Autonomous State Estimation and Mapping in Unknown Environments with Onboard Stereo Camera for MAVs

***Abstract*—Industrial micro aerial vehicles (MAVs) with robotic manipulators have numerous applications in search and rescue tasks that reduce risks to human beings. However, such tasks distinctly require MAVs to have the capability of real-time autonomous navigation only with onboard sensors, especially in GPS-denied applications. This study introduces a new approach to onboard vision-based autonomous state estimation and mapping for MAVs' navigation in unknown environments. The algorithms run on board and do not need an external positioning system to assist autonomous navigation. The state estimator is developed to provide MAV's current pose on the basis of the extended Kalman Filter by using image patch features. Inverse depth convergence monitoring and local bundle adjustment are utilized to improve the accuracy. The mapping algorithm for navigation is developed according to a real-time stereo matching method for 3D perception. Finally, we performed several experiments to demonstrate the effectiveness of the proposed approach.**

***Index Terms*—MAV, autonomous navigation, state estimation, 3D occupancy mapping**

I. Introduction

INDUSTRIAL micro aerial vehicles (MAVs) represent an ideal platform for autonomous agriculture, industry, surveillance, and search and rescue because of their appropriate size and high maneuverability and their ability to fly in various environments [1]. Many recent works demonstrated the great potential applications of MAVs equipped with lightweight manipulators [2], [3], particularly in express transportation and high-building maintenance and construction. In these applications, MAVs are required to explore and map the environment with onboard sensors. However, most existing MAV systems depend on the assistance of off-board sensors, such as GPS, which is often unreliable or completely unavailable in GPS-denied environments. Accurate pose estimation and perception of surrounding environments with onboard sensors are critical for safety and practicality. However, these tasks remain challenging for MAVs because the payload, power battery, and computation resources of MAV systems are limited compared with ground unmanned systems.

To implement autonomous navigation, different types of sensors have been used in the literature, e.g., laser range finder [4], artificial marker-aided vision [5], RGB-D sensors [6], and monocular camera [7], [8]. Although lasers have been widely used in mobile vehicles, 3D laser is extremely heavy and power consuming for MAVs. RGB-D sensors are usually unsuitable for outdoor tasks involving intense illumination, and the effective range is inadequately large. Artificial marker-aided vision is inconvenient for tasks in unexplored environments. Monocular camera can only estimate odometry up to scale, and its dense mapping requires substantial computation resources. Compared with other sensors, binocular camera system exhibits many advantages, such as lower cost than 3D lasers and more robustness than monocular cameras and RGB-D in outdoor environments. However, it suffers from extra computation cost when computing disparity images.

Without loss of generality, MAV's autonomous navigation system combines three modules, namely, MAV's state estimator, environmental map building, and path planner. The pose estimator enables the MAV to locate itself in the world frame. The mapping module builds the 2D/3D environmental map and helps the robot localize and avoid obstacles in the environments. The path planner plans a path for exploring environments or moving to a target place without obstacle collision given the built map. Every module is challenging and attracts increasing attention in MAV research. Interestingly, the combination of all three modules is a system-level work and thus a challenging task. Many approaches about the mentioned modules for mobile robots are available [9], [10]. However, achieving autonomous navigation is still a to-be-addressed problem for MAVs because both the accuracy and real-time performance of algorithms running on board need to be satisfied at the same time during practical implementations. Especially, due to the battery power and MAV payload constraints, it is difficult to equip an onboard GPU for 3D environment mapping.

The main contribution of the study lies in the development of a novel and efficient system for the autonomous navigation of MAVs through the use of onboard sensors alone. The sensors include a stereo camera and MAVs' inertial measurement unit (IMU). The contributions are three-fold.

First, inspired by [11], a fully robot-centric state estimator with improved accuracy and robustness is developed for MAV pose estimation. We improve the state estimator by applying optimization method in a maintained sliding window and monitoring the inverse depth convergence. These solutions improve the accuracy of the state estimator with equivalent computational cost compared to ROVIO. Additionally, failure detection and recovery has been developed to improve the robustness.

Second, a novel and fast stereo 3D mapping algorithm based on a stereo camera is proposed to enable MAVs to explore unknown environments autonomously. A MAV with limited computation resources needs to perceive the surroundings in real time during the navigation, especially in the aerial manipulation mission where a MAV frequently interacts with the environments. The existing methods for 3D reconstruction generally require adequate computational resources, and many of them are developed based on GPU. They are hard to meet MAVs' demand. Therefore, we provide several solutions for

MAVs' real-time 3D mapping, including triangulation method for depth calculation, optimized disparities calculation, and 3D occupancy mapping.

Final, we propose a navigation scheme including online modules of state estimator, stereo occupancy mapping, path planning, exploration and offline map optimization. Our work makes it possible to enable MAVs full navigation ability in GPS-denied environments.

The remainder of the study is organized as follows: Section II presents the related work. Section III presents the state estimator for MAVs. Section IV details the implementation of stereo dense mapping. Section V shows and discusses the experimental results. Finally, Section VI provides the conclusion.

II. Related Work

Autonomous navigation approaches in unknown and GPS-denied environments were developed mainly for ground mobile robots in the early stage [12], [13]. The existing approaches often rely on heavy but efficient sensors, such as 3D laser range finders, benefited from the high payload and computational ability of the mobile robots. Ground mobile robots are generally equipped with encoder-based odometry, and thus navigation is easier than that for MAVs. The serious flight inference to the sensors and the complex 6D motility makes MAVs' navigation a challenging task, and thus the approaches for ground mobile robots cannot directly be used for MAVs.

Some approaches to the navigation of MAVs with camera systems already exist. Several researchers attempted to use artificial markers on the ground to help navigation. Eberli *et al.* [14] utilized two concentric circles on the ground as landmarks to estimate MAV pose. Yang *et al.* [15] presented an onboard monocular vision system for MAV's autonomous takeoff, hovering, and landing, and the letter H surrounded by a circle on the ground was used to estimate MAV's state. However, artificial marker-based approaches should ensure that the marker can be seen during flight. Other researchers applied optical flow alone or integrated it with inertial measurements for navigation. Zingg *et al.* [16] used optical flow to construct a dense map for wall collision avoidance. The system successfully maneuvered a helicopter through an indoor corridor. Lippiello *et al.* [17] presented an interesting vision-based obstacle avoidance technique for MAV indoor navigation. The MAV's velocity was controlled through a repulsive force field derived from the depth map computed from the optical flow. However, optical flow can estimate only the relative speed of image features, and the estimated position of MAV always drifts over time. The above vision-based approaches are only efficient for simple applications.

Recently, simultaneous location and mapping (SLAM) has attracted attention in MAVs' autonomous navigation. Engel *et al.* [18] used extended Kalman Filter (EKF) to fuse monocular information with inertial measurements for MAVs' state estimation. The odometry drift is eliminated by the PTAM procedure [19]. However, the visual algorithms were processed off board, and the 3D perception of the environments was not developed. To achieve 3D mapping, Faessler *et al.* [8] presented a vision-based dense mapping approach. However, the dense 3D reconstruction REMODE [20] was run on a remote desktop with GPU-based assistance. Lin *et al.* [21] developed a nonlinear optimization-based state estimator. However, its monocular dense mapping depends on on-board GPU and suffers from illumination change. Bloesch *et al.* [11] presents ROVIO, a tight coupling visual-inertial odometry based on EKF, exhibiting more robustness to textureless scene and motion blur and lower computational cost than optimization-based methods. However, as a visual-inertial odometry framework, the localization drift is unavoidable, and no failure recovery was provided. Although the further framework Maplab [22] can create visual-inertial maps and provide drift-free pose, the dense map reconstructions and drift eliminate procedure are all offline. This condition provides difficulty for MAVs to navigate online in unknown environments autonomously.

Other researchers developed online navigation approaches only with CPU supports in the past years rather than GPU-based approaches. Huang *et al.* [23] presented a SLAM system by using RGB-D camera for autonomous flight, and visual odometry (VO) was used to estimate the position and velocity of MAVs in real time. To correct the drift of the real-time pose estimation, the SLAM system periodically incorporates position correction to the VO. Alternatively, Stumberg *et al.* [24] utilized a monocular camera and presented a method for autonomous flight. The vision-based navigation system builds on LSD-SLAM [25] for estimating MAV trajectory and implementing a semidense reconstruction of the environment in real time. However, the LSD-SLAM can only construct a semidense map.

This study provides an efficient onboard vision-based autonomous navigation solution for autonomous MAVs in both indoor and outdoor GPS-denied unknown environments. Unlike the existing approaches, a binocular stereo camera system is utilized for both robust state estimation and online construction of the dense 3D map of the operating environments. The proposed solution does not need GPU resources to assist the vision process, and the real-time efficiency without complex hardware support specifically meets the previously mentioned computation, payload, power and cost requirements of MAV applications. The proposed lightweight stereo vision based autonomous navigation approach dose not rely on any GPU acceleration or architecture-based compilation optimization. It is possible to apply our solution to cost and power-constrained ground robots, such as inspection robots or rescue robots, for autonomous navigation in unknown environments. In addition, our robust state estimator can also be used in devices such as mobile AR and VR which need the localization ability in GPS-denied environments. In the future, we are trying to apply the proposed approach in applications of aerial manipulators, where the aerial manipulators are autonomously navigated to grasp or transport objects in unknown environments.

Given the limitation of paper space, this study only provides the details of the two critical modules, i.e. the state estimation and 3D mapping. The algorithms in other modules are based on generally used methods and can be found in the literature [9], [10], [26].

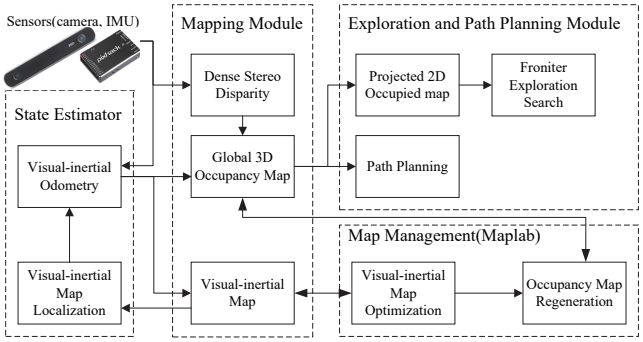


Fig. 1. Software diagram of the proposed system.

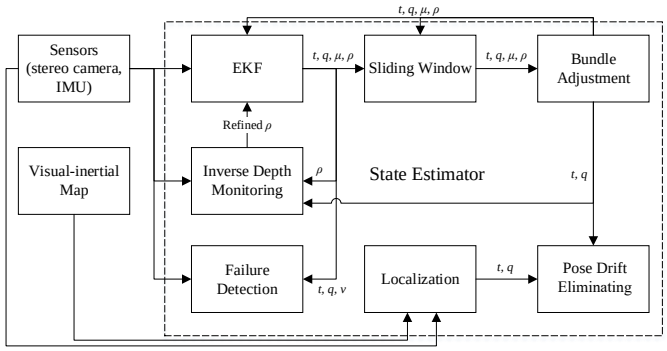


Fig. 2. Software diagram of the proposed state estimator.

III. State Estimation Algorithm

Fig.1 illustrates the software diagram of the proposed autonomous MAV navigation system, wherein the camera and IMU measurements are fused to achieve MAV state estimation and generate an origin VI-map [22]. The VI-map is composed by a pose graph of the exploration trajectory. The vertices are represented by states, such as environmental landmarks (BRISK) [27], MAV's poses and velocities. BRISK has invariance to both scale and rotation and it is also more computational friendly than SURF or SIFT. The edges connecting the two neighboring vertices are represented by IMU constraints. Stereo images are processed by a stereo matching algorithm to provide 3D point clouds. The resulting point clouds are then turned into a global dense occupancy map based on the pose data from the proposed state estimator. The built dense map is sent back to the exploration module to avoid obstacles and plan the next exploration waypoint.

State estimation plays a crucial role in an MAV system. Estimated state information is used not only for flight control but also for environmental mapping and path planning. The accuracy and robustness of the state estimator directly determines control and navigation performance. An algorithm is proposed in this section to estimate the full state of the MAV in the world frame, inspired from ROVIO [11]. Fig.2 illustrates the framework overview of the state estimator. Once a feature is detected by FAST corner detector, an image patch instead of a descriptor is used to describe and track this feature. Different from the IEKF used in ROVIO, EKF and local bundle adjustment are used together for fusing visual and IMU information. In addition, two modules including inverse depth monitoring and failure detection are developed to improve the accuracy and robustness of the state estimator. The VI-map that provides a sparse reconstruction of the explored environment is generated during exploration to eliminate localization drift of odometry.

A. EKF Formulation for State Estimation

We assume MAV body's center coincides with IMU center. The state transition and original observation models can be written as

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{w}_{k-1}), \boldsymbol{z}_k = h(\boldsymbol{x}_k, \boldsymbol{n}_k), \tag{1}$$

where k is the timestamp, and $\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{W})$ and $\boldsymbol{n} \sim \mathcal{N}(0, \boldsymbol{R})$ are noise terms. The state vector x is defined as

$$\boldsymbol{x} = (\boldsymbol{t}_{\mathcal{WB}}^T, {}_{\mathcal{B}}\boldsymbol{v}^T, \boldsymbol{q}_{\mathcal{WB}}^T, \boldsymbol{b}_a^T, \boldsymbol{b}_\omega^T, \boldsymbol{t}_{\mathcal{IC}}^T, \boldsymbol{q}_{\mathcal{IC}}^T, \boldsymbol{\mu}_0^T, ..., \boldsymbol{\mu}_N^T, \rho_0^T, ..., \rho_N^T)^T, \tag{2}$$

where $\boldsymbol{x}_M = (\boldsymbol{t}_{\mathcal{WB}}^T, {}_{\mathcal{B}}\boldsymbol{v}^T, \boldsymbol{q}_{\mathcal{WB}}^T)^T$ denotes MAV's state vector including MAV body center, velocity and attitude in the world frame. $\boldsymbol{b}_a$ and $\boldsymbol{b}_\omega$ are additive bias on the accelerometer and gyroscope, respectively. $\boldsymbol{t}_{\mathcal{IC}}$ together with $\boldsymbol{q}_{\mathcal{IC}}$ denotes the transformation from IMU to camera frame. $\boldsymbol{\mu}_i$ and ρ_i denote the bearing vector and inverse depth of landmark i in the camera frame, and N is the number of currently tracked landmarks. Note the bearing vector of each landmark represents the direction of the 3D landmark in the camera frame, and it is a 2D vector in a spanned tangent space of a 3D unit vector expressed in the camera frame.

By considering the self-consistency of the work, we briefly explain the EKF-based sensor fusion formulation that is also used in ROVIO. Image patches [27] are used as landmark descriptors, and the intensity error and re-projection error are integrated as innovation term $\boldsymbol{y}_k$ during EKF update. Given a posteriori estimate $\boldsymbol{x}_{k-1}$ and its covariance matrix $\boldsymbol{P}_{k-1}$, the prediction step is written as

$$\hat{\boldsymbol{x}}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{w}_{k-1}), \tag{3}$$

$$\hat{\boldsymbol{P}}_k = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1}\boldsymbol{F}_{k-1}^T + \boldsymbol{G}_{k-1}\boldsymbol{W}_{k-1}\boldsymbol{G}_{k-1}^T, \tag{4}$$

where $\boldsymbol{F}_{k-1}$ and $\boldsymbol{G}_{k-1}$ are the Jacobians of f w.r.t $\boldsymbol{x}_{k-1}$ and $\boldsymbol{w}_{k-1}$, respectively. The state and covariance propagation (3) and (4) are driven by the IMU measurements as follows

$$\dot{\boldsymbol{t}}_{\mathcal{WB}} = \boldsymbol{t}_{\mathcal{WB}} \times {}_{\mathcal{B}}\hat{\boldsymbol{\omega}} + {}_{\mathcal{B}}\boldsymbol{v} + \boldsymbol{w}_t, \tag{5}$$

$${}_{\mathcal{B}}\dot{\boldsymbol{v}} = {}_{\mathcal{B}}\boldsymbol{v} \times {}_{\mathcal{B}}\hat{\boldsymbol{\omega}} + {}_{\mathcal{B}}\hat{\boldsymbol{a}} + {}_{\mathcal{B}}\boldsymbol{g}, \tag{6}$$

$$\dot{\boldsymbol{q}}_{\mathcal{WB}} = -{}_{\mathcal{B}}\hat{\boldsymbol{\omega}}, \tag{7}$$

$$\dot{\boldsymbol{b}}_a = \boldsymbol{w}_{ba}, \tag{8}$$

$$\dot{\boldsymbol{b}}_\omega = \boldsymbol{w}_{b\omega}, \tag{9}$$

$$\dot{\boldsymbol{t}}_{\mathcal{IC}} = \boldsymbol{w}_{t_{\mathcal{IC}}}, \tag{10}$$

$$\dot{\boldsymbol{q}}_{\mathcal{IC}} = \boldsymbol{w}_{q_{\mathcal{IC}}}, \tag{11}$$

$$\dot{\boldsymbol{\mu}}_i = \boldsymbol{N}(\boldsymbol{\mu}_i)^T \hat{\boldsymbol{\omega}}_{\mathcal{C}} - \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} N(\boldsymbol{\mu}_i)^T \frac{\hat{\boldsymbol{v}}_{\mathcal{C}}}{d(\rho_i)} + \boldsymbol{w}_{\mu i}, \tag{12}$$

$$\dot{\rho}_i = -\frac{\boldsymbol{\mu}_i^T \hat{\boldsymbol{v}}_{\mathcal{C}}}{d'(\rho_i)} + w_{\rho,i}, \tag{13}$$

where ${}_{\mathcal{B}}\hat{\boldsymbol{\omega}}$ and ${}_{\mathcal{B}}\hat{\boldsymbol{a}}$ denotes the measurements of the gyroscope and accelerometer; ${}_{\mathcal{B}}\boldsymbol{g}$ is the gravity vector w.r.t. the body frame. Terms of $\boldsymbol{w}_*$ are Gaussian noises; $\hat{\boldsymbol{\omega}}_{\mathcal{C}}$ and $\hat{\boldsymbol{v}}_{\mathcal{C}}$ are the estimated camera linear velocity and rotational rate; $\boldsymbol{N}(\mu)$ is a projection matrix from the 3D vector to 2D tangent space of the bearing vector μ, $d(\rho_i) = 1/\rho_i$, and $d'(\rho_i)$ is the derivative w.r.t. time.

The update step of EKF is given as

$$\boldsymbol{y}_k = \boldsymbol{z}_k - h(\hat{\boldsymbol{x}}_k, \boldsymbol{n}_k), \tag{14}$$

$$\boldsymbol{S}_k = \boldsymbol{H}_k \hat{\boldsymbol{P}}_k \boldsymbol{H}_k^T + \boldsymbol{J}_k \boldsymbol{R}_k \boldsymbol{J}_k^T, \tag{15}$$

$$\boldsymbol{K}_k = \hat{\boldsymbol{P}}_k \boldsymbol{H}_k^T \boldsymbol{S}^{-1}, \tag{16}$$

$$\boldsymbol{x}_k = \hat{\boldsymbol{x}}_k + \boldsymbol{K}_k \boldsymbol{y}_k, \tag{17}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \hat{\boldsymbol{P}}_{k-1}, \tag{18}$$

where the innovation covariance $\boldsymbol{S}_k$ is updated by the Jacobians $\boldsymbol{H}_k$ and $\boldsymbol{J}_k$ of observation model h w.r.t. $\boldsymbol{x}_k$ and $\boldsymbol{n}_k$, respectively, and the innovation $\boldsymbol{y}_k$ is obtained by stacking innovation terms $\boldsymbol{y}_i$ of all landmarks at time k. The innovation term of landmark i is given as

$$\boldsymbol{y}_i = \boldsymbol{b}_i(\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_i)) + \boldsymbol{n}_i, \tag{19}$$

where $\boldsymbol{n}_i$ is the Gaussian noise; $\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_i))$ is a projection function from bearing vector $\hat{\boldsymbol{\mu}}_i$ to the corresponding pixel coordinates; $\boldsymbol{b}_i(x)$ denotes the patch alignment function for multilevel patch feature and the corresponding image pyramid. The detailed functions can be found in [11]. The Jacobian of $\boldsymbol{y}_i$ w.r.t. $\hat{\boldsymbol{\mu}}_i$ is derived by

$$\boldsymbol{H}_{y_i} = \frac{d\boldsymbol{b}_i(\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_i))}{d\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_i)} \frac{d\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_i)}{d\hat{\boldsymbol{\mu}}_i}. \tag{20}$$

In the above EKF framework, the bearing vector can be determined immediately when a feature is first observed. Its inverse depth converges within several updates as well. In which case, the state estimator does not need any special initialization procedure, and thus it is a truly power-up-and-go state estimator system. The framework is able to run on board with a feature number $N = 25$ at a frame rate of 20 Hz. However, it still suffers from some problems blocking its application to practical MAV navigation. Based on the idea of monocular ROVIO, we propose an efficient and robust algorithm by monitoring the inverse depth convergence and adding local bundle adjustment. Failure detection and recovery are also involved to guarantee flight safety.

B. Inverse Depth Convergence Monitoring

ROVIO is based on the EKF framework and thus is an ideal state estimation method for linear system. However, the MAV is a nonlinear system, suffering from the well-known linearization problem. To reduce the effect of inherent nonlinearities, ROVIO uses inverse depth landmark parameterization. The positions of landmarks projected in the new image frame are directly related to the inverse depth. The covariance term of the ith landmark, $L_{k,i}$, is extracted from the diagonal block element of $\boldsymbol{P}_k$ in EKF

$$\boldsymbol{L}_{k,i} = \begin{bmatrix} \sigma_{\mu 1}^2 & l_{12} & l_{13} \\ l_{21} & \sigma_{\mu 2}^2 & l_{23} \\ l_{31} & l_{32} & \sigma_{\rho}^2 \end{bmatrix}, \tag{21}$$

where σ_ρ is treated as the covariance of inverse depth value ρ because $l_{ij}|_{i,j\in(1,2,3)}$ is generally small value. Once the covariance is diverged, the estimation of landmark location suffers from large uncertainty, and the image-patch alignment needs several iterations to find the correct image patch. Otherwise, the feature may not be tracked. In addition, ROVIO executes EKF update for every tracked feature. One-by-one processing will greatly reduce the computation cost of matrix inverse calculation due to the sparsity of the Hessian matrix, compared with the stacking of all features. However, this step leads to a potential problem. If the inverse depth estimation of the first couple of landmarks diverges, then the consequent landmark estimation will be affected and further leads to system failure.

The robustness and accuracy of a visual state estimator are directly related to the convergence of the feature inverse depth. Based on ROVIO, we develop a method to improve the convergence performance. The key point of our method lies in detecting features with diverging estimation of inverse depth value and recalculating a good estimation after EKF update. To detect diverging features, the value σ_ρ in (21) for each tracked landmark is monitored. If the value is larger than a predefined threshold, the feature's inverse depth is then recalculated. The problem now is how to recalculate and achieve a good estimation. From the state update process in (17) and (18), the relative transformation $\boldsymbol{R}_k, \boldsymbol{t}_k$ between the last frame and current frame and the ith feature's positions $\boldsymbol{p}_{k-1,i}$ and $\boldsymbol{p}_{k,i}$ in the corresponding images is known. Therefore, we apply the triangulation method to recalculate the depth. Based on the epipolar geometry, we have

$$\frac{1}{\rho_{k,i}} \boldsymbol{p}_{k-1,i}^{\wedge} \boldsymbol{R}_k \boldsymbol{p}_{k,i} + \boldsymbol{p}_{k-1,i}^{\wedge} \boldsymbol{t}_k = 0, \tag{22}$$

where $\boldsymbol{p}_{k-1,i}^{\wedge}$ denotes the skew symmetric matrix of $\boldsymbol{p}_{k-1,i}$, and the inverse depth $\rho_{k,i}$ is recovered by calculating the least squares solution of this linear equation. If the depth recalculation fails by using triangulation, the feature is set as an outlier. The proposed module of inverse depth convergence monitoring is similar to some delayed initialization techniques such as [28], [29] on calculating the depth. However, differ from the delayed initialization methods, it does not delay the initialization but just add a monitor to recalculate the depth after the feature in the filter is added and initialized.

Using the above strategy, the inverse depth estimation converges faster than the original implementation, and the depth recalculation improves the accuracy of feature inverse depth estimation.

C. State Refinement With Local Bundle Adjustment

Linearity and Gaussian noise are critical for the good performance of EKF. However, they are difficult to be satisfied practically. Typically, the noise of landmark positions in pixel frame is non-Gaussian due to the nonlinear camera distortion model or the motion blur of images. ROVIO selects IEKF to eliminate such effects while here we added a local bundle adjustment (BA) procedure after the filter to get more accurate results. With the help of BA optimization method, there is no

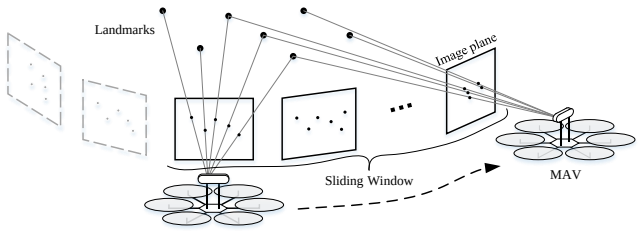


Fig. 3. Local bundle adjustment.

need to use IEKF to reduce the influence of the nonlinearity of the system. EKF, which yields less computational cost than IEKF, is enough to provide with good initial states for BA optimization so that the procedure will converge quickly. As shown in Fig.3, a sliding window in the procedure consists of at latest 20 frames, together with the converged features seen in the frames. The depth covariance of a converged feature should be below a threshold, which is set as $0.1m^2$ in our experiments empirically. A local map consisting of the converged features in the sliding window is maintained and optimized by minimizing the re-projection error of the matched features between different camera frames. The re-projection error for the jth feature in frame k is defined as

$$\boldsymbol{e}_{kj}(\boldsymbol{T}_k, \rho_j) = \boldsymbol{p}_{kj} - \boldsymbol{\pi}(\boldsymbol{T}_k, \boldsymbol{p}_j, \rho_j), \tag{23}$$

where $\boldsymbol{T}_k \in SE(3)$ is the transformation of camera frames k, and $\boldsymbol{p}_{k,j}$ and $\boldsymbol{p}_j$ denote the image coordinates of the same feature commonly observed at frame k and the frame defining ρ_j respectively; ρ_j denotes the inverse depth of the observed feature, and $\boldsymbol{\pi}$ is the camera projection function. We can construct an optimization problem to refine $\boldsymbol{T}_k$ and ρ_j by minimizing the stacked re-projection error of all features

$$\boldsymbol{T}_k, \rho_j = \underset{\boldsymbol{T}_k, \rho_j}{\arg\min} \sum_{k \in W, j \in S} \boldsymbol{L}_\delta(\boldsymbol{e}_{kj} \boldsymbol{\Omega}_{kj}^{-1} \boldsymbol{e}_{kj}), \tag{24}$$

where W and S denote the sets of fames and tracked image features in the sliding window, respectively; $\boldsymbol{L}_\delta$ is the Huber loss function, and $\boldsymbol{\Omega}_{kj}$ is the weight matrix, which is the inverse of covariance matrix associated to the feature inverse depth. By solving the optimization problem (24), all the MAV poses including current pose in the sliding window are obtained. As the initial value for optimization is obtained from the EKF-based estimation and the covariance matrix to be calculated is sparse and of small size, the Levenberg-Marquardt algorithm for the optimization problem converges very fast without costing too much time.

D. Failure Detection and Recovery

Visual-inertial systems may suffer failures because of violent illumination change or severely aggressive motion. For example, the IMU data may generate strong spikes if an MAV hits an obstacle. It may make the state estimator diverge quickly. Active failure detection and recovery can improve the robustness and safety of the system. The Kalman filter cannot detect feature outliers in measurements automatically. This condition will lead to a substantial error in velocity estimation when many outliers are involved. In general, the filter attempts to minimize the velocity error by setting the depth of features as infinity, but it leads to further divergence. Failure detection is designed as an independent thread, and the detection contains four criteria. First, the feature tracking rate is less than 20%, i.e., less than seven tracked features at each frame in our experiments. Second, a rapid change in IMU readings occurs, indicating that the MAV may hit something. Third, a large value in IMU bias estimation or a large bias for IMU extrinsic parameter estimation happens. Finally, a considerable discontinuity in the position or velocity estimations between the previous two frames occurs.

Once a failure is detected, the proposed algorithm will try to reinitialize the system. As the system does not need any special initialization procedure, the re-initialization is easy to be achieved such that the origin of new established world frame is quite close to the previous location. Then, a global localization is performed by map alignment between previous and post recovery maps.

IV. Real-time 3D Environmental Map Building

Apart from state estimation, autonomous navigation requires MAVs to perceive and map the 3D environment. Safe navigation, with the abilities of autonomous obstacle avoidance and unknown environment exploration, usually depends on a 3D volumetric representation of the operating environment. A 3D laser range finder or RGB-D camera can be used to create such a representation. However, a 3D laser is too heavy for MAVs, and RGB-D camera cannot be used under complex outdoor environments. Therefore, a stereo camera is selected in the proposed approach. Most sophisticated stereo vision algorithms are developed for ground robots or computer vision applications, and the 3D object or environmental reconstruction is time consuming, i.e., focusing on the map quality but not the running speed performance. The existing algorithms cannot be directly used for MAV's onboard navigation. To address the problem, an efficient stereo vision algorithm running in real time on MAV's onboard CPU-only computer is developed.

A. Onboard Stereo Vision Algorithm

The proposed algorithm is used to calculate the disparity from the left and right images of a binocular camera and to further obtain the 3D information of the observed scene. To accelerate the process, the commonly used approach in [30] is applied for the primary procedure, i.e., point matching between the stereo images, wherein the searching is constrained along the 1D scanline based on the epipolar geometry constraint. A group of robust matched points are detected by defining the 9x9 Sobel operator as the matching descriptor. The stereo vision algorithm requires more features than the state estimator does. Sobel operator is chosen because it is more computational-friendly and has a good response to non-corner features such as lines and edges. L1 norm is used for evaluating the difference between two points to find candidate matching, and the suspected mismatches are further removed by the left-right consistency checking.

The robust matched points are then processed using a Delaunay triangulation algorithm [31] to divide the whole image into a piecewise planar mesh. The disparity value of each

triangular vertex is already known, and the disparities inside the triangle will be calculated by applying an interpolation method. The interpolated disparities are not the real values but are treated as a prior depth prediction for further matching. To implement disparity interpolation, each 3D planar triangle in the mesh is expressed by a parameter vector, i.e., (A_i, B_i, C_i) for the ith triangle, and the plane is formulated as

$$A_i x + B_i y + C_i z + 1 = 0, \tag{25}$$

where (x, y, z) is the coordinates of the planar point expressed in the current camera frame. According to the pinhole camera model, we obtain

$$u = fx/z, \quad v = fy/z, \quad d = f\mathrm{B}/\mathrm{z}, \tag{26}$$

where (u, v) denotes the image coordinate of the planar point; d is the disparity value in pixels; B is the baseline length, and f is the focal length. Substituting (26) into (25) yields

$$A'_i u + B'_i v + d = C'_i. \tag{27}$$

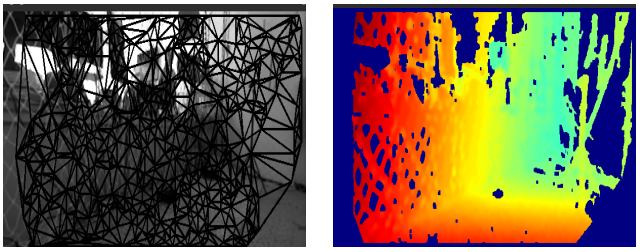
The new plane parameters (A'_i, B'_i, C'_i) are obtained using coordinates of the three known vertexes and their disparities. Given an image point (u, v) inside the triangle, the interpolated disparity d is directly obtained from (27). Based on the obtained disparity prediction, the feature correspondences search is constrained within the pixel range of $[d-20, d+20]$ along the epipolar line. The cost function for searching the best matching is defined as

$$C(d) = \textstyle\sum_W |D_1(x+i, y+j) - D_2(x+i-d, y+j)|, \tag{28}$$

where W denotes a 5×5 window, D_1 and D_2 are the Sobel descriptors of the left and right images patches at locations $(x+i, y+j)$ and $(x+i-d, y+j)$, respectively. Winner-takes-all strategy is applied to select the most likely disparity by

$$d^* = \arg\min_d (C(d)). \tag{29}$$

The optimal disparity d^* is utilized to execute Delaunay triangulation again, followed by interpolation and disparity searching on the new mesh. The entire process will be repeated, and a disparity image containing missing holes will be ultimately produced. The missing hole problem is addressed by applying the line fitting method proposed in [32]. The obtained disparity image may show incorrect spike values because of the false matchings. Therefore, a 6×6 adaptive mean filtering [33] is further executed over the disparity image to reduce noise. The proposed disparities calculation algorithm based on support points is faster than the traditional method, because it does not calculate the disparities directly by matching each pixel. The disparities of some robust support points are calculated first while others are estimated by applying an interpolation method. The proposed stereo matching algorithm costs roughly 30 ms for each 672×376 image by using only one CPU core of the i5 onboard computer without any SSE hardware acceleration. Fig.4 illustrates an example of the obtained disparity image, and most areas are constructed except for textureless areas.



(a) (b)

Fig. 4. Demonstration of the stereo vision result. (a)Delaunay triangulation result; (b)disparity image.

B. 3D Occupancy Mapping

With the disparity images of the operating environment, a 3D dense map can be reconstructed on the basis of the sequential state estimations obtained from the state estimator in Section III. Octomap [34] is utilized for the 3D reconstruction because it is robust against measurement errors, especially for temporally or spatially correlated errors received from a stereo vision system. Octomap models the world as voxels and organizes them with an octree structure. It creates an accurate environmental model from noisy sensor measurements, and the underlying measurement uncertainties, such as dynamic obstacles, have been probabilistically considered. As the proposed stereo vision algorithm balances accuracy for real-time performance, a disparity point may show an error of several pixels. However, if the error range is within the resolution of the Octomap, then the erroneous point will still contribute the same occupancy grid, preventing the disparity error from affecting the mapping accuracy. Octomap can be perceived as a method of reducing affection for matching uncertainty. Using Octree to represent the dense map can save memory and computation. For example, the lookup complexity of a random node in the map is constant by limiting octree to a fixed maximum depth d_{max}.

By integrating with the pinhole model, the 3D position of each point in the disparity image is calculated. As the MAV real-time trajectory is available from the state estimator, the obtained point cloud at each observation location can be further merged into a global dense map. Actually, calculating every frame's point cloud is unnecessary, and thus the calculation is only executed when the MAV's translation distance or rotation angle exceeds a predefined threshold. The strategy also benefits the computation cost of the proposed algorithm.

V. EXPERIMENTS

The proposed approach was verified with several experiments on a home-made MAV system, as shown in Fig.5. The MAV is designed with six rotors to supply adequate payload, and all algorithms were processed in real-time in the onboard computer Intel NUC i5-6260U. No GPU or other computation resources except an onboard CPU-based computer was used in the experiments. An onboard stereo camera (ZED) with a 672×376 resolution and 120 mm baseline was used for both state estimation and environmental perception. A Pixhawk autopilot was used for the flight control of MAV stabilization and in providing the IMU data. All the software we used are

run on Ubuntu 16.04 with ROS (Robot Operating System) as the communication framework. In the state estimator module, the visual inertial framework ROVIO in Maplab[1] is used as the base framework. In the mapping module, we utilize OctoMap [2] to efficiently build an occupied map. The informed RRT* algorithm in OMPL library [3] is used for path planning in the exploration module.

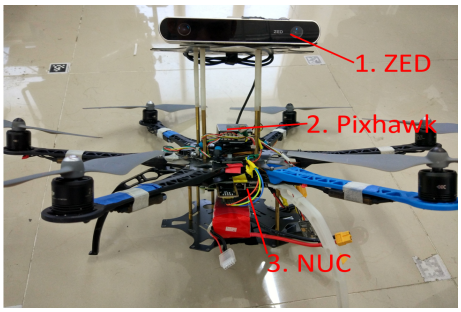


Fig. 5. Hardware system of our MAV.

A. State Estimation Experiments

We first evaluate our state estimator on the commonly used EuRoC dataset [35] for comparison. Because feature patches are used for the data association in our state estimator. We first evaluate the affection of patch size on the MAV's position estimation error. As shown in Fig. 6, experiments with sizes of 4, 6 and 8 were performed. From the figure, it is seen that both ROVIO and the proposed method perform worse with too small or too large sizes. Small patch size is difficult to achieve robust data association due to the less information and uniqueness, and therefore the proposed algorithm based on BA does not perform well when the patch size is too small. Our proposed method performs better than ROVIO when patch size is 6 and 8. A default size of 6 is pretty much appropriate value considering the speed and robustness.

For the entire dataset, the average processing time of state update with both EKF and BA optimization in the proposed algorithm on an i5-6260U-based computer is 13.884 ms compared to 12.107 ms in ROVIO. Considering the BA optimization is not performed in each EKF update, the time consumption of the proposed approach is equivalent as that of ROVIO. However, the proposed approach is more accurate.

Our experiment began when the MAV started to fly at the 43rd second of the dataset. Both ROVIO and the proposed algorithm finished the initialization immediately. Fig.7 illustrates the estimation errors of the position and yaw angle under ROVIO and the proposed algorithm, respectively, with the patch size of 6. The position estimation error is defined as the distance between the estimated position and the groundtruth. To compare the localization accuracy for the entire trajectory without letting the angle estimation error in the early stage affect the absolute position afterwards, a SE(3) alignment is performed to align the estimated trajectories with the groundtruth. Additionally,

[1]https://github.com/ethz-asl/maplab_rovio/
[2]https://octomap.github.io/
[3]https://ompl.kavrakilab.org/

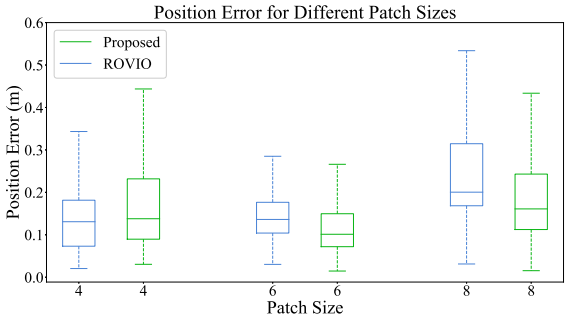


Fig. 6. Absolute position error under ROVIO and our proposed algorithm on EuRoC-MH01 with patch sizes of 4, 6 and 8, respectively. Blue box(left) denotes the errors of ROVIO, whereas green box(right) denotes those of proposed algorithm.

when comparing yaw errors, trajectories are aligned to the origin to show the drift of yaw angle. From the experimental result, it is seen that the proposed method performed better than ROVIO. The estimator runs in monocular mode with good results. Yet, it is possible to take advantage of the stereo data in our approach architecture to improve the accuracy. One method is to consider each camera as a separate monocular and estimate independently the landmarks observed in different cameras; the field of view (FOV) of the state estimator is therefore enlarged with multiple cameras. Another method is to compute the depth information of the landmarks in the perception area immediately by using epipolar geometry of the stereo data; consequently, the depth values in this undelayed initialization method will converge faster due to the good initial values. It will be our future work to integrate the stereo data into our architecture.

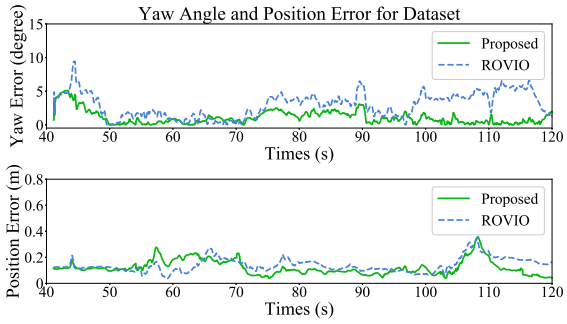


Fig. 7. Errors of position and yaw angle under ROVIO and our proposed algorithm on EuRoC-MH01. Dotted line(blue) denotes the errors of ROVIO, whereas solid line(green) denotes those of proposed algorithm.

Fig.8 demonstrates the convergence curve of the inverse depth of three randomly-selected map features. The dotted curves denote the inverse depth covariance values by using ROVIO, and the solid curve denotes those by using the proposed algorithm. The depth convergence speed was improved with our algorithm. The feature depth covariance values converge fast. Half time is required to converge to the value below 0.1, indicating that the corresponding image patch in the next frame is accurate, such that the EKF linearized point is also correct, indirectly improving the algorithm accuracy.

To demonstrate the failure detection and recovery module, the

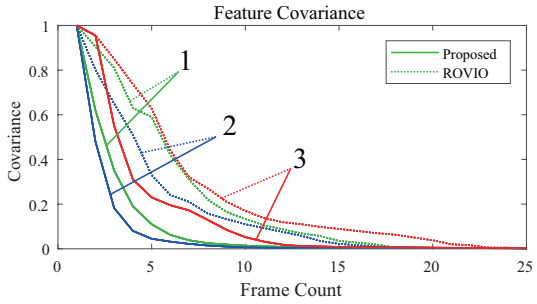


Fig. 8. Convergence of the covariance of feature inverse depth under ROVIO(dot) and the proposed algorithm(solid).

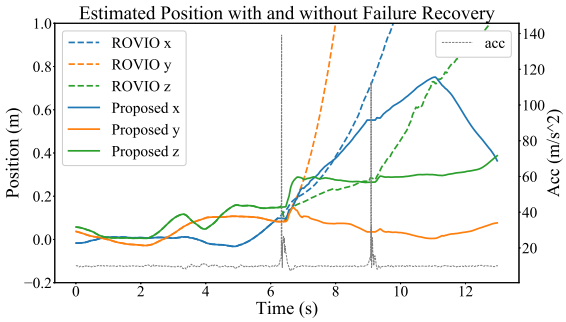


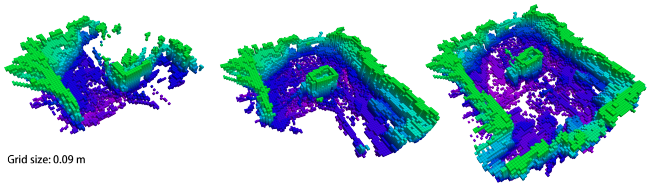
Fig. 9. Estimated position under the state estimator with and without failure recovery when IMU data is sudden changed by several strikes on module.

MAV was moved back and forth at a constant speed along the X axis, and it was hit to simulate a sudden impulse on IMU data for several times. To evaluate the performance of the proposed recovery method, we compare the localization results of ROVIO with and without recovery module. Fig.9 illustrates one of the typical state estimation result. The estimation diverged quickly under the method without recovery module when IMU data suddenly changed by a strike at about 6.5 seconds, and it performed even worse when the second strike happens at about 9 seconds. It is mainly because the bias of accelerometer is wrongly estimated. The proposed method with recovery successfully recognized the failure during the experiment. It performed recovery based on the previous pose within a very short time where the pose did not change. With the help of failure detection and recovery module, the result under the proposed method did not diverge even under several impacts. The experiment was performed 5 times, and we had the same result.

B. Exploration Experiments

An experiment was further performed in our lab environment to verify the proposed approach. Fig.10 shows the snapshots of our lab environment, containing many similar structures and textureless areas. State estimation and mapping are challenging. Similar scenes and textureless places seriously affect the dense mapping. Nevertheless, our approach continues to perform well. Fig.11 illustrates the constructed 3D mapping result. Mapping snapshots are illustrated in Figs. 11(a)-11(c). The colors in the map represent different obstacle heights. Most parts of the environment are reconstructed except a few textureless or unseen areas. Some inaccurate obstacles were observed in the reconstructed occupancy map due to the error of the MAV's state estimation and depth estimation of the entire environment. Given the inaccurate obstacles of the generated maps, the map is not suitable for localization and path planning applications in large scale. However, in our framework, a VI-map [22] is also built based on the proposed state estimator during the mapping. Then, a global bundle adjustment was performed to optimize the VI-map and MAV's trajectory poses. To evaluate the global optimization, we compared the localization in the optimized map and the raw one. The groundtruth is obtained by the motion capture system. As shown in Fig.12, the absolute position error in optimized VI-map reduced by approximately 80%.

Fig. 10. Experimental lab environment.



(a) 20s (b) 50s (c) 100s

Fig. 11. Exploration-based 3D dense mapping in an unknown environment. (a), (b), and (c) Snapshots of dense mapping at 20 s,50 s, and 100 s, respectively.

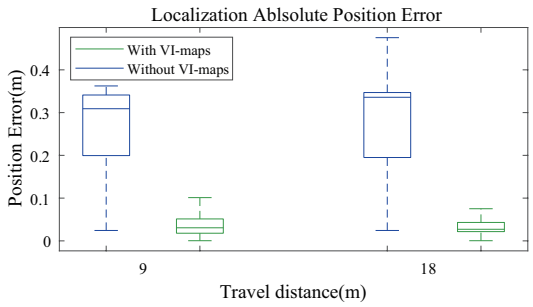
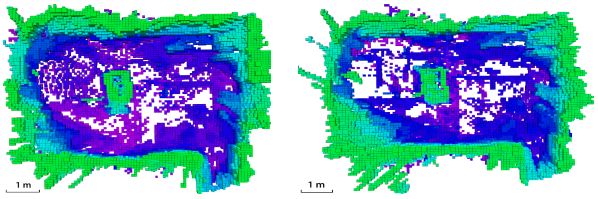


Fig. 12. Accumulated absolute position error without and with optimized VI-maps. Green box(right) denotes that with optimized VI-map.

A new Octomap was re-established by using the optimized pose, as shown in Fig.13. The optimized Octomap has less noise than the original map, and the shape of the obstacle in the middle is more accurate. The optimized Octomap was used for path planning and obstacles avoidance, whereas the VI-map was used to realize high accuracy localization. Because it is difficult to directly measure the accuracy of the built 3D map,



(a) Raw occupancy map. (b) Optimized occupancy map.

Fig. 13. Comparison of raw (a) and optimized (b) occupancy maps.

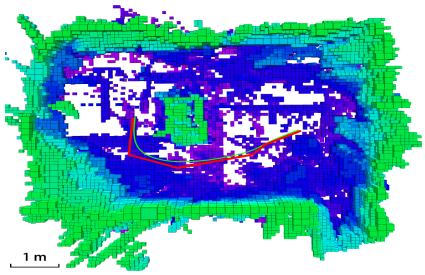


Fig. 14. Result of path planning in the optimized 3D occupancy map.

we present indirect analysis on the mapping accuracy. From the Octomap accuracy analysis in [34], the Octomap accuracy is 97% defined as the percentage of correctly mapped cells in all 3D scans. A cell in the 3D map counts as a correctly mapped cell, if it has the same maximum-likelihood state (free or occupied) in the map and the evaluated 3D scan. Therefore, the final 3D map accuracy will depends only on the algorithms of the stereo vision and state estimator. The disparity errors of 95% image pixels are below 2 pixel [30]. In our experiments, we use a ZED camera with focal length 350px and baseline 120mm. Without loss of generality, we assume that the average distance of the obstacles away from the MAV is 2m, and then the error of the depth estimation is about 20cm. By considering that the localization error of the state estimator is about 5cm, the error of the built 3D map is about 25cm. With the mapping accuracy, it is not a problem for our MAV with size 80cmx80cmx30cm to plan a safety trajectory.

C. Navigation Mission Experiments

To further verify the effectiveness of the proposed approach, a navigation experiment was performed in the optimized Octomap of our lab. The goal position was selected manually at an arbitrary area where no obstacles exist. The drone used the optimized occupancy map to plan a collision-free path successfully. The path planning was based on informed RRT* and B-spline fitting methods. As shown in Fig.14, the planned trajectory is smooth and suitable for the dynamic characteristics of the drone. Finally, the MAV flied successfully to the target.

VI. CONCLUSIONS

The study proposes an efficient approach for autonomous navigation of MAVs in unknown and unstructured environments. A visual-inertial state estimation algorithm is developed by integrating both filter- and optimization-based methodologies. Modules of local bundle adjustment in a maintained sliding window, inverse depth convergence monitoring and failure detection and recovery are proposed to guarantee localization robustness and accuracy for MAVs. A real-time stereo mapping method with autonomous exploration is developed for MAVs to achieve 3D dense map building in unknown environments. The state estimation and mapping algorithms are processed in real-time without GPU acceleration. Finally, several experiments are performed to illustrate the performance of the proposed approach. This work makes it possible to enable MAVs full navigation ability in GPS-denied environments. Our long-term goal is to enable the MAVs the abilities to grab and transport objects in unknown and unstructured environments. Aerial manipulator missions and more applications of stereo vision will be investigated based on this navigation framework in future studies.

REFERENCES

[1] F. Vanegas and F. Gonzalez, "Enabling uav navigation with sensor and environmental uncertainty in cluttered and gps-denied environments," *Sensors*, vol. 16, no. 5, 2016.

[2] L. Fang, H. Chen, Y. Lou, Y. Li, and Y. Liu, "Visual grasping for a lightweight aerial manipulator based on NSGA-II and kinematic compensation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1–6.

[3] A. Santamaria-Navarro, P. Grosch, V. Lippiello, J. SolÃ , and J. Andrade-Cetto, "Uncalibrated visual servo for unmanned aerial manipulation," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 4, pp. 1610–1621, Aug 2017.

[4] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained mav," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 20–25.

[5] T. Cheviron, T. Hamel, R. Mahony, and G. Baldwin, "Robust nonlinear fusion of inertial and visual data for position, velocity and attitude estimation of uav," in *Proceedings 2007 IEEE International Conference on Robotics and Automation (ICRA)*, April 2007, pp. 2010–2016.

[6] A. Bachrach, S. Prentice, R. He, P. Henry, A. S. Huang, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Estimation, planning, and mapping for autonomous flight using an rgb-d camera in gps-denied environments," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1320–1343, 2012.

[7] X. Zhang, B. Xian, B. Zhao, and Y. Zhang, "Autonomous flight control of a nano quadrotor helicopter in a gps-denied environment using on-board vision," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6392–6403, Oct 2015.

[8] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, "Autonomous, vision-based flight and live dense 3d mapping with a quadrotor micro aerial vehicle," *Journal of Field Robotics*, vol. 33, no. 4, pp. 431–450, 2016.

[9] I. A. Şucan and L. E. Kavraki, *Kinodynamic Motion Planning by Interior-Exterior Cell Exploration*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 449–464.

[10] Y. Gao, H. Chen, Y. Li, C. Lyu, and Y. Liu, "Autonomous wi-fi relay placement with mobile robots," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 6, pp. 2532–2542, Dec 2017.

[11] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research*, vol. 36, pp. 1053–1072, 09 2017.

[12] M. Montemerlo, S. Thrun, D. Roller, and B. Wegbreit, "Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, San Francisco, CA, USA, 2003, pp. 1151–1156.

[13] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[14] D. Eberli, D. Scaramuzza, S. Weiss, and R. Siegwart, "Vision based position control for mavs using one single circular landmark," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 495–512, Jan 2011.

[15] S. Yang, S. A. Scherer, and A. Zell, "An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1, pp. 499–515, Jan 2013.
[16] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart, "Mav navigation through indoor corridors using optical flow," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 3361–3368.
[17] V. Lippiello, G. Loianno, and B. Siciliano, "Mav indoor navigation based on a closed-form solution for absolute scale velocity estimation using optical flow and inertial data," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, Dec 2011, pp. 3566–3571.
[18] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 2815–2821.
[19] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," in *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, Oct 2009, pp. 83–86.
[20] M. Pizzoli, C. Forster, and D. Scaramuzza, "Remode: Probabilistic, monocular dense reconstruction in real time," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 2609–2616.
[21] Y. Lin, F. Gao, T. Qin, W. Gao, T. Liu, W. Wu, Z. Yang, and S. Shen, "Autonomous aerial navigation using monocular visual-inertial fusion," *Journal of Field Robotics*, vol. 35, no. 1, pp. 23–51, 2018.
[22] T. Schneider, M. T. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robotics and Automation Letters*, 2018.
[23] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an rgb-d camera," in *Robotics Research*. Springer, 2017, pp. 235–252.
[24] L. von Stumberg, V. Usenko, J. Engel, J. Stückler, and D. Cremers, "From monocular slam to autonomous drone exploration," in *2017 European Conference on Mobile Robots (ECMR)*, 2017, pp. 1–8.
[25] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 834–849.
[26] X. Wang, H. Chen, Y. Li, and H. Huang, "Online extrinsic parameter calibration for robotic camera-encoder system," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2019.
[27] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust invariant scalable keypoints," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2548–2555, 2011.
[28] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, pp. 1403–1410, 2003.
[29] J. Civera, A. Davison, and J. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, oct 2008.
[30] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Asian conference on computer vision*, 2010, pp. 25–38.
[31] H. Si, "Tetgen, a delaunay-based quality tetrahedral mesh generator," *ACM Transactions on Mathematical Software (TOMS)*, vol. 41, no. 2, p. 11, 2015.
[32] A. Hosni, M. Bleyer, C. Rhemann, M. Gelautz, and C. Rother, "Real-time local stereo matching using guided image filtering," in *2011 IEEE International Conference on Multimedia and Expo*, 2011, pp. 1–6.
[33] R. Chandel and G. Gupta, "Image filtering algorithms and techniques : A review," *International Journal of Advanced research in computer science and software Engineering*, vol. 3, no. 10, pp. 198–202, 2013.
[34] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: an efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, Apr 2013.
[35] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, 2016.