2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

Enabling computer vision-based autonomous navigation for Unmanned Aerial Vehicles in cluttered GPS-denied environments

F. Valenti[1], D. Giaquinto[1], L. Musto[1], A. Zinelli[1], M. Bertozzi[1], and A. Broggi[1]

***Abstract*— This paper presents a synthesis of techniques enabling vision-based autonomous Unmanned Aerial Vehicle (UAV) systems. A full stack of computer vision processing modules are used to exploit visual information to simultaneously perceive obstacles and refine localization in a GPS-denied environment. An omni-directional stereo-vision based setup is used to build a 3D representation of the surroundings. A fully 3D local obstacle grid, maintained through multiple frames and updated accordingly to the UAV movement, is built accumulating multiple observations coming from the 360 stereo vision sensing suite. Visual data is also used to extract information regarding the drone attitude and position while exploring the environment. Sparse optical flow collected from both front and down facing stereo cameras is used to estimate UAV movement through multiple frames. The down-looking stereo pair is also used to estimate the drone height from the ground and to refine the pose estimation in a Simultaneous Localization and Mapping (SLAM) fashion. An improved A* planning algorithm exploits both the 3D representation of the surroundings and precise localization information in order to find the shortest path and reach the goal through a three dimensional safe trajectory.**

***Index Terms*— Intelligent vehicles, autonomous UAV, indoor navigation, stereo vision, self-localization, obstacle detection.**

I. Introduction

Unmanned Aerial Vehicles are enjoying significant popularity due to their low cost and flexibility in navigation. Nowadays, they are used to fulfill various tasks where humans are unable to operate because of either hazards or physical hindrances. These include surveillance, monitoring, aerial photography, inspection, or search and rescue. In many of these cases, UAVs have to navigate through environments that may be hard to explore with remote control such as collapsed buildings or large sites. In these situations connection instability or pilot limited field of view could render remote control unavailable.

Autonomous UAVs with on-board processing would be a strong solution to the mentioned problems, as it would allow navigation in any kind of environment without the need for human supervision. Using this technology human operators would only need to initialize the UAV and analyze the collected data once the mission is completed and the aircraft has returned to the ground station.

One of the main barriers to robot navigation is the need for obstacle detection capabilities. An autonomous aircraft must be able to detect obstacles all around it to fly safely. A well-established solution for obstacle detection is given by stereo vision. Stereo cameras are relatively cheap and light if compared to other depth sensors but are also capable of providing high density measurements.

In this paper we describe an autonomous guidance system for an UAV that exploits computer vision processing for localization, obstacles detection and planning. As shown in Fig. 1a, the hardware we employed includes 4 stereo cameras, which ensure that the UAV has complete awareness of its surroundings.

The remainder of this paper is structured as follows. Sec. II presents a summary of the related work. Sec. III shows an overview of the hardware. The localization subsystem of the UAV is described in Sec. IV, while the obstacles detection and mapping algorithms are presented in Sec. V. Path planning and control modules are described in Sec. VI. Then we describe the experimental results in Sec. VII and finally we discuss them and explore future directions in Sec. VIII.

II. Related work

Recently, there has been a lot of work in enabling autonomous navigation for UAVs. A drone capable of independently taking decisions and carrying out complex tasks in an unknown scenario has to satisfy a number of requisites: it must be able to gain information regarding the environment in which it is going to operate; it needs to estimate its pose in order to perform a generically described task; it has to be able to efficiently plan and follow a safe trajectory from its position to a certain goal pose.

A. Obstacle detection

Environment awareness is a key component for developing autonomous navigation, and recently a lot of work has been done in this direction. Various approaches mainly differ with respect to which sensor setup is used to perceive the environment and which world representation is built upon collected measurements.

Gronka et al. [1] equipped a drone with a 2D laser scanner. In this case the drone is capable of carrying out simultaneous localization and mapping tasks by exploiting very precise measurements but perception and obstacle avoidance is limited to the navigation plane. In [2] Nieuwenhuisen et al. used a 2D laser scanner as main sensor mounted on a continuously rotating servo actuator to achieve 3D sensing. This approach allows to have a 3D perception of the environment without using 3D laser scanners, which would impact significantly on the payload. Incoming depth

[1]F. Valenti, D. Giaquinto, L. Musto, A. Zinelli, M. Bertozzi and A. Broggi are with VisLab srl, an Ambarella company - c/o Dipartimento di Ingegneria dell'Informazione, v.le G.P. Usberti 181/D, Parma, Italy {valenti, giaquinto, musto, zinelli, bertozzi, broggi}@vislab.it

(a) (b)

Fig. 1: Proposed system setup consists in an enhanced DJI consumer drone. (1a) Perspective view of the modified DJI Matrice 100. (1b) Bottom view with highlighted perception suite: (1) custom processing board; (2) lateral stereo pairs; (3) down-looking stereo pair; (4) front-looking stereo pair.

measurements are stored in a hybrid local multi-resolution map, where cells contain both occupancy information and perceived distances. Individual grid cells are stored in ring buffers and their size increases with distance from the drone center. Multiple ring buffers are interlaced to achieve tri-dimensional environment description. This representation is fixed in size and egocentric, enabling local planning capabilities. Their perception setup also comprises a pair of fish-eye stereo cameras for visual odometry and current pose estimation. In [3] the authors presented an autonomous UAV, equipped with a stereo camera looking forward as their main sensor. Problems arising from limited field of view were solved by accumulating measurements over multiple frames in an allocentric map. They used a tiled octree-based 3D occupancy grid map with dynamic tile caching. The nearest four tiles to the drone position make up a map that is continuously updated as incoming measurements are processed. Remaining tiles are stored on disk and dynamic caching is responsible for loading and storing tiles as the UAV moves inside the grid.

This paper presents a multi sensor setup based mainly on stereo cameras, where multiple observations are fused in a fully 3D occupancy grid. Grid size is fixed and kept egocentric in order to represent the drone surroundings and allowing for local obstacle-free path planning.

B. Localization

The problem of localizing aerial vehicles in GPS-denied conditions, such as indoor environments, using vision-based systems is well documented in literature. Most of the current state-of-art approaches are based on SLAM (Simultaneous Localization And Mapping) or Visual Odometry techniques. In [4] a monocular SLAM framework based on keyframes and FAST features is used to perform 6DOF stabilization, allowing to correct the drift, eliminate GPS dependency and enabling autonomous navigation. Von Stumberg et al. [5] have developed a system for autonomous drone exploration based on LSD-SLAM [6], one of the current state-of-art SLAM algorithms. In [7] a Visual Odometry approach based on stereo vision and SIFT [8] features is used to estimate the aircraft location.

The approach presented in this paper combines both techniques by fusing the information coming from two pairs of stereo cameras, a front-looking stereo pair and down-looking one. The system comprises three asynchronous modules: a visual odometry module, a landmark detector and a Visual Height Estimation module. These modules send their results to an Extended Kalman Filter (EKF) which is responsible for estimating both position and attitude of the drone. A detailed description of a similar filtering algorithm can be found in [9].

III. System overview

The UAV used for this project is a DJI Matrice 100 MAV [1] (see Fig. 1a). This quadrotor has a diagonal length of 65 cm and four 33.02 cm diameter propellers (11.43 cm thread pitch). The maximum load weight for takeoff is 3.6 Kg. In our tests we used a load of 2.7 Kg and each flight lasted approximately 10 minutes. Data incoming from DJI on-board sensors can be accessed through serial communication using the official SDK [2].

The custom processing board connected to the drone has been realized by Ambarella [3]. A total of 4 stereo cameras and one sonar are attached to the board, as shown in Fig. 1b. A forward-looking stereo camera is responsible for detecting obstacles in front of the UAV. These cameras are equipped with narrow-lenses and are thus suited for long distance obstacle detection. This design choice was made under the assumption that the drone could fly faster when moving forward. Two stereo cameras are mounted laterally and slightly rotated backwards. They were equipped with fish-eye lenses in order to achieve nearly omni-directional sensing. A down-looking stereo camera, used for localization purposes, completes the vision sensing suite. A detailed specification for each stereo camera is provided in Table I. The board also provides access to other sensors like accelerometers, gyroscopes and barometer, as well as WiFi and Ethernet connection interfaces. A proprietary chip developed by Ambarella is responsible for processing video signals incoming from all stereo pairs.

[1] Shenzhen, China, https://www.dji .com/matrice100
[2] https://developer.dji.com/onboard-sdk/
[3] https://www.ambarella.com/

Camera	Sensor	HFov(deg)	Focal Lenght(mm)	Base Line(cm)	Resolution(px)
Front	Rolling shutter	91	1.8	15	1920x1080
Left	Rolling shutter	175	FishEye	15	1280x720
Right	Rolling shutter	175	FishEye	15	1280x720
Down	Rolling shutter	105	2.05	15	1280x720

TABLE I: Perception setup

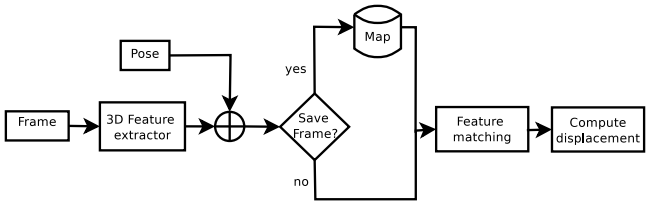


Fig. 2: Landmark localization algorithm.

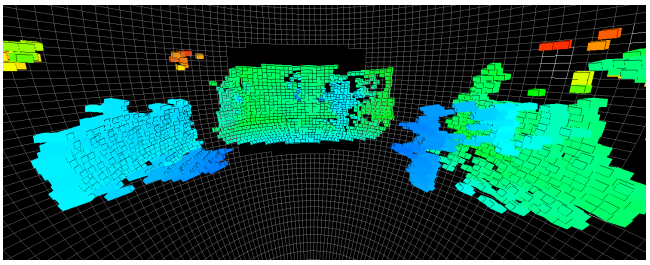


Fig. 3: Obstacle spherical grid obtained from multiple stereo cameras.

IV. Localization

Localization is the process of estimating the pose of a given reference frame with respect to another. In this case it aims at recovering the pose of the drone with respect to an inertial reference frame, usually coincident with the takeoff position. When global positioning systems are not available, localization is based on motion reconstruction techniques, such as odometry, using observations from on-board sensors. In this system, an EKF filter is capable of estimating the pose using data received from various sensors (IMU, gyroscope) and data incoming from computer vision algorithms, such as Visual Odometry (VO), Visual Height Estimation (VHE) and Landmark Localization.

A. Landmark Localization

The landmark localization algorithm is responsible for estimating the drone displacement with respect to the position of a fixed landmark. Such information is used by the localization filter to correct the drone pose and recover from the odometry drift. This algorithm, which has been favored in contrast to a pure SLAM algorithm because of strict memory consumption constraints, is designed to simultaneously build and exploit a map made of keyframes, also referred to as landmarks. A keyframe contains both image-related 3D feature descriptors and pose information. The pose contained in each keyframe represents the drone pose with respect to the map when that keyframe was computed.

This algorithm takes as input a pair of stereoscopic images as well as the current drone pose. From each pair of images, 3D feature descriptors are extracted and, along with the current drone pose, this information is used to build a current keyframe. A keyframe nearest to the current drone position is searched for in the map. If the extracted landmark is too far from the current one, the currently computed keyframe is stored in the map. Otherwise, both extracted and current keyframe are used for estimating the landmark pose with respect to the camera reference system. This is done by comparing 3D descriptors using a brute-force search based on the hamming-distance metric.

B. Visual Odometry

UAV odometry is estimated exploiting images acquired by both front-looking and down-looking stereo cameras. The distortion problems caused by the adoption of rolling shutter cameras were addressed by performing a compensation step, as described in [10]. For each stereo pair the same pipeline is executed. 3D visual descriptors are extracted from a pair of stereoscopic images, then those features are searched for in the previous frame and a relative camera motion is estimated in a standard way, as described in [11]. Finally, an optimization step is performed by combining the odometry estimation extracted from each stereo camera.

C. Visual Height Estimation

As mentioned previously, a down-looking stereo camera is used to estimate UAV height. This estimation is useful for localization purposes, as it provides height corrections that would otherwise be unavailable in indoor environments. This algorithm takes as input a disparity map and provides as output the height above a suitable perceived plane. The input disparity map is initially sub-sampled in order to reduce the amount of data to be processed. During this process only maximum disparities are taken into consideration, since they represent the nearest surface in case that points belonging to different objects are collapsing into the same pixel. The sub-sampled disparity map is used to compute a 3D point cloud representing a perceived portion of environment beneath the drone. This point cloud is then processed and an iterative plane fitting algorithm is used to estimate a plane model. Distance between the drone and computed plane is then used as height estimate and a landing feasibility measure is computed over the model.

V. Obstacles detection and mapping

As outlined before, the obstacle detection sensor setup comprises three stereo cameras, providing depth information from all-around the drone.

A disparity map incoming from each stereo camera is processed and a dense point clouds is extracted. Perceived 3D

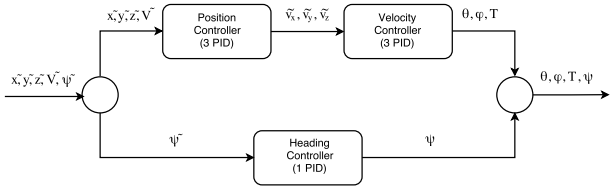


Fig. 4: Control algorithm scheme.

points, expressed with respect to the drone reference frame, are converted from cartesian (x, y, z) to spherical (ρ, θ, ϕ) coordinates.

In order to build a compact 2D representation of the environment without losing sensible depth information, we decided to project the obtained points cloud into a bi-dimensional drone-centric spherical grid. A linear mapping on (θ, ϕ) is then used to retrieve, for each spherical point, in which cell it is projected. Therefore only ρ is stored in a set D of depth measurements contained in each cell. For each cell, an overall depth measurement is then estimated from D. A sample result from this step is depicted in Fig. 3. Obstacle mapping is performed using a well-known occupancy grid mapping algorithm [12]. The map is egocentric, meaning that the grid content is described with respect to the drone current position. The egocentric property is maintained by adjusting the map content accordingly to the drone movement. This is done only when the drone exits from a cell, which allows to keep the computational load relatively low at the expense of introducing a minimal parallax error. The previously computed spherical grid is then used, in combination with an inverse sensor model, for estimating occupancy probabilities. A detailed description of the inverse sensor model used can be found in [3] (Sec. 5.2). These probabilities are used to update map content and iteratively obtain a probabilistically accurate description of the environment state. An explanatory result of this stage is depicted in Fig. 9b.

VI. Planning and Control

Path planning is a crucial part for any autonomous vehicle, since it is responsible for delivering obstacle-free navigation instructions. Control module then ensures that the computed instructions are followed with maximum precision.

A. Planning

The planning module takes as input a binary obstacle grid, which is obtained by applying a threshold over the occupancy values contained in the obstacle grid previously computed. Cells classified as obstacles are then expanded taking into account the drone size. This step is necessary since the planning algorithm makes the assumption that the drone is collapsed into a single point. Next, an improved A* algorithm searches over the cells space for the shortest path connecting the drone actual position to the target position. By adding the constraint that the planned trajectory cannot pass through any obstacle cell, it is ensured that

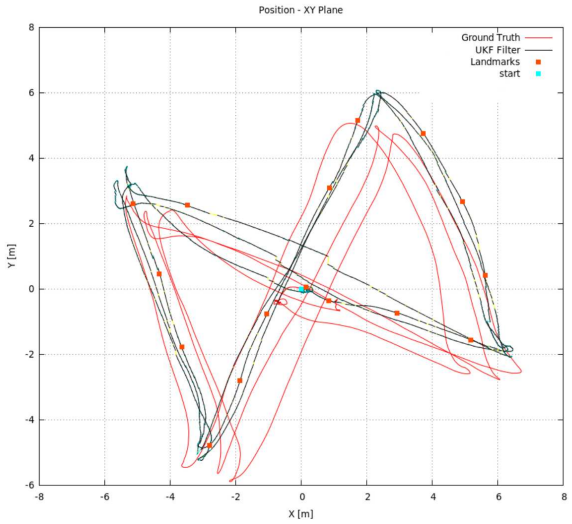


Fig. 5: Localization results compared to accurate GPS signal.

a collision-free path is always followed. The computed trajectory is smoothed using a spline fitting approach in order to satisfy continuity and control feasibility constraints. A set of waypoints is then sampled from this trajectory, each of which contains a desired position (x, y, z), velocity V and heading ψ.

B. Control

Planning results are then fed to a control module that is responsible for making the drone actually follow the planned trajectory. Given the drone current pose and velocity, a reference point is extracted from the set of waypoints and fed to the control module. This module has to ensure that the drone reaches the reference point by minimizing simultaneously position, velocity and heading errors. This control problem is solved by computing attitude (θ, ϕ) and thrust T signals and separately heading ψ, as shown in Fig. 4. These signals are then sent to the flight controller that controls the motors, using the DJI onboard SDK.

VII. Results

In this section we will present some results obtained with the presented platform.

Results from the individual parts as well as the overall system will be shown.

A. Localization

In order to evaluate the accuracy of our localization system, we performed some tests outdoor and used GPS signal as ground truth. In each test the drone was forced to perform takeoff and landing at the same position. This was used, as explained later, to further verify localization accuracy. During these tests the drone was instructed to fly always at an altitude of 2 meters and through four waypoints in a particular order. An example of such test is shown in Fig. 5. In this case the red line represents the ground truth (GPS) and the black one depicts the position as computed by the localization filter. Regarding the latter one, it is possible

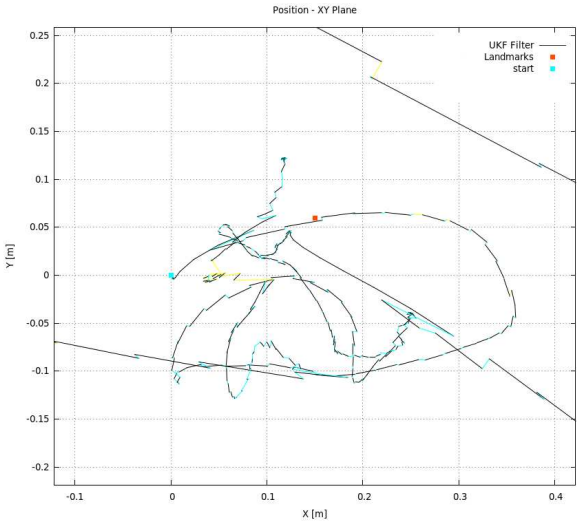


Fig. 6: Detail of the localization results.

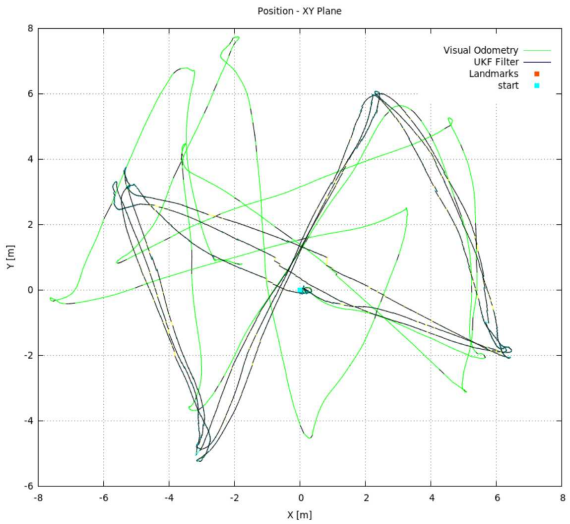


Fig. 7: Localization results compared to the Visual Odometry signal.

to observe a number of red squares that correspond to the keyframes taken by the Landmark Detector and used to send corrections to the filter, depicted in yellow. Finally there are further corrections that represent the contribution of the Visual Odometry, shown in blue. A detail of the localization result, taken in the correspondence of the takeoff and landing position, is shown in Fig. 6. In this case it is shown that the error given by the localization filter is less than 20 cm. Some comparative tests were made among certain inputs of the localization filter and the overall estimation result. For example in Fig. 7 it is shown how the Visual Odometry alone can lead to incorrect results due to significant drift in pose estimation. In contrast by exploiting other visual information the filter is capable of performing a correct estimation, as shown by the black line. Other tests as shown in the Fig. 8 show how the drone manages to maintain the required altitude and how the spikes from the VO are compensated by the filter through the use of the other data, also we note

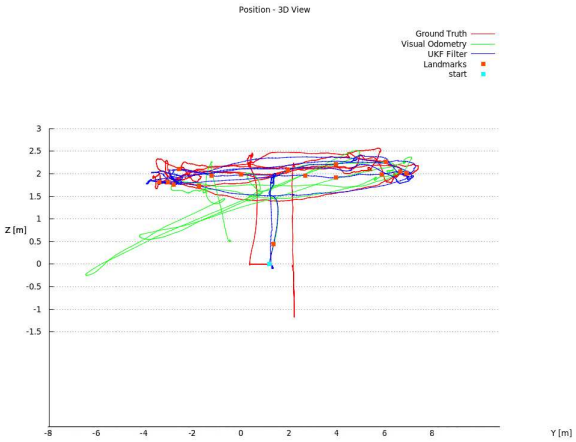


Fig. 8: Lateral view of the localization results.

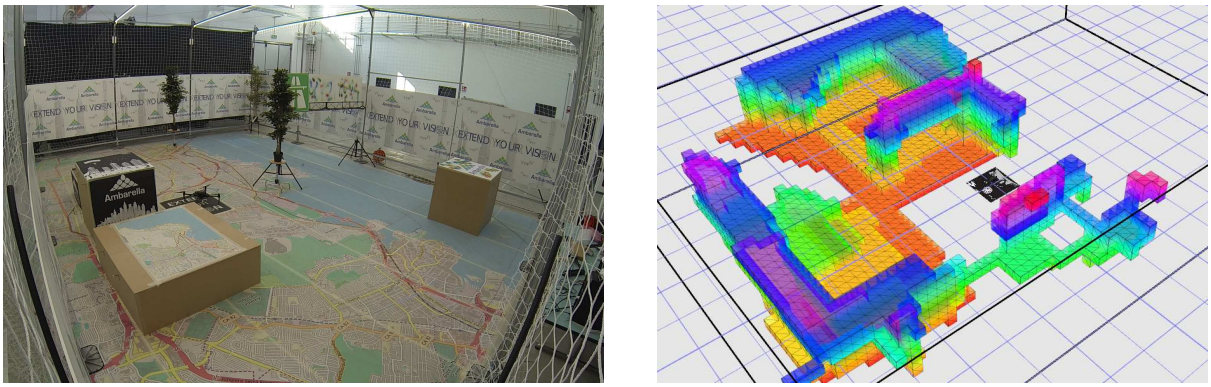
how the system is more reliable than the gps in providing the position during the landing phase.

B. Obstacles detection and mapping

The obstacle detection and mapping system exploits depth information collected from multiple stereo cameras in order to build a representation of the environment. Its first step comprises the projection of perceived 3D points into a local spherical grid. The configuration used during the tests is such that an omni-directional area around the drone is covered by the grid. Top and bottom areas where left uncovered because of the lacking of measures. Therefore the grid covers an area defined by ranging over angles zenith $\theta \in [20°, 160°)$ and azimuth $\phi \in [-180°, 180)°$. Each cell covers and area of 2x2 degrees, resulting in a grid containing 180x70 cells. The spherical grid is then projected onto a 3D egocentric occupancy grid map. Since all the tests were done indoor, in the environment depicted in Fig. 9a, the grid were configured to cover an area of 16x16x5 m. On each dimension a constant discretization step of 25 cm was used, since it is comparable to the disparity accuracy at a distance of about 10 m. With this configuration we obtained a grid fixed in size made of 64x64x20 cells. Each cell contains an occupancy value stored in 16 bits yielding an overall memory utilization of 160 KB. Each cell can also be classified with respect to its occupancy as containing an obstacle or a free area. This information can be stored in a single bit, which will further reduce memory utilization to 10 KB in case eight consecutive obstacles cells are stored in a single byte. An example of obstacle detection and mapping is show in Fig. 9b. In this case accumulation over multiple frames was disabled, such that the field of view of perception is better illustrated. It can be seen how different pieces of four walls are simultaneously detected by all cameras, providing a reasonable free-space estimation of the environment even if takeoff has not been performed yet.

VIII. Conclusions and Future Works

We presented an autonomous drone setup, along with a set of computer vision-based algorithms whose combination enables self-flying capabilities. The challenge



(a) (b)

Fig. 9: Obstacles detection and mapping results. (9a) Our indoor testing environment. (9b) Perceived obstacle grid if the drone is placed over the rightmost platform.

of self-localization has been tackled by using a Simultaneous Localization and Mapping (SLAM) algorithm. This approach uses extracted 3D features as landmarks. The generality of these landmarks makes our platform capable of navigating in any scenario and simultaneously localizing itself. This algorithm works in pair with an EKF that fuses data incoming from other sensors (IMU, Visual Odometry, VHE) in order to provide localization information with centimeter accuracy. Our current limitation is represented by memory constraints. The map of keyframes is automatically acquired and continuously updated with new landmarks, which are stored in memory as the drone moves. Therefore the explorable area that the drone can reach without needing a map reset is currently limited in size. This problem can be addressed by adding a caching strategy that keeps in memory only the landmarks nearest to the current position and stores the remainder of the map on disk. In case that a secondary storage is not available, those parts of the map can be also discarded under the assumption that it is not likely that the drone will pass over those landmarks again. Precise localization information is a strong requirement for both obstacle mapping and the planning module. Three stereo cameras mounted on-board are able to collect depth information from all around the drone. Thus the proposed system is capable of perceiving obstacles along any direction. This amount of data is processed and accumulated over time in order to build an egocentric 3D grid representing occupancy information in the surroundings. This obstacle detection and mapping algorithm can be used in a wide variety of environments, but its dependency from stereo matching disparity makes it subject to weaknesses similar to those of stereo matching algorithms. For example, featureless or low contrasted regions will result in absent or noisy disparity maps. This problem can be overcome by using other sensing strategies. For instance a depth map estimated from a sequence of monocular images and the currently computed disparity map could be fused together in order to obtain an accurate and rich depth information.

REFERENCES

[1] Slawomir Grzonka, Giorgio Grisetti, and Wolfram Burgard. A fully autonomous indoor quadrotor. *IEEE Transactions on Robotics*, 28(1):90–100, 2012.

[2] Matthias Nieuwenhuisen, David Droeschel, Marius Beul, and Sven Behnke. Obstacle detection and navigation planning for autonomous micro aerial vehicles. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 1040–1047. IEEE, 2014.

[3] Lionel Heng, Dominik Honegger, Gim Hee Lee, Lorenz Meier, Petri Tanskanen, Friedrich Fraundorfer, and Marc Pollefeys. Autonomous visual mapping and exploration with a micro aerial vehicle. *Journal of Field Robotics*, 31(4):654–675, 2014.

[4] Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments. *J. Field Robotics*, 28:854–874, 2011.

[5] Lukas von Stumberg, Vladyslav Usenko, Jakob Engel, Jörg Stückler, and Daniel Cremers. From monocular slam to autonomous drone exploration. In *Mobile Robots (ECMR), 2017 European Conference on*, pages 1–8. IEEE, 2017.

[6] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 834–849, Cham, 2014. Springer International Publishing.

[7] Syaril Azrad, Mohammad Fadhil, Farid Kendoul, and Kenzo Nonami. Quadrotor uav indoor localization using embedded stereo camera. In *Applied Mechanics and Materials*, volume 629, pages 270–277. Trans Tech Publ, 2014.

[8] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004.

[9] Simon Lynen, Markus W Achtelik, Stephan Weiss, Margarita Chli, and Roland Siegwart. A robust and modular multi-sensor fusion approach applied to mav navigation. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3923–3929. IEEE, 2013.

[10] Olivier Saurer, Marc Pollefeys, and Gim Hee Lee. Sparse to dense 3d reconstruction from rolling shutter images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3337–3345, 2016.

[11] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part i: The first 30 years and fundamentals. *IEEE Robotics and Automation Magazine*, 18(4):80–92, 2011.

[12] Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.