An MAV Localization and Mapping System Based on Dual Realsense Cameras

Yingcai Bi[*], Jiaxin Li, Hailong Qin, Menglu Lan, Mo Shan, Feng Lin and Ben M. Chen
National University of Singapore, Singapore

Abstract

Online localization and mapping in unknown environment is essential for Micro Aerial Vehicles (MAVs). Both accuracy and robustness are required in the realtime applications. In this paper, we present a dual camera system to estimate the pose of an MAV and generate an obstacle map for navigation. The recently released light-weight Intel Realsense depth cameras are utilized to build the system, one is forward-facing and the other is downward-facing. The downward-facing camera provides a high-frequency (60hz) velocity measurement while the forward-facing camera computes a low-frequency (10hz) position measurement. Experiments demonstrate the good performance of our proposed system.

1 Introduction

MAV is an efficient tool for multiple tasks, such as inspection and exploration, due to the small size and high maneuverability. The development of autonomous MAVs is a hot topic in both research community and commercial market. Currently, most of the MAVs rely on Global Positioning System (GPS) and Inertial Management Unit (IMU) to localize and fly autonomously in open space. Some MAVs are integrated with camera-based velocity estimation component (optical flow) to stabilize while GPS signal is not available. However, it is still very challenging for a more complete task like autonomous fly in GPS-denied environment. To achieve this, the MAV needs an intelligent navigation system with the capability of localization, mapping, obstacle avoidance and planning. The focus of this paper is to develop an efficient Simultaneous Localization and Mapping (SLAM) system for MAVs.

SLAM is a problem of building map and simultaneously keeping track of location for a robot in unknown environment. The challenges are mainly two-fold. On one hand, the hardware including processors and sensors should meet the requirement of size, weight, computation and power consumption. On the other hand, the software including software architecture and algorithms should meet the requirements of robustness and efficiency. Currently, the main sensors used for SLAM are laser scanners and cameras. There are already many robust and precise laser based SLAM solutions. However, integration of laser scanners on MAVs is still not efficient enough because of size, weight and cost limit. Comparing with laser based SLAM, visual SLAM is more promising. Cameras can provide much more data than laser scanner with smaller size and cheaper price. But, because of the difficulties in handling noise and image understanding, there are still few complete and robust solutions for visual SLAM.

In this paper, we will discuss the problem of how to deploy a complete visual SLAM system on MAVs. By investigating state-of-the-art methods from the perspectives of both hardware and software, we propose our solution to equip our MAV a realtime robust visual SLAM system. The investigated methods concern laser scanner, monocular camera, stereo camera and RGBD camera based methods, especially those representative methods which push the realtime robust performance to practical application phase in last several years. In our work, we choose Intel Realsense cameras (Fig. 1) as our visual sensors, Intel x86 portable PC as our processor, ORB-SLAM as our SLAM framework. The choice is discussed in detail and the implementation and experiments are covered in this paper.

The contributions of this work are as follows: We address the challenges and current achievements on visual SLAM which become practical to use in real world scenarios. Also we show the early attempt to deploy Intel Realsense depth camera on MAV to demonstrate a complete SLAM solution. Furthermore, we present the point cloud filtering and octomap building modules, which are the essential components for obstacle avoidance and path planning.

Figure 1: Intel Realsense camera

2 Related Works

According to different sensor type, the SLAM method varies. In this section, we will discuss current methods based on three categories. First is laser based SLAM, which has the longest history due to the high accuracy of range

[*] Email address: yingcaibi@u.nus.edu

measurements. Second one is visual SLAM, which is the hot topic nowadays. Third one is visual odometry, which can be considered as a subset of visual slam without loop closing. The visual odometry discussed here is a good way to estimate the short-term motion and benefit the visual SLAM method. Furthermore, we discuss about some existing applications on MAVs.

2.1 Laser SLAM

For the laser based methods, we discuss two representative ones. Kohlbrecher et al. present a 2D laser based method named Hector SLAM [1]. It is a robust localization and mapping method already verified on many practical applications. Hector SLAM calculates the relative position and orientation by aligning the incoming scan with the map maintained by all previous scans. The disadvantage of Hector SLAM is that it can only work under the assumption of 2D workspace. It needs to utilize more sensors to get a complete 6DOF estimation. Ji Zhang et al. propose an impressive 3D laser based localization and mapping method, which is called LOAM [2]. LOAM uses only one 3D laser scanner, either the rotating one-axis or non-rotating two-axis laser scanner. It solves the problem that range measurements are not synchronized and suffer severe distortion with external motion. It divides the complex problem of SLAM into two steps. One is distortion correction and fine registration of point cloud. The other is motion estimation at high frequency to estimate velocity of the laser scanner. The algorithm can provide accurate motion measurements and point cloud registration iteratively with both low-drift and low-computational complexity.

2.2 Visual SLAM

Klein and Murray propose Parallel Tracking And Mapping (PTAM) method for estimating the camera pose in a small-scale unknown scene [3]. This is the milestone of feature based realtime visual SLAM method with the first proposed idea of concurrent tracking and mapping. One thread is responsible for tracking the camera motion relative to the local map while another thread maintains a 3D map built from 3D points in keyframes by using bundle adjustment[4]. PTAM beated all the filter-based real-time methods at that time, including the famous EKF-SLAM and FastSLAM. However, there are still several shortages in PTAM including the less robustness of feature points, lack of efficient loop closing and lack of large scale environment handling.

Felix Endres et al. propose a fully functional SLAM method based on RGBD camera called RGBDSLAM [5]. They develop a typical feature based graph slam pipeline, which can detect the closing loop and optimize on the pose graph. Since they use frame to frame feature matching to find connection in pose graph, the computational complexity is high. Also, they don't handle the graph size efficiently, which will make the system slow down when the map size keeps growing.

Raúl Mur-Artal et al. recently propose ORB-SLAM, which is a comprehensive integration and innovation with the best ideas proposed in previous feature based methods [6, 7, 8]. The drawbacks of previous PTAM and RGBDSLAM are solved. It can work on both monocular and depth camera, showing a great potential for practical applications. Competing with the best feature based methods, Jakob Engel et al. propose a Large-Scale Direct Monocular SLAM (LSD-SLAM) method in ECCV 2014 [9]. It does not use corners or any other local features, but performs direct tracking by image-to-image alignment. Coarse-to-fine pyramid searching with a robust Huber loss is used to handle large motion and outliers. Extensions based on omnidirectional camera and stereo camera are also presented [10, 11] to solve the absolute scale and handle strong rotation.

2.3 Visual Odometry

Andreas Geiger et al. propose a realtime stereo vision based 3D reconstruction method called libviso2 [12]. They use simple blob and corner detector as features and calculate 3D coordinates of the features through triangulation in the previous image pair. The optimal motion is calculated by minimizing image reprojection error. Albert S. Huang et al. propose a visual odometry and mapping method using a RGB-D camera called libfovis [13]. It uses the FAST feature detector with an adaptive threshold on three Gaussian pyramid levels. They use similar method as Geiger to get the motion. Both algorithms use feature descriptors that are not invariant to rotation or scale changes. Therefore the running frequency of both algorithms need to be high to cope with movements.

Christian Forster et al. propose a Semi-Direct Visual Odometry (SVO) algorithm that is precise, robust, and fast. SVO detects feature points and find feature correspondence by direct motion estimation instead of feature matching. Thus, it achieves an increased speed by avoiding feature extraction in every frame, and increased accuracy through subpixel feature alignment. The problem of SVO is that it is designed and limited to work on global-shutter high-framerate downward facing camera.

2.4 Applications on MAVs

There are some real-time applications of SLAM methods on MAVs. Bry et al. presented impressive flight results for both fixed-wing and quadrotor using a 2D laser scanner and sensor fusion to get 3D state estimation. There are also some different designs from the literature [14, 15, 16]. Bachrach et al. proposed an estimation, planning and mapping solution using a RGBD camera [17]. Schmid et al. presented an autonomous navigation solution based on stereo vision [18]. They demonstrate that stereo vision works in both indoor and outdoor environments with a Core2Duo board and a FPGA card for the heavy computation.

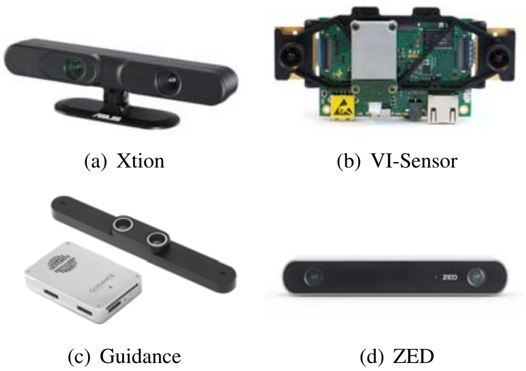
3 Hardware Selection

SLAM for MAV is a systematic problem. The core of algorithm development becomes mature, but the success still lies on a complete hardware-software solution. There is still not any commercial visual SLAM solution for MAV in the market while the recently released Dyson 360 Eye Robot Vacuum demonstrates a 360-degree camera based SLAM solution. In this section, we investigate the available visual sensors for MAV and discuss about the usability for a fully autonomous MAV.

Laser Scanner or Lidar is a classic accurate range measurement device based on the calculation of laser flight time. It can easily achieve centimeter accuracy working in both indoor and outdoor environment. However, to get a dense 3D scanning result, additional optical components and motors are needed. This make the laser scanner expensive and bulky. For example, the latest model of small-size 3D lidar from Velodyne weighs more than 800 grams [1]. On the other hand, camera obviously is a more compact solution as a 3D sensor. A typical camera weighs only 20 grams and can provide an image resolution up to 4K. Also, camera is cheap and accessible everywhere. However, the deployment of cameras on MAVs rely heavily on the hardware and software design to achieve enough accuracy.

Monocular camera is the most compact configuration, but monocular SLAM method is up to unknown scale and must include a metric initialization process to solve the scale problem. Stereo cameras are widely used in many industry applications for measuring the distance of objects by triangulation and demonstrated to be more robust. RGBD cameras based on structure light or Time of Flight (ToF) becomes very popular in recent years. RGBD cameras can provide depth information, thus share many properties with stereo cameras. The differences between them are the range and spatial density of depth data. Since RGBD cameras calculate depth with an active infrared structure light projector, they can estimate depth in areas with poor visual texture, but the range is limited and can only be used indoor. On the other hand, stereo cameras based on image matching are limited to rich texture environment but work well in outdoor environment. The stereo and RGBD camera based SLAM methods degrade to the monocular ones if depth is partially or totally unknown due to the low-texture or strong illumination change.

Kinect is the first commercial available RGB-D camera released in 2011 by Microsoft. It's demonstrated to be a huge success in the area of Human Computer Interface (HCI). Since the launch, it has been widely used on robots such as ground robots, service robots and MAVs. As the successor of Kinect, Asus Xtion(Fig. 2(a)) become popular because of the compact size and less power consumption. The VI-Sensor (Fig. 2(b)) is a light-weight, time-synchronized stereo camera for visual-inertial applications. It features a high-quality global shutter stereoscopic camera and an industrial-grade inertial measurement system. DJI Guidance (Fig. 2(c)) is another off-the-shelf stereo camera for MAVs. It uses ultrasonic sensors and cameras to gather realtime information about its surroundings. ZED stereo camera (Fig. 2(d)) developed by StereoLab is another popular options for MAVs with the capability of realtime depth calculation by GPU.



(a) Xtion

(b) VI-Sensor

(c) Guidance

(d) ZED

Figure 2: Camera options

Table 1: Camera comparison

Camera	Weight	Range	Environment	Depth
Xtion	Middle	Middle	Indoor	Yes
VI-Sensor	High	Middle	Indoor/Outdoor	No
Guidance	Middle	Middle	Indoor/Outdoor	Yes
ZED	Middle	Long	Indoor/Outdoor	Yes(GPU)
Realsense	Low	Middle	Indoor/Outdoor	Yes

Intel Realsense R200 camera has one camera providing RGB image and one stereo infrared camera producing depth. Table 3 shows a comparison of different cameras considering weight, range and operation environment. Realsense camera is the lightest one, and can work both indoor and outdoor with the help of a laser projector. Compared with standard visible light stereo camera like VI-sensor, Guidance and ZED, Realsense can provide short range depth information even in textureless environment. The indoor range is approximately 0.5-3.5 meters and outdoor range is up to 10 meters. In this paper, we try to push the performance of small size Realsense cameras to get a good localization and mapping result for MAVs.

4 Camera Model for Realsense R200 Camera

To fully configure and customize the algorithm, we use only the raw color image and depth image from Realsense camera for processing. The rgb image is aligned with the color camera, while the depth image is aligned with the left ir camera. Also, the resolution of them could be different according to configuration. Therefore, the color and depth

[1] http://velodynelidar.com/vlp-16.html

camera registration is needed to align the pixels from two different cameras.

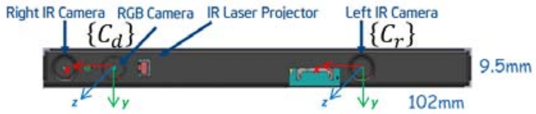


Figure 3: Realsense camera model, color camera coordinate system is based on RGB camera, depth coordinate system is based on left IR camera

As illustrated in Fig. 3, we define the corresponding coordinate systems. The depth coordinate system is denoted as C_d and color camera coordinate system is denoted as C_r. A 3D point coordinate in C_d is represented as P_id, where i is the index number. Similarly, A 3D point coordinate in C_r is represented as P_ir. The extrinsic matrix $[R_{rd}|t_{rd}]$ determines the transformation between P_id and P_ir.

$$P_ir = [R_{rd}|t_{rd}]P_id \tag{1}$$

With the 3D point coordinates in color camera frame, we can project these 3D points to image plane so that rgb pixels are aligned with their depth. The pinhole camera model is used to calculate the intrinsic matrix. The projection function is defined as,

$$\pi(P) = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \times P_ir \tag{2}$$

In our implementation, the extrinsic rotation matrix is an identity matrix and the translation only happens in the y direction. The image distortion is not included in the intrinsic matrix since the image is already rectified. We can get the related intrinsic and extrinsic calibration parameters from the factory calibration provided by the driver. This alignment procedure makes the system flexible to use diverse stream simultaneously.

5 Localization and Mapping

5.1 Downward-facing Velocity Estimation

We estimate the 3D velocity by following the state-of-the-art feature based visual odometry method enhanced by the depth image [19]. The main advantage of this method is that color images and depth images are processed separately without the need of strict synchronization. This method matters because we observe that the color image and depth image from Realsense camera can not always arrive at the same time.

The algorithm structure is shown in Fig. 4. FAST corner features are detected from current image and tracked in the subsequent images by KLT optical flow method. The depth information of the visual features could be known and unknown. The corresponding known depth is searched in a k-d tree point cloud map which is built with all previous tracking poses and depth images. Those feature points with unknown depth can be tracked between multiple frames and triangulated to get the depth. The pose is calculated using Gauss-Newton optimization to minimize the projection error between consecutive two images. A point cloud map is maintained using the pose interpolation according to the timestamp.

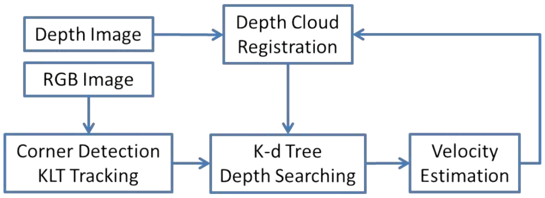


Figure 4: Velocity estimation algorithm structure

5.2 Forward-facing SLAM

As discussed in section 2, ORB-SLAM is currently one of the best SLAM method which benefits quite a lot from the feature based SLAM research in recent years. There are three concurrent threads in ORB-SLAM, namely tracking, mapping and loop closing. In the tracking thread, a constant velocity motion model is used to predict current camera pose. By searching the corresponding map points observed in the last frame, current camera pose is optimized. If tracking is lost due to small feature number, the frame will be converted into bag of words and queried in the recognition database to do global relocalization.

In the mapping thread, keyframes from tracking thread are inserted to the map as fast as possible followed by a mechanism to cull redundant keyframes. This make the tracking more robust to fast camera motion while keeping the efficiency of mapping. A covisibility graph is maintained to do local map optimization for both camera pose and map points.

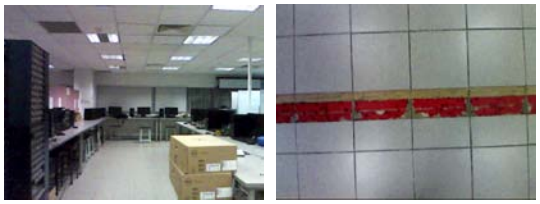
In the loop closing thread, the image candidates sharing the similar view will be searched. The search is accomplished efficiently thanks to the inverse index in the vocabulary tree, which stores weights of the words in the images they appear. With the detected loop candidates, an essential graph is generated to optimize the global map with minimal pose.

5.3 Octomap Generation

Octomap is a compact representation of obstacles for MAV navigation. With the robust pose estimation, an octomap can be generated and used for path planning. Since the depth image from Realsense camera is noisy, we need to do filtering on it before inserting to octomap. In our implementation, we transform the depth image to point cloud and apply three different kinds of filters on point cloud. First, we use a voxel grid downsampling filter to reduce the data size according to the map resolution required by the planning module. Second, we transform the point cloud from camera frame to the map frame and use a pass through filter to cover the specific height range. This filter can be tuned to remove the ground plane and further reduce data size. At last, an outlier filter is used to remove small patch obstacles, which highly possible to be sensor noise. The filters make sure we get a fine quality global map for planning use.

6 Experiments

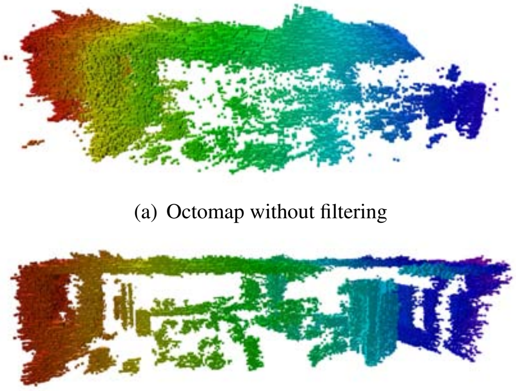
The indoor test images are captured in our lab, which is a typical clustered indoor environment. Two Realsense cameras are mounted on a rectangle box. one is facing forward and the other facing downward. The box is hand carried and moved in a rectangle path. We assume the two cameras are perpendicular to each other according to the installation. Fig. 5 shows the color image samples from both forward-facing camera and downward-facing camera.



(a) Forward image sample (b) Downward image sample

Figure 5: The indoor environment for experiments

Fig. 6 shows the octomap built from the depth cloud of forward camera. The designed filters can effectively remove noise and make the map clearer. The noise is significant because Realsense relies on a laser projector to generate infrared pattern, which is not very stable under indoor light condition. We can remove them assuming the depth continuity of the obstacle. This assumption is true because the depth is not negligible when the obstacle gets closer to the camera.

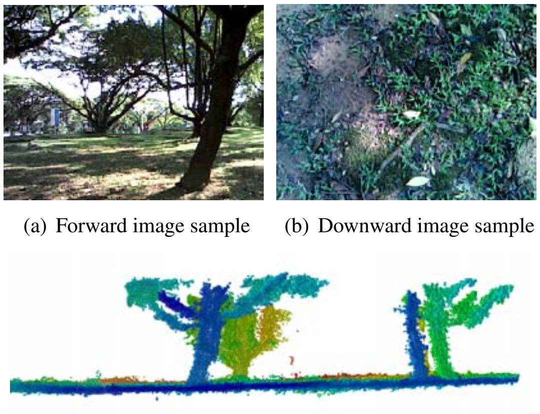


(a) Octomap without filtering

(b) Octomap after filtering

Figure 6: The octomap result

We also test the outdoor capability of our system in a typical canopy environment. The octomap is built using the same method as indoor, but the range in outdoor environment can be increased up to 10 meters. The outdoor environment and octomap result are shown in Fig. 7.

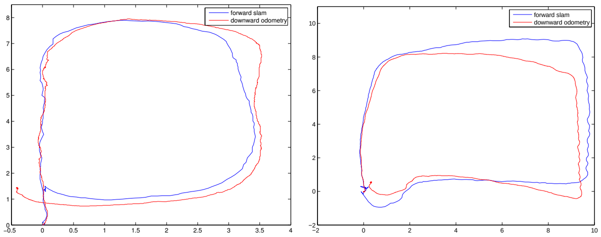


(a) Forward image sample (b) Downward image sample

(c) Octomap

Figure 7: The outdoor environment and octomap result

We compare the position outputs of two cameras excluding height. The result is shown in Fig. 8. The total length of the indoor camera movement is around 22m. The velocity output of downward facing camera is integrated as position. Through multiple tests, We observe that the total position error (the position difference of starting point on the loop) of ORB-SLAM is less than 20cm in average. The position error of downward facing camera accumulates and is a bit larger, but the velocity estimation is enough to stabilize the MAV. The outdoor test shows a similar result. We observe that Realsense camera can provide a better depth map in the good light condition of outdoor texture-rich environment. A fix exposure time performs better than auto exposure when the light condition varies in the canopy.



(a) Indoor result (b) Ourdoor result

Figure 8: The position output

7 Conclusions

In this paper, we present a localization and mapping system with dual Realsense cameras. The downward facing camera provides velocity estimation for stabilization of the

MAV. At the same time, the forward facing camera provides less-drift position estimation for navigation. We demonstrate the effectiveness of our system in both indoor and outdoor environment. In future work, we will test the system on our quadrotor platform and perform vision based autonomous flight.

REFERENCES

[1] Stefan Kohlbrecher, Oskar Von Stryk, Johannes Meyer, and Uwe Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 155–160. IEEE, 2011.

[2] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems Conference (RSS)*, 2014.

[3] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[4] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustmentła modern synthesis. In *Vision algorithms: theory and practice*, pages 298–372. Springer, 1999.

[5] Felix Endres, Jurgen Hess, Jurgen Sturm, Daniel Cremers, and Wolfram Burgard. 3-d mapping with an rgb-d camera. *Robotics, IEEE Transactions on*, 30(1):177–187, 2014.

[6] Raúl Mur-Artal and Juan D Tardós. Fast relocalisation and loop closing in keyframe-based slam. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 846–853. IEEE, 2014.

[7] Raúl Mur-Artal, JMM Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *Robotics, IEEE Transactions on*, 31(5):1147–1163, 2015.

[8] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *Robotics, IEEE Transactions on*, 28(5):1188–1197, 2012.

[9] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[10] David Caruso, Jakob Engel, and Daniel Cremers. Large-scale direct slam for omnidirectional cameras. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 141–148. IEEE, 2015.

[11] Jakob Engel, Jörg Stückler, and Daniel Cremers. Large-scale direct slam with stereo cameras. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1935–1942. IEEE, 2015.

[12] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV)*, 2011.

[13] Albert S Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *International Symposium on Robotics Research (ISRR)*, pages 1–16, 2011.

[14] Adam Bry, Charles Richter, Abraham Bachrach, and Nicholas Roy. Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments. *The International Journal of Robotics Research*, page 0278364914558129, 2015.

[15] 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.

[16] Friedrich Fraundorfer, Lionel Heng, Dominik Honegger, Gim Hee Lee, Lorenz Meier, Petri Tanskanen, and Marc Pollefeys. Vision-based autonomous mapping and exploration using a quadrotor mav. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4557–4564. IEEE, 2012.

[17] Abraham Bachrach, Samuel Prentice, Ruijie He, Peter Henry, Albert S Huang, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Estimation, planning, and mapping for autonomous flight using an rgb-d camera in gps-denied environments. *The International Journal of Robotics Research*, 31(11):1320–1343, 2012.

[18] Korbinian Schmid, Philipp Lutz, Teodor Tomić, Elmar Mair, and Heiko Hirschmüller. Autonomous vision-based micro air vehicle for indoor and outdoor navigation. *Journal of Field Robotics*, 31(4):537–570, 2014.

[19] Ji Zhang, Michael Kaess, and Sanjiv Singh. A real-time method for depth enhanced visual odometry. *Autonomous Robots*, pages 1–13, 2015.