<http://www.diva-portal.org>

Preprint

This is the submitted version of a paper published in *Lecture Notes in Computer Science*.

Citation for the original published paper (version of record):

Fan, Y., Aramrattana, M., Shahbandi, S G., Nemati, H M., Åstrand, B. (2016)

Infrastructure Mapping in Well-Structured Environments Using MAV.

Lecture Notes in Computer Science, 9716: 116-126

http://dx.doi.org/10.1007/978-3-319-40379-3_12

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-31645>

Infrastructure Mapping in Well-Structured Environments Using MAV

Yuantao Fan, Maytheewat Aramrattana, Saeed Gholami Shahbandi
Hassan M. Nemati, and Björn Åstrand

School of Information Technology
Halmstad University, Halmstad, Sweden

{yuantao.fan, maytheewat.aramrattana, saeed.gholami_shahbandi,
hassan.nemati, bjorn.astrand}@hh.se

Abstract. In this paper, we present a design of a surveying system for warehouse environment using low cost quadcopter. The system focus on mapping the infrastructure of surveyed environment. As a unique and essential parts of the warehouse, pillars from storing shelves are chosen as landmark objects for representing the environment. The map are generated based on fusing the outputs of two different methods, point cloud of corner features from *Parallel Tracking and Mapping* (PTAM) algorithm with estimated pillar position from a multi-stage image analysis method. Localization of the drone relies on PTAM algorithm. The system is implemented in Robot Operating System(ROS) and MATLAB, and has been successfully tested in real-world experiments. The result map after scaling has a metric error less than 20 *cm*.

1 Introduction

To achieve an industrial automatic warehouse environment, where all the autonomous vehicles can localize themselves, perform task planning and interact with each other, a map that represents the infrastructure and goods in the environment is required. One of the approaches to achieve such warehouse environment is to create the map automatically using unmanned vehicle. To explore an unknown environment, simultaneous localization and mapping (SLAM) technique is often used. It is the joint estimation of the unmanned vehicle's position in the environment and a model of its surroundings i.e., the map. It is the key requirement for any truly autonomous mobile vehicle to operate in an unknown workspace. A vast number of SLAM implementations have been proposed for aerial or ground vehicles. In this paper, a quadrotor helicopter was used to create the map of a warehouse.

1.1 Related works

Compared to ground vehicles, the micro aerial vehicles (MAVs) have advantages in form of mobility. For instance, they can operate in three-dimensional space and do not require the ground to run on. While MAVs enable opportunities for

performing new tasks, they contain more challenges compared to the ground robots. Especially, weight and power constraints, as discussed in [16]. Many studies have been performed on SLAM using MAVs in indoor [7, 17, 21, 23] and outdoor [18, 3] environments. In these works different type of sensors such as camera [10, 14], and lightweight laser scanner are commonly used for mapping and tracking purposes. In contrast to laser rangefinders, cameras are affordable, small, and light. Furthermore, they can be used in both indoor and outdoor environments with less limitations. These characteristics make cameras more suitable choice for aerial vehicles.

Parallel Tracking and Mapping (PTAM) [15] is one of the popular keyframe-based algorithms. By using image streams from a camera, one can construct point cloud of corner features in the environment. PTAM can be used for tracking-while-mapping purposes [3, 11]. By using PTAM, no prior model of the scene is required, and the algorithm provides a 3D map of corner features in the observed image frames.

Majority of the performed researches were using MAV platform from Ascending Technologies GmbH (AscTec). This platform was used mostly to perform autonomous exploration and navigation tasks such as the work from *Shen et al.* [22, 23], *Weiss et al.* [24], *Bachrach et al.* [4, 6, 5] and *Pravitra et al.* [21]. The platform offers high performance on-board processors with around 600-650 grams payload. Another alternative is the platform from MikroKopters [2], which as an example is used in *Piratla et al.*'s work [19]. Finally, *Parrot AR.Drone 2.0*, is a quadcopter platform with cheaper price. It does not offer any payload. Only a few research have been done using *AR.Drone*. For example, autonomous indoor flight from *Bills et al.* [7], vision based navigation from *Blosch et al.* [8] and *Engel et al.* [12]. The work from *Chaves et al.* [9] proposed real-time 6-DOF SLAM using *AR.Drone* but with pre-mounted special tags as known landmarks. The interesting feature of the *AR.Drone* is the cost; it doesn't have a payload, which makes it more suitable for exploration mission and human robot interaction (HRI) [20].

In this work we employ a cheap and easy to use platform, *AR.Drone*, to explore and map the infrastructure in a real warehouse environment. The system provides a map, representing the structure of the surveyed environment. Our experiments are done in a warehouse where pillars from storing shelves are desired as landmark objects. In order to detect pillars, we combine the information from PTAM with a multi-stage image analysis algorithm which uses the prior knowledge about the unique and uniform color of the pillars. This information fusion provides us a robust and accurate position estimation of pillars location in the warehouse. The system is implemented in Robot Operating System (ROS) and MATLAB, and has been successfully tested in a real-world experiments. The map generated after scaling has a metric error less than 20 cm.

The remaining of this paper is structured as follows. In Section 3.1, our system architecture and modifications are elaborated. Section 2 explain the proposed method, and we describe our experiments and results in section 3. We summarize our contribution and discuss future work in Section 4.

2 Method

Fig. 1: System overview

System overview is elaborated in Fig. 1. Based on image stream from *AR.Drone*, Parallel Tracking and Mapping (PTAM) [15] provides position of the drone and point cloud of corner features in the environment. Localization of drone by means of the *PTAM* and required modifications are described in section 2.1. With the same video stream, image analysis techniques are applied to detect pillars and project them into 2D world coordinates. In section 2.2, the process of pillar detection and corresponding assumptions are presented. Finally, the map of pillars in world coordinates is created using the correspondence between PTAM's point cloud and 2D projection of pillars (described in section 2.3).

2.1 Localization and point cloud map

Localization of the drone is carried out by employing an implementation of PTAM, which is a vision based tracking system designed for augmented reality.

It provides position of the drone in the environment as well as point cloud of corner features. The package employs Extended Kalman Filter, of which the control gains are used for EKF prediction, location of extracted corner features and navigation data are used for EKF update. After the initialization of PTAM, a map $\mathbf{m} = \{m_1, m_2, \dots, m_k\}$ that contains the location of corner features in the image is added or updated.

2.2 Landmark detection

Pillars from shelves are the best representative objects for a warehouse environment, which capture the structural layout of the warehouse. An example is shown in Fig. 2. As a common infrastructure of the warehouse, they are painted with unique and uniformed color, following a certain installation pattern i.e. structural layout. In this work, a multi-stage image analysis and a grouping algorithm is employed to obtain pillar's position $\mathbf{A} = \{\lambda_1, \lambda_2, \dots, \lambda_l\}$ from image sequence and drone's odometry [13].

Fig. 2: (left) A scene in the warehouse, where pillars are common infrastructure with uniformed and unique colour. (right) Structural layout of the warehouse: pillar can be considered as landmark of the environment.

Initially the image acquired from *AR.Drone* is rotated to align the vertical axis of the image with the corridor. This rotation will facilitate correlating the image coordinate and the point cloud data (provided by PTAM). The first stage of the image analysis algorithm is color segmentation. This task is performed in HSV color space. Pixels belonging to pillars are extracted by thresholding the HSV component. Afterwards, the edges of pillars are detected using *canny edge detector*. At last, Hough transform is employed to extract lines that represent the edges of the pillars. Since the correlation between image coordinate and the point cloud is provided in an open loop, only horizontal edges are accepted to improve the accuracy of the result. The pillar positions $\mathbf{A}$ in the global frame are simultaneously published into ROS topic and stored. After all the image

Fig. 3: (left) shows the view of bottom camera. Red lines in (right) are detected edges of pillars.

sequences are processed, the estimated pillar positions will be grouped into several clusters, each cluster representing a pillar. Then, weighted average mean is applied on all pillar projections and the mean position will be considered as the estimated pillar position of each cluster.

2.3 Mapping pillars using data fusion

Creation of pillar map relies on two sources, point cloud $\{m_1, m_2, \dots, m_k\}$ from PTAM and estimated pillar position $\{\lambda_1, \lambda_2, \dots, \lambda_l\}$ from pillar extraction method. The mapping algorithm [13] fuses the two sources and generates a 2D map of pillar's position $\hat{\lambda} = \{\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_l\}$. Point cloud map from PTAM includes 3D position of corner features in image frames, some of the corners extracted from the pillars (since there are structural rectangle hole patterns on every pillar) and therefore indicate pillar's 2D position $(x_{\lambda_t}, y_{\lambda_t})$. Estimated pillar position from pillar detection could be utilized to locate these points and filter out the irrelevant points (extracted from other object). To find the dominant orientation of the point cloud. Radon transform is employed. Through this operation, the point cloud data and estimated pillar position are correlated in the same coordinates. At the end, a filter (equation 1) is applied to accept the point cloud that is close to the estimated pillar position.

$$\hat{\lambda}_n = f(\mathbf{m}, \lambda_n, W_{size}) \quad (1)$$

$\mathbf{m}$ is the position of all corner features, λ_n is the estimated 2D position of a pillar and W_{size} is the size of a window function. Any corner point that is located within the window function of a pillar is selected as a candidate of the correspondent pillar. An average weight is applied on each candidate and mean value of these points along x and y axis is calculated. Algorithm 1 describe the details of the pillar localization and mapping method. Figure 4a shows the point cloud that aligned to the corridor and figure 4b shows the point cloud that is belong to the pillars filtered by the algorithm 1, the red points are considered as the estimated 2D position of the pillar $\hat{\lambda}$.

```

Data: pointcloud, pillar_pos
Result: pillar_filtered_pos
angle = pointcloud.radontransform();
pointcloud = pointcloud.rotate(angle);
for All point in pointcloud do
    | pillar_cloud = Filter(point, pillar_pos);
end
for All pillar_cloud_cluster in pillar_cloud do
    | pillar_filtered_pos[i] = pillar_cloud_cluster[i].average();
end

Function Filter(point, pillar_pos)
for All pos in pillar_pos do
    | if point.inrange(pos, filter_range) then
        | | pillar_cloud.add(point);
        | | pillar_cloud_cluster[i].add(point);
        | | break;
    | end
end
return pillar_cloud;

```

Algorithm 1: Procedure of mapping algorithm

Fig. 4: 4a shows point cloud from PTAM, 4b shows filtered point cloud and the mapping result is shown in 4c

3 Experiments and results

3.1 System Architecture and Modifications

Overview of our system is shown in figure 1. It contains two major parts. First, *Parrot AR.Drone 2.0*, a quadrotor helicopter platform that carries two cameras, one looking forward (front camera) and the other facing downward (bottom camera). Second, a computer, which gather data from the *AR.Drone* into robot operating system (ROS). The computer and the *AR.Drone* is connected with a wireless connection. The data consists of *a*) image stream from a selected camera and *b*) position of the *AR.Drone* estimated by inertial measurement unit (IMU). ROS use the data to run PTAM algorithm, which provide three-dimensional coordinates of corner features detected in the environment, referred to as point cloud in Fig. 1. Moreover, pillars are detected from images using color segmentation. “2D Pillar projection” in Fig. 1 is the estimated positions of the pillars in two-dimensional space using this pillar detection method. Later, MATLAB was used to combined the data and generate a map of pillars of the environment. For more details regarding methods, please refer to Section 2.

One major modification to the original *AR.Drone* was made, which is to move the front camera to the bottom of the *AR.Drone* looking downward. The main reason for this modification is because of the PTAM algorithm. PTAM’s performance depends on tracking corner features and rapid changes of the scenes cause the algorithm to fail (e.g. a sharp yaw turn by *AR.Drone*). Moreover, PTAM expect the motion of the camera to be in parallel to the scene. In front camera set up, the scene changes topologically as the drone flies into the corridor, hence the features’ motion is not in parallel to the camera. Figure 5 illustrates the view comparison between front and down camera configurations. However, the bottom camera has a considerably small field of view and low resolution which doesn’t provide enough feature for the algorithm to track. Therefore, to satisfied the requirements, the front camera set up on the *AR.Drone* was modified. This camera setup was inspired by *ethzasl-ptam* package[1], where with PTAM is employed with a similar camera setup mounted on a high altitude aerial vehicle.

3.2 Results

The experiment was conducted in a warehouse. The AR.Drone flew straight through a corridor in a warehouse, start from $(0, 0)$ going in $+y$ direction. It was manually controlled using joystick. Following data are collected from the *AR.Drone* through ROS: *a*) image stream; *b*) IMUs data; and *c*) control command that sent to the *AR.Drone*. Then, the recorded data was used in localization, pillar detection method and generate point cloud map. Finally, outputs from the methods were used to generate an object map in MATLAB. The map is illustrated in figure 4c. Pillars are represented with (x, y) value in world coordinates. Corridor width is calculated based on taking the distance between two peaks of point cloud projection at the dominant orientation of Radon transform,

Fig. 5: Illustration of scenes from original and modified camera

which is 2.8 meters for this dataset. Distances between adjacent pillars is calculated in x and y direction. After that, mean and variance of distances, shown in table 1, are calculated for further evaluation.

In order to evaluate the result, the map is scaled up to the actual distance in the environment. In this case, we scale the map based on three criteria:

a) pillars on left side; *b)* pillars on right side; and *c)* width of the corridor.

Table 2 presents errors compared to real distances after scaling.

Pillar	D_x (meters)		D_y (meters)	
	Mean	Variance	Mean	Variance
Left Side	0.0044	0.0071	2.7554	0.8973
Right Side	-0.0091	0.0307	2.6766	0.1767

Table 1: Mean and variance of distance between pillars in meters

Scaling Base	Error (%)		
	Left	Right	Width
Left Pillar Distance	0	-2.8592	-6.9931
Right Pillar Distance	2.9397	0	-4.2550
Corridor Width	7.5189	4.4469	0

Table 2: Estimation error after scaling

4 Conclusion

Towards the goal of industrial automation in warehouses, a surveying system based on a low cost platform, *Parrot AR.Drone 2.0*, is proposed in this paper. Based on image stream and IMUs, the drone employs *Parallel Tracking and Mapping* to localize itself in the environment. Moreover, PTAM generates point cloud of corner features in the environment, which is used in pillar localization and mapping. Concurrently, our pillar detection method detects and provide 2D pillar projections. Finally, a pillar map is created by finding the correspondence between the point cloud and pillar projections.

As a result, the system provides a pillar map with the biggest error of 20.3 centimeters(7.5189%). Cost of the system is comparatively low and doesn't need any prior setup on the environment.

Discussion

One problematic issue of the map generated by PTAM is the scale, which greatly depends on manual initialization of the first stereo pair, the first two key frames. Therefore the map generated from two different trails can be inconsistent.

In addition, image sequences from *AR.Drone* are not always smooth and sometimes do not have good quality for tracking. As a result, the algorithm eventually lose tracking and fail the localization. Later, image sequences were improved to be more smooth by reducing the control speed of *AR.Drone*.

Pillar detection method is based on the color of pillars, which is known and pre-defined before the experiment. Our test environment has uniform color of pillars, which might be different in some other environment. Thus, a more general approach is desired.

Prospective

Long-term autonomy of intelligent vehicles within warehouse environment requires autonomous exploration, obstacle avoidance and self-charging functionality. The proposed work can be integrated into autonomous warehouse for surveillance purpose, e.g. MAVs can be deployed to modelling traffics in the warehouse, provide useful information for planning the path of ground vehicles as well as detect anomalies in the warehouse.

References

- [1] Ethzasl ptam (25 september 2013), http://ros.org/wiki/ethzasl_ptam
- [2] Mikrokopters (25 september 2013), <http://www.mikrokopter.de/en/home>
- [3] Achtelik, M., Achtelik, M., Weiss, S., Siegwart, R.: Onboard imu and monocular vision based control for mavs in unknown in-and outdoor environments. In: Robotics and automation (ICRA), 2011 IEEE international conference on. pp. 3056–3063. IEEE (2011)

- [4] Bachrach, A., He, R., Roy, N.: Autonomous flight in unknown indoor environments. *International Journal of Micro Air Vehicles* 1(4), 217–228 (2009), <http://multi-science.metapress.com/index/80586KML376K2711.pdf>
- [5] Bachrach, A., Prentice, S., He, R., Henry, P., Huang, A.S., Krainin, M., Maturana, D., Fox, D., Roy, N.: 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), <http://ijr.sagepub.com/content/31/11/1320.short>
- [6] Bachrach, A., Prentice, S., He, R., Roy, N.: Range-robust autonomous navigation in gps-denied environments. *Journal of Field Robotics* 28(5), 644–666 (2011), <http://onlinelibrary.wiley.com/doi/10.1002/rob.20400/full>
- [7] Bills, C., Chen, J., Saxena, A.: Autonomous mav flight in indoor environments using single image perspective cues. In: *Robotics and automation (ICRA), 2011 IEEE international conference on*. pp. 5776–5783. IEEE (2011), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5980136
- [8] Bloesch, M., Weiss, S., Scaramuzza, D., Siegwart, R.: Vision based mav navigation in unknown and unstructured environments. In: *Robotics and automation (ICRA), 2010 IEEE international conference on*. pp. 21–28. IEEE (2010), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5509920
- [9] Chaves, S., Cohen, S., O’Keefe, P., Ozog, P.: Real-time 6-dof slam for a quadrotor helicopter
- [10] Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29(6), 1052–1067 (2007)
- [11] Engel, J., Sturm, J., Cremers, D.: Camera-based navigation of a low-cost quadcopter. In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. pp. 2815–2821. IEEE (2012)
- [12] Engel, J., Sturm, J., Cremers, D.: Camera-based navigation of a low-cost quadcopter. In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. pp. 2815–2821. IEEE (2012), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6385458
- [13] Fan, Y., Aramrattana, M.: *Exploration and mapping of warehouses using quadrotorhelicopters* (2013)
- [14] Kerl, C., Sturm, J., Cremers, D.: Dense visual slam for rgb-d cameras. In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. pp. 2100–2106. IEEE (2013)
- [15] Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*. pp. 225–234. IEEE (2007), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4538852
- [16] Kumar, V., Michael, N.: Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research* 31(11), 1279–1291 (2012), <http://ijr.sagepub.com/content/31/11/1279.short>
- [17] Morris, W., Dryanovski, I., Xiao, J.: 3d indoor mapping for micro-uavs using hybrid range finders and multi-volume occupancy grids. In: *RSS 2010 workshop on RGB-D: Advanced Reasoning with Depth Cameras, Zaragoza, Spain* (2010)
- [18] Newman, P., Cole, D., Ho, K.: Outdoor slam using visual appearance and laser ranging. In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. pp. 1180–1187. IEEE (2006), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1641869

- [19] Piratla, V., Malode, S.B., Saini, S.K., Jakhota, A., Sao, A.K., Rajpurohit, B.S., Haobijam, G.: Autonomous navigation in gps denied indoor environment using rgb-d sensor, kinect
- [20] Pourmehr, S., Monajjemi, V.M., Vaughan, R., Mori, G.: you two! take off!: Creating, modifying and commanding groups of robots using face engagement and indirect speech in voice commands. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. pp. 137–142. IEEE (2013)
- [21] Pravitra, C., Chowdhary, G., Johnson, E.: A compact exploration strategy for indoor flight vehicles. In: Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on. pp. 3572–3577. IEEE (2011), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6161200
- [22] Shen, S., Michael, N., Kumar, V.: Autonomous multi-floor indoor navigation with a computationally constrained mav. In: Robotics and automation (ICRA), 2011 IEEE international conference on. pp. 20–25. IEEE (2011), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5980357
- [23] Shen, S., Michael, N., Kumar, V.: Autonomous indoor 3d exploration with a micro-aerial vehicle. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on. pp. 9–15. IEEE (2012), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6225146
- [24] Weiss, S., Achtelik, M., Kneip, L., Scaramuzza, D., Siegwart, R.: Intuitive 3d maps for mav terrain exploration and obstacle avoidance. *Journal of Intelligent & Robotic Systems* 61(1-4), 473–493 (2011), <http://link.springer.com/article/10.1007/s10846-010-9491-y>