2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Hierarchical Quadtree Feature Optical Flow Tracking Based Sparse Pose-Graph Visual-Inertial SLAM

Hongle Xie, Weidong Chen*, Jingchuan Wang and Hesheng Wang

Abstract—**Accurate, robust and real-time localization under constrained-resources is a critical problem to be solved. In this paper, we present a new sparse pose-graph visual-inertial SLAM (SPVIS). Unlike the existing methods that are costly to deal with a large number of redundant features and 3D map points, which are inefficient for improving positioning accuracy, we focus on the concise visual cues for high-precision pose estimating. We propose a novel hierarchical quadtree based optical flow tracking algorithm, it achieves high accuracy and robustness within very few concise features, which is only about one fifth features of the state-of-the-art visual-inertial SLAM algorithms. Benefiting from the efficient optical flow tracking, our sparse pose-graph optimization time cost achieves bounded complexity. By selecting and optimizing the informative features in sliding window and local VIO, the computational complexity is bounded, it achieves low time cost in long-term operation. We compare with the state-of-the-art VIO/VI-SLAM systems on the challenging public datasets by the embedded platform without GPUs, the results effectively verify that the proposed method has better real-time performance and localization accuracy.**

I. Introduction

High-accuracy and efficient simultaneous localization and mapping (SLAM) algorithms under limited resources have received increasing attention [1]-[6]. Since the localization and navigation demands of robots are not constant time, the SLAM systems often need to run for a long time continuously within the resource-constrained on-board computing power [1]-[4],[42], during the long-term operation, there are lots of problems to be solved, such as the poor robustness of the front-end visual tracking [4], [43], unbounded computational complexity of the back-end optimization [2], [3], [37] and the increasing resources costs [7], [20], [26].

Assisted by IMU, visual-inertial odometry (VIO) and visual-inertial (VI)-SLAM achieves higher accuracy and better robustness compared with pure vision algorithm [8]-[13], which can adapt to various unstructured environments and run stably in different challenging conditions. However, they have higher computational cost. To ensure the efficiency, lots of the state-of-the-art VIO/VI-SLAM [8]-[13] utilize the lightweight front-end methods, such as FAST Corner detector [14], ORB feature [15], [40], Harris or Shi-Tomasi Corner with KLT optical flow tracker [16], [17]. The accuracy and robustness of those methods are relatively weak, which cause low-quality outliers and reduce the tracking accuracy. Due to the accumulated error of front-end feature tracking and data association, the whole VIO/VI-SLAM system may occur tracking lost or numerical optimization divergence easily during long-term operation.

We propose a novel hierarchical quadtree based optical flow tracking algorithm, which is a fast, robust and accurate front-end tracking method for general VIO/VI-SLAM. The quadtree searching is one of best the two-dimensional search methods [21], [22], we improve it by hierarchical pyramid with histogram and Wiener filtering image enhancement [34], which can deal with the rapid motion, weak textures and uncalibrated illumination changes, those severe challenging conditions may reduce the performance of VIO/VI-SLAM severely. The proposed method selects the most informative robust features for optical flow tracking by hierarchical quadtree strategy, and it distributes the features evenly. Our algorithm achieves robust track within only very few concise features in optical flow, this is suitable for the limited resources platform and has low time cost and latency.

The most related works to ours are the attention in visual-inertial navigation (VIN) [4] and the good feature selection [23] method. In [4], the attention mechanism in VIN is presented, according to the proposed two algorithms: minEig and logDet in greedy and lazy modes, which can choose the most informative features as the suitable visual cues for VIN. In [23], a new solution is proposed to solve the Max-logDet metrics in feature selection. All of those methods are tightly coupled and limited within the particular least squares optimization, and it has high computational complexity and time-consuming. Our algorithm is lightweight and decoupled, which can be implement to a general SLAM framework.

In long-term operation, although loop closing reduces the drift, the computational complexity of the optimization still tends to grow rapidly without boundary [2], [6], which is one of the biggest downsides of SLAM [20], the accumulated features and 3D map points can cause extremely high computational complexity [7], [20], and it has little benefits for positioning. Instead, the proposed visual-inertial SLAM focus on maintaining the sparse pose-graph in sliding window and local VIO, and achieve loop closing with full SLAM keyframe database, the whole trajectories are refined though the global optimization. The main contributions are:

1) To the best of our knowledge, this is the first work proposed the hierarchical quadtree based optical flow tracking algorithm, which is a fast, robust and accurate front-end method for general VIO/VI-SLAM system.

2) We design an efficient, robust and lightweight visual-inertial SLAM (SPVIS), based on a state-of-the-art baseline, and have improved its pose-graph optimization sparsity with fewer features and map points in long-term operation.

This work was supported by the National Natural Science Foundation of China under Grant U1813206 and 61573243.

The authors are with the Institute of Medical Robotics and Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China and Key Laboratory of System Control and Information Processing, Ministry of Education of China (e-mail: {xiehongle, *corresponding author: wdchen, jchwang, wanghesheng} @ sjtu.edu.cn).

3) We present the new finding and applicability of robust tracking within only very few concise features in visual-inertial estimator under the limited resources conditions.

4) We validate our method by comparing with others on the challenging datasets, results show that our method has better accuracy and lower time cost on an embedded platform.

II. System Overview

The framework of the proposed visual-inertial SLAM is shown in Fig. 1. We have implemented our system based on a state-of-the-art baseline VINS-Mono [9] and inspired by OKVIS [8]. First of all, during the sensor's measurements pre-processing module, we synchronize the monocular images and IMU measurements the by optimizing or calibrating the time interval if the hardware synchronization is inefficient.

In the tracking module, the relative motion between consecutive frames is estimated with images and IMU's measurements. After image enhancement, we estimate the initial motion between consecutive frames efficiently by our proposed hierarchical quadtree optical flow tracking algorithm, and the IMU's state is pre-integrated at the synchronization frequency of images. In the visual-inertial alignment module, we combine the measures of the sensors to get an accurate pose estimation, in the meantime, the coarse map points are generated. And the new keyframe is selected according to the mean hierarchical quadtree optical flow movement.

The back-end optimization module of our algorithm is improved based on a fixed-lag VIO sliding window [6], [9]. A highly sparse back-end optimization is achieved with concise map points. the proposed system can run stably within only about the one fifth of feature number of the general VIO systems. To limit the size of optimization, the oldest keyframe is marginalized, and the pose graph is further sparsified by maintaining the high-information measurements. We add these measurements to a local VIO optimization. For the demand of long-time operation, we maintain a full SLAM pose-graph database, the relevant frames can be refined by the loop closing. Once the correct loop is detected, the whole relevant trajectories will be refined in global optimization, and the sparse pose-graph is maintained for long-term operation.

III. Hierarchical Quadtree Optical Flow Tracking

In this section, we present a novel hierarchical quadtree based optical flow tracking algorithm, which is a fast, robust and accurate front-end method for general VIO/VI-SLAM.

A. Optical Flow Tracking

The goal of optical flow is to estimate the relative motion between two image frames by tracking sparse or dense features [16], [17]. The dense optical flow tracks most of the pixels in the image, which is time-consuming. To speed-up the tracking, the sparse feature based optical flow is used. Assuming there is a pair of grayscaled images M and N, whose gray value functions are $M(x,y)$ and $N(x,y)$ separately. Given the feature points set F of image M, the coordinates of F_i in M are $\alpha_i=[x_M^i \ y_M^i]^T$. The aim is to find correct F_i' in N that the gray value $M(x_M^i,y_M^i)$ and $N(x_N^i,y_N^i)$ are similar [17].

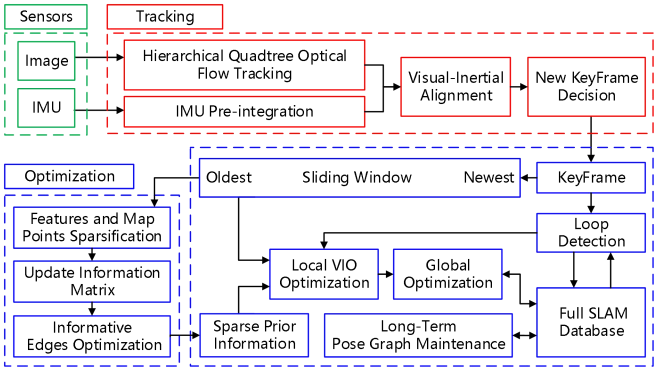


Fig. 1. System framework of the proposed visual-inertial SLAM (SPVIS).

Let $\beta_i=[x_N^i \ y_N^i]^T$ be the coordinates of F_i' in image N, and $d^i=[\Delta x^i \ \Delta y^i]^T$ be the relative motion vector of optical flow, so the relationship between α_i and β_i vectors can be written as:

$$\beta_i=\alpha_i+d^i=[x_M^i+\Delta x^i \ y_M^i+\Delta y^i]^T \tag{1}$$

Assuming that the images M and N are consecutive frames of optical flow tracking, whose corresponding time are t and $t+\Delta t$ respectively. According to the brightness constancy invariant principle [17], they satisfy the following formula:

$$I(x_M^i+\Delta x^i, y_M^i+\Delta y^i, t+\Delta t)=I\left(x_M^i, y_M^i, t\right) \tag{2}$$

we reserve the first-order approximation of Taylor expansion:

$$I(x_M^i+\Delta x^i, y_M^i+\Delta y^i, t+\Delta t)=I\left(x_M^i, y_M^i, t\right)+ \frac{\partial I}{\partial x_M^i}\Delta x^i+\frac{\partial I}{\partial y_M^i}\Delta y^i+\frac{\partial I}{\partial t}\Delta t+\varepsilon \tag{3}$$

where ε is the infinitesimal of higher order. According to the brightness invariance, the following formula can be obtained:

$$\frac{\partial I}{\partial x_M^i}\frac{\Delta x_M^i}{\Delta t}+\frac{\partial I}{\partial y_M^i}\frac{\Delta y_M^i}{\Delta t}=-\frac{\partial I^i}{\partial t} \tag{4}$$

where $\nabla x_M^i=\partial I/\partial x_M^i$ and $\nabla y_M^i=\partial I/\partial y_M^i$ are respectively the gradients in the x_M and y_M directions. The velocities of the relative motion in the image plane are denoted as follows: $u^i=\Delta x_M^i/\Delta t$, $v^i=\Delta y_M^i/\Delta t$. Furthermore, optical flow tracking can be written in matrix form as following:

$$[\nabla x_M^i \quad \nabla y_M^i]\begin{bmatrix}u^i\\ v^i\end{bmatrix}=-\frac{\partial I^i}{\partial t} \tag{5}$$

For each feature point, the $(2w+1)\times(2w+1)$ block of pixels are tracked around the feature point [25]. We find the correct relation of pixels by minimizing the average optical flow tracking error. The relative translation and relation between adjacent frames can be calculated by solving the optical flow equations (6) of all features in the original image.

$$\begin{bmatrix}\nabla x_M^1 & \nabla y_M^1\\ \vdots & \vdots\\ \nabla x_M^k & \nabla y_M^k\end{bmatrix}\begin{bmatrix}u\\ v\end{bmatrix}=-\begin{bmatrix}\partial I^1/\partial t\\ \vdots\\ \partial I^k/\partial t\end{bmatrix} \tag{6}$$

where $[u,v]^T$ are the maximum likelihood solution of relative motion vectors in image M, and k is the number of features.

However, for the strong assumption of brightness constancy, sparse optical flow tracking is often mismatched, and outliers may cause serious data association errors. Furthermore, during fast motion, lots of feature points are lost easily, and they can't be tracked by the next frames. Those weaknesses destroy the

tracking stability and decrease the accuracy of the estimator rapidly. To solve those problems, we propose a novel robust hierarchical quadtree based optical flow tracking method.

B. Hierarchical Quadtree Segmentation

Quadtree is one of the best suitable algorithms for locating pixels in two-dimensional images [22], [24]. The core idea of quadtree index is to segment image space into quadtree structures of different levels recursively, and according to quadtree based two-dimensional search method, this is done recursively until the tree level reaches a certain depth or meets certain requirements. We recursively segment the space of known image range into four equal subspace parts, and extract image features from each quadtree image domain at hierarchical layers. The hierarchical layers are generated by image pyramid at various levels. For each small quadtree image domain, we only retain one best feature point by non-maximum suppression strategy, the features are selected by its strength response, such as Harris response, we summarize all the features of each quad leaf node in tracking thread. The above steps will be repeated recursively until we find the desired correct feature points in hierarchical image layers.

C. Hierarchical Quadtree Optical Flow Tracking

Robust and accurate front-end feature tracking method with efficient data association is very important in VIO/VI-SLAM [8]-[13]. However, towards the long-term operation, there are a few methods that can be deal with most varieties of challenging scenarios, such as illumination change, motion blurring, dark and textureless scenes, and fit the requirements of real-time performance simultaneously.

So, before segmenting the image into hierarchical layers by calculating the image pyramid, we conduct a series of image enhancement processing. For the abruptly illumination and automatic photometric calibration of camera, the brightness of image frames can be changed suddenly [43], which is an intractable problem of optical flow tracking method. we utilize an adaptive histogram equalization, where we try to limited the contrast of the image frame to enhance image with the photometric change smoothly, which is also efficient for feature extraction in dark images. The Wiener filter method is used to solve the motion blurring alternatively [34].

After the image pre-processing, we estimate the motion by hierarchical pyramidal optical flow feature tracking. Based on a general KLT-Tracker in OpenCV [25]. We extract the Shi-Tomasi and FAST corners efficiently, through the quadtree-based two-dimensional search method, the best features for each quad leaf node are selected in different pyramidal levels. Aggregating all of the good feature points, which forms the initial feature points set $\Phi_0 = \{F_i, Pts_i, Desp_i\}_i, i \in [1, N_0]$ to be tracked by optical flow. According to the proposed optical flow tracking method, the initial features set is updated to Φ_0' cyclically. In Φ_0', we have removed the track lost outliers.

According to the proposed hierarchical quadtree optical flow tracking method, we execute the quadtree-based image spatial segmentation, feature extraction and hierarchical optical flow tracking iteratively, until the number of feature points tracked by quadtree optical flow meets the expected requirements. The estimated optical flow is further refined by fundamental matrix RANSAC [35] recursively. The detailed algorithm of our hierarchical quadtree based optical flow tracking method is described in Alg. 1.

Algorithm 1 Hierarchical Quadtree Optical Flow Tracking

Input: Current frame image $I(x, y), x \in [1, X_0], y \in [1, Y_0]$; Optical flow tracked feature points set in the last frame $\Phi_0 = \{F_i, Pts_i, Desp_i\}_i, i \in [1, N_0]$; Expected feature points number N_{Aim}.

Output: Current frame stable optical flow feature points set $\Phi_1 = \{F_i, Pts_i, Desp_i\}_i, i \in [1, N_{Aim}]$ for tracking.

Calculate $I(x, y)$ pyramid of K layers:$\{I_k, k \in [1, K]\}$;
KLT Track and Update Φ_0';
Number of new features should be added $N_w = N_{Aim} - N_0$, denote $N_w = \sum_{k=1}^{K} N_k, k \in [1, K]$;
for each $i \in [1, K]$ **do**
    initialize the number of feature points in i-th layer:
    $N_w^i = \frac{N_w(1-f)}{1-f^K} f^{i-1}$;
    where f is the reciprocal of scale factor;
end for
for all I_k such that $k \in [1, K]$ **do**
    generate root nodes X_{root}^k for each pyramid layer I_k;
    $N_{root}^k = \lfloor Y_0 / X_0 \rfloor$, $N_F^k = N_{root}^k$;
    if $N_{root}^k \geq N_w^k$ **then**
        extract features, by maximizing harris response;
        each node retains only one best feature;
        return features set $\Phi_0' \cup \Phi_{root}^k$; **break**;
    end if
    for each $i \in [1, N_{root}^k]$ **do**
        divide $X_{root}^k(i)$ into sub-node $\{R_U^i, R_B^i, L_U^i, L_B^i\}$;
        initialize new sub-node feature, $N_F^k = N_F^k + 3$;
    end for
    $\Phi_{root}^k = \Phi_{root}^k \cup [\bigcup_{i=1}^{N_{root}^k} \{R_U^i, R_B^i, L_U^i, L_B^i\}] \setminus X_{root}^k$;
    while $N_F^k \leq N_w^k$ **do**
        select one of the quad sub-nodes $\Phi_{root}^k(i^*)$;
        with the most feature points at hierarchical level;
        divide $\Phi_{root}^k(i^*)$ into sub-node $\{R_U^{i^*}, R_B^{i^*}, L_U^{i^*}, L_B^{i^*}\}$;
        initialize new sub-node feature, $N_F^k = N_F^k + 3$;
        $\Phi_{root}^k = \Phi_{root}^k \cup \{R_U^{i^*}, R_B^{i^*}, L_U^{i^*}, L_B^{i^*}\} \setminus \Phi_{root}^k(i^*)$;
    end while
end for
return optical flow features set $\Phi_1 = \Phi_0' \bigcup_{k=1}^{k=K} \Phi_{root}^k$.

IV. Sparse Pose-Graph Visual-Inertial SLAM

A. Sparse Pose-Graph Optimization

Graph-based visual-inertial SLAM represents the back-end as a series of the factor optimizations [2], [26]-[29]. The node factors represent variables, such as the state of camera pose and map points, the edge factors represent the geometric measurements between node factors. For each edge factor, it denotes a measurement z_{mn} between the node x_m and x_n, let

$h(x)$ be the measurement model of sensor. The relative error residual between measurement and expectation is defined as:

$$f_{mn} = f(x_m, x_n, z_{mn}) = h(x_m, x_n) - z_{mn} + v \tag{7}$$

where the sensor's noise is $v \sim N(0, \Sigma_{mn})$. For each sensor measurement, such as the camera pose, IMU state and map point, we define an error function by (7). Summarizing all of the factors Ψ , the whole objective optimization function X can be written as:

$$X^{\text{MAP}} = \min \sum_{(m,n)\in\Psi} \left\| h(x_m, x_n) - z_{mn} \right\|^2_{\Lambda^{-1}_{mn}} \tag{8}$$

where Σ_{mn} is covariance matrix, and the information matrix is $\Lambda_{mn} = \Sigma^{-1}_{mn}$. We solve this problem and find its maximum a posteriori (MAP) solution by minimizing the Mahalanobis distance. The optimization equation can be converted into:

$$\Delta x^* = \arg\min_{\Delta x} \sum_{(m,n)\in\Psi} \left\| \{ h(x_m, x_n) - z_{mn} \} + J_{mn} \Delta x \right\|^2_{\Sigma_{mn}} \tag{9}$$

where J_{mn} refers to the Jacobian matrix, and Δx^* is the target optimal step for updating the optimization. Several methods can be applied to solve the problems, such as QR [27], Cholesky [29] and incremental methods [32]. However, the computational complexity of the accumulated full pose-graph and 3D map points optimization is too large to meet the real-time requirements in the long-term running, we speed-up it by the fixed-lag sliding window and local VIO optimization.

B. Sparse Graph-based Visual-inertial SLAM

The proposed visual-inertial SLAM back-end optimization is implemented based on a state-of-the-art VI-SLAM baseline [9]. Similar to [10], [11], we have implemented an efficient loop closing algorithm based on the DBoW2 [30] with BRIEF descriptor [33]. Inspired by [6], all the states of camera, IMU and map points are added into a joint coupled optimization. We improve its efficiency by reducing the redundancy of features and map points. To limit computational complexity, the sliding window optimization in VIO is adopted. The state variables included in the sliding window states are *n* sensor's states, including its translation p^w_k , rotation R^w_k , velocity v^w_k , the bias of accelerometer b_a and gyroscope b_g , and the Huber loss function refined inverse depth of *m* 3D map points ρ^*_k [31]. The motion of camera and IMU are associated through the external parameters from camera to IMU. The core state variables are as follows:

$$\mathcal{X} = \left[T_0, T_1, \cdots, T_n, \rho^*_0, \rho^*_1, \cdots, \rho^*_m \right] \tag{10}$$

$$T_k = [C^w_k, U_k] = [p^w_k, R^w_k, v^w_k, b_a, b_g] \tag{11}$$

where T_k represents the state of the *k*-th frame, it includes the states of camera $C^w_k = [p^w_k, R^w_k]$ and IMU's measurements U_k , *w* means in world coordinate. Full optimization function of the visual-inertial SLAM is defined as follows:

$$\min_{\mathcal{X}} \left\{ \sum_{k\in U} \left\| R^U_k \right\|^2_{\Sigma_k} + \sum_{(m,n)\in C} \left\| R^C_{mn} \right\|^2_{\Sigma_{mn}} + \left\| R^P \right\|^2_{\Sigma_0} + \sum_{v\in L} \sigma \left\| R^L_v \right\|^2_{\Sigma_v} \right\} \tag{12}$$

$$\min \sum_{k\in U} \left\| R^U_k \right\|^2_{\Sigma_k} \approx \min \sum_{k\in U} \left\| r_U(z^{u_k}_{u_{k+1}}, \mathcal{X}_0) + J^{u_k}_{u_{k+1}} \delta\mathcal{X} \right\|^2_{\mathrm{P}^{u_k}_{u_{k+1}}} \tag{13}$$

$$\min \sum_{(m,n)\in C} \left\| R^C_{mn} \right\|^2_{\Sigma_{mn}} \approx \min \sum_{(m,n)\in C} \left\| r_C(z^{c_n}_m, \mathcal{X}_0) + J^{c_n}_m \delta\mathcal{X} \right\|^2_{\mathrm{P}^{c_n}_m} \tag{14}$$

$$\min \left\| R^P \right\|^2_{\Sigma_0} \approx \min \left\| r_0 - J_0 \mathcal{X} \right\|^2_{\mathrm{P}_0} \tag{15}$$

where R^U_k and R^C_{mn} are the measurement residuals of IMU and camera, J and P are the Jacobian and covariance matrix of the measurement residual [9]. When each keyframe enters the sliding window, we marginalize the earliest keyframe in the sliding window to ensure the bounded size of the optimization, according to the Schur complement [36], R^P means the prior information residuals of marginalization, and R^L_v is the loop closing residuals, σ is the kernel function. The optimization problem is solved by minimizing Mahalanobis distance [29], [31], [32] and finding the optimal $\delta\mathcal{X}$ interactively:

$$\begin{aligned} &\left(J_0^T P_0^{-1} J_0 + \sum J^{u_k\ T}_{u_{k+1}} P^{u_k\ -1}_{u_{k+1}} J^{u_k}_{u_{k+1}} + \sum J^{c_n T}_m P^{c_n -1}_m J^{c_n}_m + H_L \right) \delta\mathcal{X} \\ &= b_P + \sum J^{u_k\ T}_{u_{k+1}} P^{u_k\ -1}_{u_{k+1}} r_U + \sum J^{c_n T}_m P^{c_n -1}_m r_C + b_L \end{aligned} \tag{16}$$

$$H^* \delta\mathcal{X} = \left(H_P + H_U + H_C + H_L \right) \delta\mathcal{X} = b_P + b_U + b_C + b_L = b^* \tag{17}$$

where H^* refers to the information matrix. The size of H^* and its sparsity directly affects the computational efficiency [2], [3], [39], especially the marginalization of camera poses and 3D map points in the sliding window and loop closing, this is one of the largest shortcomings of VIO/VI-SLAM [20], whose computational complexity increases rapidly with the number of features and map points. Instead of the full global optimization, we solve the incremental equation $H^* \delta\mathcal{X} = b^*$ with subspace method by partially update variables [39], and convert it into subgraph, which is faster and more efficient.

Particularly, we simplify the complexity of map points optimization in fixed-lag sliding window by significantly reducing the number of feature points through the proposed hierarchical quadtree optical flow tracking method. And we decouple the features from global optimization, we focus on optimizing the sparse pose-graph without maps points, where the concise features are used for fast initial estimating and local mapping. Reducing the computation for optimizing the map points improves the real-time performance. We achieve competitive accuracy by only using an extraordinarily small number of feature points, about a fifth of the original number during the optical-flow tracking thread, which achieves lower time cost than the state-of-the-art VIO/VI-SLAM systems.

V. Experimental Results

We have integrated the proposed SPVIS system in a resource-constrained embedded platform, which is the Intel UP Squared development kits, it carries Apollo Lake low-cost on-board N4200 processors. All the experiments are carried out on this embedded platform. The real-time tests on this lightweight board verify that our system is suitable for mini-type drones and general robot platforms. Each process thread in our system is processed in real-time without any GPUs.

A. Localization Accuracy Comparison

We evaluate the performance of the proposed SPVIS on EuRoC benchmark [38], which contains various challenging sequences. We only use the left camera's images and the ADIS16448 IMU's measures. Our results are compared with the state-of-the-art VIO/VI-SLAM, such as the sparse feature based OKVIS [8], VINS-Mono [9], VINS-Fusion [10], ICE-

TABLE I

LOCALIZATION ACCURACY RESULTS COMPARISON ON THE EUROC DATASETS IN METERS

Sequences	Length /m	**Proposed SPVIS**	OKVIS [8]	VINS-Mono [9]	VINS-Fusion [10]			ICE-BA [11]		VI-DSO [12]	R-VIO [13]
					Stereo	Mono+IMU	Stero+IMU	w/ loop	w/o loop		
MH_01_easy	79.84	**0.0609**	0.33	0.12	0.54	0.18	0.24	0.11	0.09	**0.06**	0.19
MH_02_easy	72.75	**0.0435**	0.37	0.12	0.46	0.09	0.18	0.08	0.07	**0.04**	0.31
MH_03_medium	130.58	0.0696	0.25	0.13	0.33	0.17	0.23	**0.05**	0.11	0.12	0.29
MH_04_difficult	91.55	**0.1037**	0.27	0.18	0.78	0.21	0.39	0.13	0.16	0.13	0.76
MH_05_difficult	97.32	0.1199	0.39	0.21	0.50	0.25	0.19	**0.11**	0.27	0.12	0.45
V1_01_easy	58.51	**0.0490**	0.09	0.07	0.55	0.06	0.10	0.07	0.05	0.06	0.08
V1_02_medium	75.72	**0.0446**	0.14	0.08	0.23	0.09	0.10	0.08	0.05	0.07	0.11
V1_03_difficult	78.77	0.0866	0.21	0.19	X	0.18	0.11	**0.06**	0.11	0.10	0.12
V2_01_easy	36.34	0.0657	0.09	0.081	0.23	0.06	0.12	0.06	0.12	**0.04**	0.16
V2_02_medium	83.01	0.0667	0.17	0.16	0.20	0.11	0.10	**0.04**	0.09	0.06	0.16
V2_03_difficult	85.23	**0.0852**	0.23	0.22	X	0.26	0.27	0.11	0.17	0.17	0.27
Avg.	80.978	**0.0734**	0.23	0.14	X	0.15	0.19	0.08	0.12	0.09	0.26

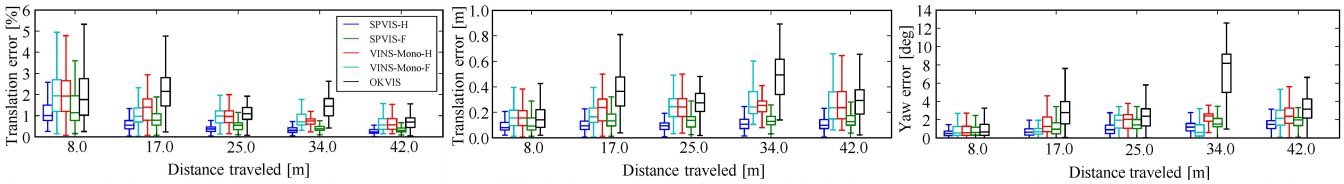


Fig. 2. Translation and yaw errors on the EuRoC V2_03_difficult datasets. From the left to right is the percentage error of translation, absolute translation and yaw error respectively. We also compare with the OKVIS [8], which is one of the best VIO system, we test its performance without loop detection.

BA [11], R-VIO [13] and one of best direct method based VI-DSO [12]. Our algorithm is tested on the UP Squared N4200 processors only with CPUs. We evaluate the localization accuracy by comparing the root mean square error (RMSE) of ATE [18], [19], in which we utilize high-precision mode. The experimental results are shown in Table I, where the total traveling distance of the datasets is about 900 m, the error of our system is only 0.0734 m, the results show that we achieve superior accuracy than the state-of-the-art methods.

For analyzing the translation and yaw angle estimating accuracy, we further conduct the following experiments. Since the proposed system can achieve accurate and robust tracking only with very few features. For fair comparison, we proposed two different modes of the visual-inertial SLAM system operation, one is the fast-tracking (-F) operation mode and the other is high-precision (-H) operation mode.

In the fast-tracking mode, we reduce the optimization time and the number of iterations. Using the Ceres solver [31], we set the maximum number of iterations to a low level, such as 8 times for all methods in the fast-tracking mode, the maximum number of feature points extracted by VINS-Mono-F is set to 100 to ensure that it can run stably. If the number of feature points is further reduced, the accuracy of the VINS-Mono will decrease rapidly due to the reduction of the number of feature points effectively tracked, in MH_04_difficult and V2_03_difficult scenarios, there will be large tracking errors, or even tracking loss. Particularly, the proposed SPVIS achieves high precision and stable tracking even within 25 features, so the maximum feature point number of SPVIS-F is set to 25 and 2 hierarchical quadtree layers in fast-tracking mode.

For the high-precision mode, we increase the optimization time and number of iterations for each operation of the system processing threads, and the maximum number of iterations is set to 30 in the high precision mode, VINS-Mono-H sets the

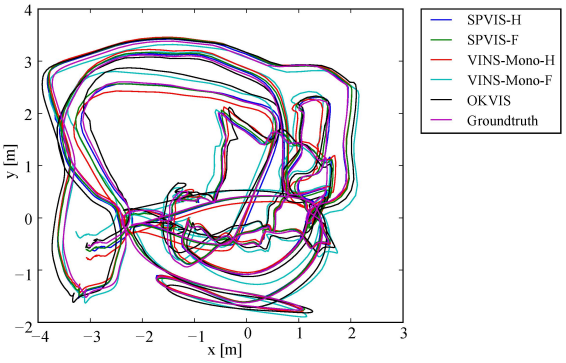
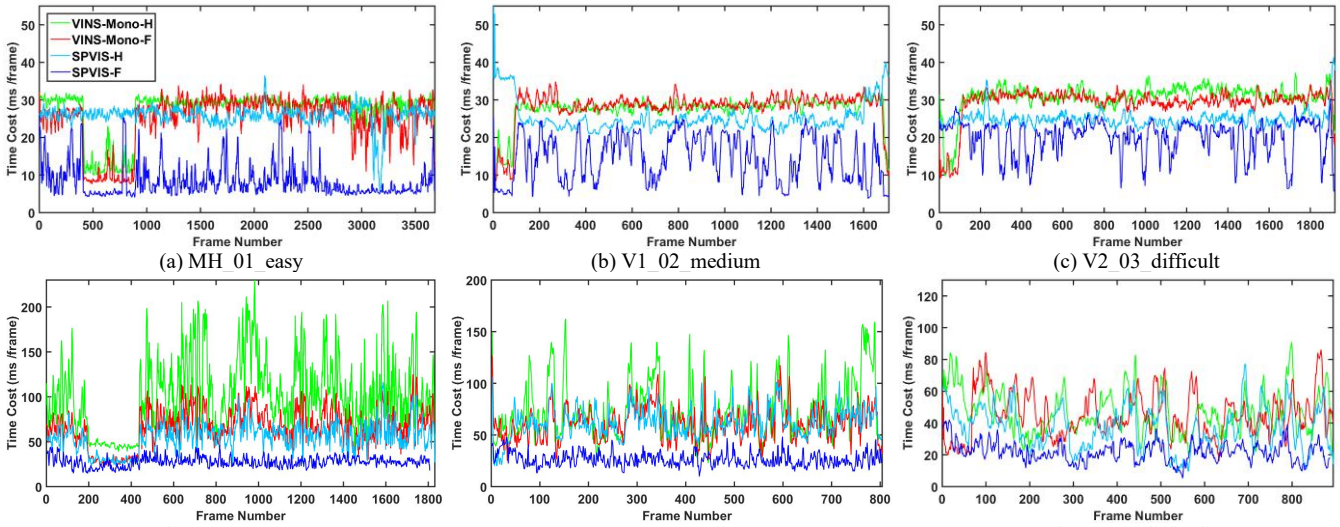


Fig. 3. Trajectory estimation results on the EuRoC V2_03_difficult dataset.

number of features to 150 for improving the accuracy. As the SPVIS system uses the hierarchical quadtree optical flow tracking method, which ensures the robust tracking, so it is redundant to extract more features for higher accuracy in SPVIS, we set the maximum feature number of SPVIS-H to 100 and 5 layers particularly.

We compare all methods on the V2_03_difficult datasets, one of the most difficult sequence of EuRoC datasets. All of the experiments are conducted on the UP Squared platform. Experimental results are shown in Fig. 2. Our SPVIS-H and SPVIS-F achieve the best accuracy, the average localization errors are less than 8.5 cm and 10cm separately, and the third best is VINS-Mono-H, whose error is about 15cm as shown in Fig. 2. In VINS-Mono, the translation errors become smaller when more feature points are used, but due to the poor quality and unstable features are added to the tracking thread, which may cause the low-quality data association and inaccurate estimation of map points, the yaw angle errors are still not reduced significantly, as shown in Fig. 2. Furthermore, the whole estimated trajectories are shown in Fig. 3, the proposed SPVIS achieves the best localization results.



(a) MH_01_easy (b) V1_02_medium (c) V2_03_difficult

(d) MH_01_easy (e) V1_02_medium (f) V2_03_difficult

Fig. 4. Comparison of processing time cost. In the high-precision operation mode, VINS-Mono-H and SPVIS-H are depicted in green and sky blue separately. In the fast-tracking operation mode, VINS-Mono-F and SPVIS-F are depicted in red and blue separately. The figures (a), (b) and (c) are the average tracking time consumption of each frame. The figures (d), (e) and (f) are the average back-end optimization time consumption of each keyframe.

These results are benefited from the high-quality tracking of the proposed hierarchical quadtree optical flow, which can avoid the additional errors caused by low-quality feature optical flow tracking. Tracking with a smaller number of features can improve the sparsity of system, which is efficient for optimization and significant in long-term operation.

B. Real-Time Performance Analysis

For evaluating the real-time performance of the proposed algorithm, we randomly select several datasets in different scenarios, such as MH_01_easy, V1_02_medium and V2_03 _difficult scenes. For each dataset, we compare the CPU time cost of tracking and optimization under different operation modes. As shown in the Fig. 4, the proposed SPVIS-F algorithm achieves the fastest optical flow tracking whose optimization time cost achieves bounded complexity, and it can track feature points stably, especially in the low speed motion in MH01 sequence, our algorithm does not need to search and generate new feature points frequently, which is benefiting from the proposed hierarchical quadtree optical flow tracking. Particularly, as shown in Table II, the average CPU time cost of hierarchical quadtree optical flow tracking is only about 8.4 ms/frame, this means our proposed method achieves over 115 Hz within the limited resource embedded platform, which achieves the state-of-the-art real- time performance. The optimization cost is shown in Table III, the proposed SPVIS has better real-time performance.

The proposed algorithm is robust and also suitable for high-speed motion conditions. As shown in Fig. 4c, in the V2_03_ difficult datasets, the drone moves very fast with rapid translation and pure rotation, the optical flow tracking algorithm must constantly increase the optical flow features, resulting in a large increase in the time cost of the algorithm. However, in this challenging situation, the proposed SPVIS can still maintain stable tracking and accurate localization, furthermore, as shown in Table II and Table III. Even in the most difficult conditions, the real-time performance of the proposed SPVIS is still better than other algorithms.

TABLE II

AVERAGE TIME COST OF TRACKING

Time (ms)	MH_01_easy	V1_02_medium	V2_03_difficult
VINS-Mono-H	27.0996	27.5480	30.7026
VINS-Mono-F	24.6112	28.1416	29.2631
SPVIS-H	25.8372	25.4017	24.9281
SPVIS-F	**8.4017**	**14.5611**	**20.0088**

TABLE III

AVERAGE TIME COST OF EACH KEYFRAME OPTIMIZATION

Time (ms)	MH_01_easy	V1_02_medium	V2_03_difficult
VINS-Mono-H	101.9261	73.9116	46.7598
VINS-Mono-F	65.2593	61.3139	44.8913
SPVIS-H	54.8646	61.9649	37.2081
SPVIS-F	**27.5037**	**26.3928**	**21.7114**

VI. CONCLUSION AND FUTURE WORK

In this paper, we have developed a fast, accurate and robust visual-inertial SLAM. In particular, we have proposed a novel hierarchical quadtree optical flow tracking method. Through effective hierarchical quadtree search mechanism, the front-end optical flow can achieve robustly track the informative features stably. Furthermore, we simplify the sparsity of the system by tracking with very few features. By reducing the cost of processing redundant map points, we improve the sparsity of pose-graph optimization in sliding window and local VIO, and drift is reduced by loop closing. The proposed system achieves the competitive accuracy and better real-time performance under limited resources than the state-of-the-art VIO/VI-SLAM. For future works: First, we will focus on the more efficient visual cue selection approach for a more robust optical flow tracker in dynamic scenes. Second, we will refine the visual-inertial estimator optimization and resource reuse efficiency for continuous running and mapping, and further evaluate our algorithms on resources limited drones.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309-1332, Dec. 2016.

[2] J. Vallvé, J. Solà and J. Andrade-Cetto, "Pose-graph SLAM sparsification using factor descent," *Robot. Auton. Syst*, vol. 119, pp. 108-118. 2019.

[3] D. N. Ta, N. Banerjee, S. Eick, S. Lenser and M. E. Munich, "Fast nonlinear approximation of pose graph node marginalization," in *Proc. IEEE Int. Conf. Robot. Autom*. Brisbane, QLD, 2018, pp. 2494-2501.

[4] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 1-20, Feb. 2019.

[5] Y. Yang, P. Geneva, X. Zuo, K. Eckenhoff, Y. Liu and G. Huang, "Tightly-coupled aided inertial navigation with point and plane features," in *Proc. IEEE Int. Conf. Robot. Autom*, Montreal, QC, Canada, 2019, pp. 6094-6100.

[6] J. Hsiung, M. Hsiao, E. Westman, R. Valencia, and M. Kaess, "Information sparsification in visual-inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*., Madrid, 2018, pp. 1146-1153.

[7] P. Muhlfellner, M. B ¨ urki, M. Bosse, W. Derendarz, R. Philippsen, and P. Furgale, "Summary maps for lifelong visual localization," *J. Field Robot.*, vol. 33, pp. 561–590, 2015. [Online]. Available: http://dx.doi.org/10.1002/rob.21595.

[8] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual inertial odometry using nonlinear optimization," *Int. J. Robot Res.*, vol. 34, no. 3, pp. 314–334, Mar. 2015.

[9] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[10] T. Qin, J. Pan, S. Cao, and S. Shen. "A general optimization-based framework for local odometry estimation with multiple sensors," arXiv preprint arXiv:1901.03638, 2019.

[11] H. Liu, M. Chen, G. Zhang, H. Bao, and Y. Bao, "ICE-BA: incremental, consistent and efficient bundle adjustment for visual-inertial slam," in *Proc. of the IEEE Int. Conf. on Pattern Recog*, 2018, pp. 1974–1982.

[12] L. V. Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *Proc. IEEE Int. Conf. Robot. Autom*., Brisbane, QLD, Australia, 2018, pp. 2510–2517.

[13] Z. Huai and G. Huang, "Robocentric visual-inertial odometry," *Int. J. Robot Res.*, 2019. [Online]. Available: https://journals.sagepub.com/ doi/abs/10.1177/02783649 19853361.

[14] E. Rosten, R. Porter, and T. Drummond, "Faster and better: a machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 32, pp. 105-119, 2010.

[15] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.

[16] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell*. Vancouver, Canada, Aug. 1981, pp. 24–28.

[17] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Int. Conf. Pattern Recog*., 1994, pp. 593–600.

[18] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.

[19] Z. Zhang and D. Scaramuzza. "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7244-7251.

[20] P. Geneva, J. Maley and G. Huang. "An efficient Schmidt-EKF for 3D visual-inertial SLAM," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2019, pp. 12105-12115.

[21] D. Sharma and S. Vatta, "Optimizing the search in hierarchical database using quad tree," *Int. J. Sci. Res. Sci. Eng. Tech*, vol. 1, no. 4, pp. 221-226, 2015.

[22] C. Zhang, Y. Zhang, W. Zhang and X. Lin, "Inverted linear quadtree: efficient top k-spatial keyword search," *IEEE Trans.Knowledge and Data Engineering,* vol. 28, no. 7, pp. 1706-1721, 1 July 2016.

[23] Y. Zhao and P. A. Vela, "Good feature selection for least squares pose optimization in VO/VSLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*, Madrid, 2018, pp. 1183-1189.

[24] Z. F. Muhsin, A. Rehman, A. Altameem, T. Saba and M. Uddin, "Improved quadtree image segmentation approach to region information," *The Imaging Science Journal*, vol. 62, no. 1, pp. 56-62. 2014.

[25] OpenCV Developers Team, "Open source computer vision (OpenCV) library," [Online]. Available: http://opencv.org. 3.

[26] K. Lenac, J. Ćesić, I. Marković and I. Petrovi, "Exactly sparse delayed state filter on Lie groups for long-term pose graph SLAM". *Int. J. Robot Res.*, vol. 37, no.6, pp. 585-610, May. 2018.

[27] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Delbert, "iSAM2: Incremental smoothing and mapping using the bayes tree," *Int. J. Robotics Res*., vol. 31, no. 2, pp. 216–235, 2011

[28] K. Khosoussi, M. Giamou, G. S. Sukhatme, S. Huang, S. Dissanayake and J. P How, "Reliable graphs for SLAM," *Int. J. Robot Res.*, vol. 38 no.3, pp. 260-298. 2019.

[29] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom*., Shanghai, China, May 2011, pp. 3607–3613.

[30] D. Galvez-Lopez and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot*., vol. 28, no. 5, pp. 1188–1197, Oct. 2012

[31] S. Agarwal et al., "Ceres solver." [Online]. Available: http://ceressolver. org., 2019.

[32] V. Ila, L. Polok, M. Solony, and P. Svoboda, "SLAM++–A highly efficient and temporally scalable incremental SLAM framework," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 210–230, Feb. 2017.

[33] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha and P. Fua, "BRIEF: computing a local binary descriptor Very Fast," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281-1298, July 2012.

[34] Y. Tai and S. Lin, "Motion-aware noise filtering for deblurring of noisy and blurry images," in *Proc. IEEE Int. Conf. Pattern Recog*., Providence, RI, 2012, pp. 17-24.

[35] Y. Zheng, S. Sugimoto and M. Okutomi, "A branch and contract algorithm for globally optimal fundamental matrix estimation," in *Proc. IEEE Int. Conf. Pattern Recog*., Colorado Springs, CO, USA, 2011, pp. 2953-2960.

[36] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *J. Field Robot.*, vol. 27, no. 5, pp. 587–608, Sep. 2010.

[37] N. Carlevaris-Bianco, M. Kaess and R. M. Eustice, "Generic node removal for factor-graph SLAM," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1371-1385, Dec. 2014.

[38] M. Burri et al., "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, 2016, pp. 1157–1163.

[39] M. Yokozuka, S. Oishi, S. Thompson, and B. Atsuhiko, "VITAMIN-E: visual tracking and mapping with extremely dense feature points," in. *Proc. of the IEEE Int. Conf. on Pattern Recog.,* 2019, pp. 9641-9650.

[40] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255-1262, Oct. 2017.

[41] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler and D. Scaramuzza, "Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset," in *Proc. IEEE Int. Conf. Robot. Autom*., Montreal, QC, Canada, 2019, pp. 6713-6719.

[42] Z. Z. Nejad, A. H. Ahmadabadian, "ARM-VO: an efficient monocular visual odometry for ground vehicles on ARM CPUs," *Machine Vision and Applications*, vol. 30, no. 6, pp. 1-10, Sep. 2019.

[43] X. Wu and C. Pradalier, "Illumination robust monocular direct visual odometry for outdoor environment mapping," *in Proc. IEEE Int. Conf. Robot. Autom*., Montreal, QC, Canada, 2019, pp. 2392-2398.

[44] K. Eckenhoff, P. Geneva, J. Bloecker and G. Huang, "Multi-camera visual-inertial navigation with online intrinsic and extrinsic calibration," in *Proc. IEEE Int. Conf. Robot. Autom*., Montreal, QC, Canada, 2019, pp. 3158-3164.

[45] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robot. Autom. Lett.,* vol. 2, no. 2, pp. 796-803, April 2017.

[46] J. Jackson, K. Brink, B. Forsgren, D. Wheeler and T. McLain, "Direct relative edge optimization, a robust alternative for pose graph optimization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1932-1939, April. 2019.