565–3572.
29. Veth, M.M.; Raquet, J. Fusing Low-Cost Image and Inertial Sensors for Passive Navigation. *Navigation* **2007**, *54*, 11–20.
30. Tardif, J.P.; George, M.; Laverne, M.; Kelly, A. A new approach to vision-aided inertial navigation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan,

18–22 October 2010; pp. 4161–4168.
31. Jones, E.S.; Soatto, S. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *Int. J. Robot. Res.* **2011**, *30*, 407–430.
32. Kelly, J.; Sukhatme, G.S. Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration. *Int. J. Robot. Res.* **2011**, *30*, 56–79.
33. Achtelik, M.; Achtelik, M.; Weiss, S.; Siegwart, R. Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3056–3063.
34. Weiss, S.M. Vision Based Navigation for Micro Helicopters. Ph.D. Dissertation, ETH Zurich, Zürich, Switzerland, 2012.
35. Lupton, T.; Sukkarieh, S. Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments without Initial Conditions. *IEEE Trans. Robot.* **2012**, *28*, 61–76.
36. Li, M.; Mourikis, A.I. High-precision, consistent EKF-based visual-inertial odometry. *Int. J. Robot. Res.* **2013**, *32*, 690–711.
37. Lynen, S.; Achtelik, M.W.; Weiss, S.; Chli, M. A robust and modular multi-sensor fusion approach applied to MAV navigation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 3923–3929.
38. Sa, I.; He, H.; Huynh, V.; Corke, P. Monocular vision based autonomous navigation for a cost-effective MAV in GPS-denied environments. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Wollongong, Australia, 9–12 July 2013; pp. 1355–1360.
39. Weiss, S.; Achtelik, M.W.; Lynen, S.; Chli, M. Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 957–964.
40. Guo, C.X.; Roumeliotis, S.I. IMU-RGBD camera 3D pose estimation and extrinsic calibration: Observability analysis and consistency improvement. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 2935–2942.
41. Guo, C.; Kottas, D.; Dutoit, R.; Ahmed, A.; Li, R.; Roumeliotis, S. Efficient Visual-Inertial Navigation using a Rolling-Shutter Camera with Inaccurate Timestamps. In Proceedings of the Robotics: Science and Systems, Berkeley, USA, 12–16 July 2014.
42. Asadi, E.; Bottasso, C.L. Tightly-coupled stereo vision-aided inertial navigation using feature-based motion sensors. *Adv. Robot.* **2014**, *28*, 717–729.
43. Leutenegger, S.; Lynen, S.; Bosse, M.; Siegwart, R.; Furgale, P. Keyframe-based visual-inertial odometry using nonlinear optimization. *Int. J. Robot. Res.* **2015**, *34*, 314–334.
44. Leutenegger, S. Unmanned Solar Airplanes: Design and Algorithms for Efficient and Robust Autonomous Operation. Ph.D. Dissertation, ETH Zurich, Zürich, Switzerland, 2014.
45. Leutenegger, S.; Furgale, P.; Rabaud, V.; Chli, M.; Konolige, K.; Siegwart, R. Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization. In Proceedings of the Robotics: Science and Systems, Berkeley, CA, USA, 12–16 July 2014; pp. 789–795.
46. Wu, K.; Ahmed, A.; Georgiou, G.; Roumeliotis, S. A Square Root Inverse Filter for Efficient Vision-aided Inertial Navigation on Mobile Devices. In Proceedings of the Robotics: Science and Systems, Rome, Italy, 13–17 July 2015.
47. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. In Proceedings of the Robotics: Science and Systems, Rome, Italy, 13–17 July 2015.
48. Burri, M.; Oleynikova, H.; Achtelik, M.W.; Siegwart, R. Real-time visual-inertial mapping, re-localization and planning onboard MAVs in unknown environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany, 28 September–2 October 2015; pp. 1872–1878.
49. Brunetto, N.; Salti, S.; Fioraio, N.; Cavallari, T.; Stefano, L.D. Fusion of Inertial and Visual Measurements for RGB-D SLAM on Mobile Devices. In Proceedings of the IEEE International Conference on Computer Vision Workshop, Santiago, Chile, 13–16 December 2015; pp. 148–156.
50. Tanskanen, P.; Naegeli, T.; Pollefeys, M.; Hilliges, O. Semi-direct EKF-based monocular visual-inertial odometry. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany, 28 September–2 October 2015; pp. 6073–6078.

51. Bloesch, M.; Omari, S.; Hutter, M.; Siegwart, R. Robust visual inertial odometry using a direct EKF-based approach. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany, 28 September–2 October 2015; pp. 298–304.
52. Keivan, N.; Patron-Perez, A.; Sibley, G. Asynchronous Adaptive Conditioning for Visual-Inertial SLAM. *Int. J. Robot. Res.* **2015**, *34*, doi:10.1177/0278364915602544.
53. Clement, L.E.; Peretroukhin, V.; Lambert, J.; Kelly, J. The Battle for Filter Supremacy: A Comparative Study of the Multi-State Constraint Kalman Filter and the Sliding Window Filter. In Proceedings of the Computer and Robot Vision, Halifax, NS, Canada, 3–5 June 2015; pp. 23–30.
54. Huai, J.; Toth, C.K.; Grejner-Brzezinska, D.A. Stereo-inertial odometry using nonlinear optimization. In Proceedings of the International Technical Meeting of the Satellite Division of the Institute of Navigation, Tampa, FL, USA, 14–18 September 2015.
55. Concha, A.; Loianno, G.; Kumar, V.; Civera, J. Visual-inertial direct SLAM. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 1331–1338.
56. Usenko, V.; Engel, J.; Stückler, J.; Cremers, D. Direct visual-inertial odometry with stereo cameras. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 1885–1892.
57. Munguía, R.; Nuño, E.; Aldana, C.I.; Urzua, S. A Visual-aided Inertial Navigation and Mapping System. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 94.
58. Falquez, J.M.; Kasper, M.; Sibley, G. Inertial aided dense & semi-dense methods for robust direct visual odometry. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Daejeon, Korea, 9–14 October 2016; pp. 3601–3607.
59. Palézieux, N.D.; Nägeli, T.; Hilliges, O. Duo-VIO: Fast, light-weight, stereo inertial odometry. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Daejeon, Korea, 9–14 October 2016; pp. 2237–2242.
60. Mur-Artal, R.; Tardós, J.D. Visual-Inertial Monocular SLAM with Map Reuse. *IEEE Robot. Autom. Lett.* **2017**, *2*, 796–803.
61. Laidlow, T.; Bloesch, M.; Li, W.; Leutenegger, S. Dense RGB-D-inertial SLAM with map deformations. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, Canada, 24–28 September 2017; pp. 6741–6748.
62. Fang, W.; Zheng, L.; Deng, H.; Zhang, H. Real-Time Motion Tracking for Mobile Augmented/Virtual Reality Using Adaptive Visual-Inertial Fusion. *Sensors* **2017**, *17*, 1037.
63. Bloesch, M.; Burri, M.; Omari, S.; Hutter, M.; Siegwart, R. Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback. *Int. J. Robot. Res.* **2017**, *36*, 1053–1072.
64. Sa, I.; Kamel, M.; Burri, M.; Bloesch, M.; Khanna, R.; Popovic, M.; Nieto, J.; Siegwart, R. Build Your Own Visual-Inertial Drone: A Cost-Effective and Open-Source Autonomous Drone. *IEEE Robot .Autom. Mag.* **2017**, *25*, 89–103.
65. Piao, J.; Kim, S. Adaptive Monocular Visual-Inertial SLAM for Real-Time Augmented Reality Applications in Mobile Devices. *Sensors* **2017**, *17*, 2567.
66. Liu, Y.; Chen, Z.; Zheng, W.; Wang, H.; Liu, J. Monocular Visual-Inertial SLAM: Continuous Preintegration and Reliable Initialization. *Sensors* **2017**, *17*, 2613.
67. Hesch, J.A.; Kottas, D.G.; Bowman, S.L.; Roumeliotis, S.I. Consistency Analysis and Improvement of Vision-aided Inertial Navigation. *IEEE Trans. Robot.* **2017**, *30*, 158–176.
68. Clark, R.; Wang, S.; Wen, H.; Markham, A.; Trigoni, N. VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
69. Vidal, A.R.; Rebecq, H.; Horstschaefer, T.; Scaramuzza, D. Hybrid, Frame and Event based Visual Inertial Odometry for Robust, Autonomous Navigation of Quadrotors. *arXiv* **2017**, arXiv:1709.06310.
70. Yang, Z.; Gao, F.; Shen, S. Real-time monocular dense mapping on aerial robots using visual-inertial fusion. In Proceedings of the IEEE International Conference on Robotics and Automation, Marina Bay, Singapore, 29 May–3 June 2017; pp. 4552–4559.
71. Kasyanov, A.; Engelmann, F.; Stückler, J.; Leibe, B. Keyframe-Based Visual-Inertial Online SLAM with Relocalization. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, BC, Canada, 24–28 September 2017.

72. Zhang, Z.; Liu, S.; Tsai, G.; Hu, H.; Chu, C.C.; Zheng, F. PIRVS: An Advanced Visual-Inertial SLAM System with Flexible Sensor Fusion and Hardware Co-Design. *arXiv* **2017**, arXiv:1710.00893.
73. Chen, C.; Zhu, H. Visual-inertial SLAM method based on optical flow in a GPS-denied environment. *Ind. Robot Int. J.* **2018**, *45*, 401–406.
74. Liu, H.; Chen, M.; Zhang, G.; Bao, H.; Bao, Y. ICE-BA: Incremental, Consistent and Efficient Bundle Adjustment for Visual-Inertial SLAM. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake, UT, USA,18–22 June 2018; pp. 1974–1982.
75. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596.
76. Yang, Z.; Shen, S. Monocular Visual-Inertial State Estimation with Online Initialization and Camera-IMU Extrinsic Calibration. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 39–51.
77. Engel, J.; Koltun, V.; Cremers, D. Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 611–625.
78. Harris, C. A combined corner and edge detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, September, 1988; pp. 147–151.
79. Rosten, E.; Porter, R.; Drummond, T. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 105–119.
80. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the IEEE International Conference on Computer Vision, Toronto, ON, Canada, 27–30 May 2012; pp. 2564–2571.
81. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
82. Bay, H.; Ess, A.; Tuytelaars, T.; Gool, L.V. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359.
83. Brito, D.N.; Nunes, C.F.G.; Padua, F.L.C.; Lacerda, A. Evaluation of Interest Point Matching Methods for Projective Reconstruction of 3D Scenes. *IEEE Lat. Am. Trans.* **2016**, *14*, 1393–1400.
84. Gao, X.; Zhang, T. Robust RGB-D simultaneous localization and mapping using planar point features. *Robot. Autonom. Syst.* **2015**, *72*, 1–14.
85. Yang, S.; Song, Y.; Kaess, M.; Scherer, S. *P*op-up SLAM: Semantic monocular plane SLAM for low-texture environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Daejeon, Korea, 9–14 October 2016; pp. 1222–1229.
86. Kong, X.; Wu, W.; Zhang, L.; Wang, Y. Tightly-Coupled Stereo Visual-Inertial Navigation Using Point and Line Features. *Sensors* **2015**, *15*, 12816–12833.
87. Yang, S.; Scherer, S. Direct monocular odometry using points and lines. In Proceedings of the IEEE International Conference on Robotics and Automation, Marina Bay, Singapore, 29 May–3 June 2017; pp. 3871–3877.
88. Zhang, G.; Lee, J.H.; Lim, J.; Suh, I.H. Building a 3-D Line-Based Map Using a Stereo SLAM. *IEEE Trans. Robot.* **2015**, *31*, 1364–1377.
89. Enkelmann, W. Investigation of multigrid algorithms for the estimation of optical flow fields in image sequences. *Comput. Vis. Graph. Image Process.* **1988**, *43*, 150–177.
90. Hassen, W.; Amiri, H. Block Matching Algorithms for motion estimation. In Proceedings of the IEEE International Conference on E-Learning in Industrial Electronics, Vienna, Austria, 10–13 November 2013; pp. 136–139.
91. Weng, J. A theory of image matching. In Proceedings of the International Conference on Computer Vision, Osaka, Japan, 4–7 December 1990; pp. 200–209.
92. Holmgren, D.E. An invitation to 3-D vision: From images to geometric models. *Photogramm. Rec.* **2004**, *19*, 415–416.
93. Sibley, G.; Matthies, L.; Sukhatme, G. Sliding window filter with application to planetary landing. *J. Field Robot.* **2010**, *27*, 587–608.
94. Baker, S.; Matthews, I. Lucas-Kanade 20 Years On: A Unifying Framework. *Int. J. Comput. Vis.* **2004**, *56*, 221–255.
95. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary Robust invariant scalable keypoints. In Proceedings of the IEEE International Conference on Computer Vision, Toronto, ON, Canada, 27–30 May 2012; pp. 2548–2555.
96. Alahi, A.; Ortiz, R.; Vandergheynst, P. REAK: Fast Retina Keypoint. In Proceedings of the IEEE Conference

on Computer Vision and Pattern Recognition, Rhode Island, USA, 16–21 June 2012; pp. 510–517.
97. Quan, M.; Piao, S.; Tan, M.; Huang, S.S. Map-Based Visual-Inertial Monocular SLAM using Inertial assisted Kalman Filter. *arXiv* **2017**, arXiv:1706.03648v2.
98. Labbé, M.; Michaud, F. Long-term online multi-session graph-based SPLAM with memory management. *Auton. Robot.* **2017**, 42, 1133–1150.
99. Kümmerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; Burgard, W. G2o: A general framework for graph optimization. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3607–3613.
100. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
101. Carlone, L.; Kira, Z.; Beall, C.; Indelman, V. Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors. In Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–7 June2014; pp. 4290–4297.
102. 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.
103. Miller, M.; Chung, S.J.; Hutchinson, S. The Visual-Inertial Canoe Dataset. *Int. J. Robot. Res.* **2018**, *37*, 13–20.
104. Majdik, A.L.; Till, C.; Scaramuzza, D. The Zurich urban micro aerial vehicle dataset. *Int. J. Robot. Res.* **2017**, *36*, 269–273.
105. Schubert, D.; Goll, T.; Demmel, N.; Usenko, V.; Stückler, J.; Cremers, D. The TUM VI Benchmark for Evaluating Visual-Inertial Odometry. *arXiv* **2018**, arXiv:1804.06120.
106. Pfrommer, B.; Sanket, N.; Daniilidis, K.; Cleveland, J. PennCOSYVIO: A challenging Visual Inertial Odometry benchmark. In Proceedings of the IEEE International Conference on Robotics and Automation, Marina Bay, Singapore, 29 May–3 June 2017; pp. 3847–3854.
107. Beeson, P.; Modayil, J.; Kuipers, B. Factoring the Mapping Problem: Mobile Robot Map-building in the Hybrid Spatial Semantic Hierarchy. *Int. J. Robot. Res.* **2010**, *29*, 428–459.
108. Lowry, S.; Sünderhauf, N.; Newman, P.; Leonard, J.J.; Cox, D.; Corke, P.; Milford, M.J. Visual Place Recognition: A Survey. *IEEE Trans. Robot.* **2016**, *32*, 1–19.
109. Galvez-López, D.; Tardos, J.D. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Trans. Robot.* **2012**, *28*, 1188–1197.
110. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the Computer Vision and Pattern Recognition, Columbus, USA, 24–27 June 2014; pp. 580–587.
111. Gao, X.; Zhang, T. Unsupervised learning to detect loops using deep neural networks for visual SLAM system. *Auton. Robot.* **2017**, *41*, 1–18.
112. Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Trans. Pattern Anal.* **2017**, 40, 1437–1451.
113. Kim, A.; Eustice, R.M. Active visual SLAM for robotic area coverage: Theory and experiment. *Int. J. Robot. Res.* **2014**, *34*, 457–475.
114. Thrun, S. *Exploration in Active Learning*; MIT Press: Cambridge, MA, USA, 1995; pp. 381–384.
115. Engel, J.; Stückler, J.; Cremers, D. Large-scale direct SLAM with stereo cameras. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany, 28 September–2 October 2015; pp. 1935–1942.
116. Tateno, K.; Tombari, F.; Laina, I.; Navab, N. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, UAS, 21–26 July 2017; pp. 6565–6574.
117. Rambach, J.R.; Tewari, A.; Pagani, A.; Stricker, D. Learning to Fuse: A Deep Learning Approach to Visual-Inertial Camera Pose Estimation. In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality, Merida, Mexico, 23–26 September 2016; pp. 71–76.
118. Shamwell, E.J.; Leung, S.; Nothwang, W.D. Vision-Aided Absolute Trajectory Estimation Using an Unsupervised Deep Network with Online Error Correction. *arXiv* **2018**, arXiv:1803.05850.
119. Gregorio, D.D.; Stefano, L.D. SkiMap: An efficient mapping framework for robot navigation. In Proceedings of the IEEE International Conference on Robotics and Automation, Marina Bay, Singapore, 29 May–3 June 2017; pp. 2569–2576.

120. Jeong, J.; Cho, Y.; Kim, A. Road-SLAM: Road marking based SLAM with lane-level accuracy. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), California, USA, 11–14 June 2017; pp. 1473–1736.
121. Huang, J.; Dai, A.; Guibas, L.; Niessner, M. 3Dlite: Towards commodity 3D scanning for content creation. *ACM Trans. Graph.* **2017**, *36*, 1–14.
122. Abouzahir, M.; Elouardi, A.; Latif, R.; Bouaziz, S.; Tajer, A. Embedding SLAM algorithms: Has it come of age? *Robot. Auton. Syst.* **2018**, 100, 14–26.
123. Yousef, K.A.M.; Mohd, B.J.; Al-Widyan, K.; Hayajneh, T. Extrinsic Calibration of Camera and 2D Laser Sensors without Overlap. *Sensors* **2017**, *17*, 2346.
124. Zhang, J.; Singh, S. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, USA, 26–30 May 2015; pp. 2174–2181.
125. Rodríguez-Arévalo, M.L.; Neira, J.; Castellanos, J.A. On the Importance of Uncertainty Representation in Active SLAM. *IEEE Trans. Robot.* **2018**, 34, 829–834.
126. Parulkar, A.; Shukla, P.; Krishna, K.M. Fast randomized planner for SLAM automation. In Proceedings of the IEEE International Conference on Automation Science and Engineering, Fort Worth, TX, UAS, 21–24 August 2012; pp. 765–770.
127. Carlone, L.; Du, J.; Ng, M.K.; Bona, B.; Indri, M. Active SLAM and Exploration with Particle Filters Using Kullback-Leibler Divergence. *J. Intell. Robot. Syst.* **2014**, *75*, 291–311.
128. Lai, K.; Fox, D. Object Recognition in 3D Point Clouds Using Web Data and Domain Adaptation. *Int. J. Robot. Res.* ***2010***, *29*, 1019–1037.
129. Indelman, V.; Carlone, L.; Dellaert, F. Planning in the Continuous Domain: A Generalized Belief Space Approach for Autonomous Navigation in Unknown Environments. *Int. J. Robot. Res.* **2015**, *34*, 1021–1029.
130. Berg, J.V.D.; Patil, S.; Alterovitz, R. Motion planning under uncertainty using iterative local optimization in belief space. *Int. J. Robot. Res.* **2012**, *31*, 1263–1278.
131. Saarinen, J.P.; Andreasson, H.; Stoyanov, T.; Lilienthal, A.J. 3D Normal Distributions Transform Occupancy Maps: An Efficient Representation for Mapping in Dynamic Environments. *Int. J. Robot. Res.* **2013**, *32*, 1627–1644.
132. Zou, D.; Tan, P. CoSLAM: Collaborative Visual SLAM in Dynamic Environments. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 354–366.