2017 International Conference on
Unmanned Aircraft Systems (ICUAS)
June 13-16, 2017, Miami, FL, USA

Real-time graph-based SLAM in unknown environments using a small UAV

Arun Annaiyan[1], Miguel A. Olivares-Mendez[1], Holger Voos[1]

***Abstract*— Autonomous navigation of small Unmanned Aerial Vehicles (UAVs) in cluttered environments is still a challenging problem. In this work, we present an approach based on graph slam and loop closure detection for online mapping of unknown outdoor environments using a small UAV. Here, we used an onboard front facing stereo camera as the primary sensor. The data extracted by the cameras are used by the graph-based slam algorithm to estimate the position and create the graph-nodes and construct the map. To avoid multiple detections of one object as different objects and to identify re-visited locations, a loop closure detection is applied with optimization algorithm using the g2o toolbox to minimize the error. Furthermore, 3D occupancy map is used to represent the environment. This technique is used to save memory and computational time for the online processing. Real experiments are conducted in outdoor cluttered and open field environments.The experiment results show that our presented approach works under real time constraints, with an average time to process the nodes of the 3D map is 17.79ms**

I. Introduction

In last decades the use of autonomous small Unmanned Aerial Vehicles (UAVs) has been increased rapidly in military and civil applications. This was mainly lead by the advances in the reduction of the electronics, increment of the computational power capabilities and the compact size and high maneuverability of this type of UAVs. Recent applications include package delivery, traffic surveillance, search and rescue, target tracking, aerial photography, 3D modeling. For instance, during package delivery, the UAV has to navigate in unknown environments at low altitudes. When a UAV flies at low altitude in urban scenarios, GPS accuracy is not enough to localize precisely the UAV. As shown in Figure 1, when the UAV is flying at low altitude, trees and buildings can partially block or reflect the GPS signal of different satellites. In this case either we have to use an accurately predefined map or we need to generate the precise online map. Even if we use predefined map it is obvious that new objects could be added over the time, especially in urban environments. A solution for this type of environments is the online 3D map generation. With the help of real time 3D maps, it is possible to retrieve the obstacle position, shape, size, distance from the sensor and to generate the shortest safe path to reach the desired destination.

Online map generation is still consider as challenging in terms of computation and real time constraints, and even more when the data is processed onboard. Since this work concentrates on the unknown environment in outdoor scenario, we focus on 3D online map generation problem. In this paper, we address the issue of the online mapping of the outdoor environment by combining the graph based slam and loop closure detection algorithm. All the data extracted from the environment have been processed online and onboard, and computational capability is proven through the real time experiments. To the best of our knowledge it is the first approach in the literature combining graph slam, loop closure detection algorithm using the g2o optimization method applied to UAVs using a stereo camera.

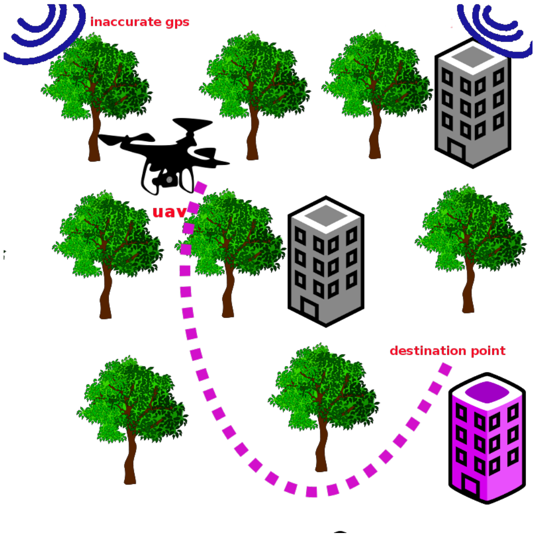

Fig. 1. illustration of the outdoor scenario

The paper is organized as follows, section II shows the current state of the art. Section III explains the methods of graph slam and global loop closure detection, IV describes the experiments and its setup, Section V presents the discussion from the results obtained and future work.

II. Related Work

The online map construction in mobile robotics classically implemented by using Simultaneous Localization And Mapping (SLAM) [6]. The SLAM algorithm depends on a perception system to extract the information from the environment. It is especially important in the scenarios where the GPS information is partially not available or not precise enough to process it. Sensors like RGB-D cameras [15],

*This work was supported by FNR Fonds national de la Recherche (Luxembourg) through AFR Aides a la Formation-Recherche Ph.D. grant scheme No. 10156266

[1]All the authors are associated with Automation and Robotics research group, Interdisciplinary Centre for Security, Reliability and Trust (SnT),University of Luxembourg. Corresponding author - arun.annaiyan@uni.lu

978-1-5090-4494-8/17/$31.00 ©2017 IEEE

Laser [18], monocular camera [21], and stereo cameras [3] are used mobile robotics applications. The selection of the appropriate sensors suit depends on many factors, such as the robot's payload capability, computational power, light variations, size of the environment, color intensities, or the textures richness of the outdoor scenario. Based on the reduced weight, price and high efficiency and scenario characteristics, which will be shown in section IV we choose a stereo camera for this work, as shown in Figure 2. Hence in this paper, we focus on stereo vision based online mapping of outdoor environments.

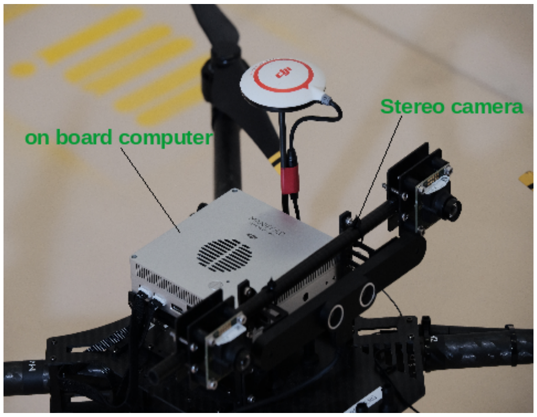

Fig. 2. setup of stereo camera and onboard computer

Vision based mapping is an active research topic in the recent years [10] [25] [17] and its already proven in aerial robotics [9]. For real time 3D environment exploration the trade off is in between the mapping and localization methods. Here using the visual data, the construction of the map and calculation of a precise trajectory inside the map can be done by using visual slam [4] [10]. Slam problem is usually defined in probabilistic form. This comprises two fundamental slam techniques called *online slam* and *full slam*.

Online slam is represented as $p(x_t, m|z_{1:t}, U_{1:t})$, where x_t, $z_{1:t}$ and $U_{1:t}$ denote the pose, measurements and control, respectively. It calculates the pose at the time $'t'$ and it does not take into account the past measurements. On the other hand, the full slam takes the past and present measurements to calculate the pose as shown in equation (1).

$$p(x_{1:t}, m|z_{1:t}, U_{1:t}) \tag{1}$$

State of the art includes Extended Kalman Filter SLAM (EKF-SLAM), particle filter, graphical model based approaches. In this work [8] it is shown in the result that the classical EKF-SLAM is inconsistent. Even with the further improvements presented in [7], this approach presents a lack of accuracy when it is used in large scale mapping scenarios [14]. The uses of the particle filter technique [23] is well known in landmark-based and grid representations [26]. According to [24], particle filter has some basic limitations due to its bounded number of particles required to estimate the pose and trajectory. This number can increase enormously depending on the noise of the sensor. Recent developments in the SLAM community based on graph-SLAM provides accuracy and computational efficiency. This method calculates the pose of the system and store it in the nodes. Each node is connected with the one which has the geometrical transformation. Sensor measurements are used as constraints in between them. These connections between are called edges. In this work [20] the system of equations are analyzed from the graph-SLAM to reduce the noise coming out of the raw measurements by a global optimization. They follow the Gauss-Newton optimization approach which is critical for long-term online mapping. This is then improved by conjugate gradient optimization technique [27]. Later in [11] it is proposed a method that selects the specific variables needed for efficient optimization operation. In real time experiments, it is essential to keep the memory under real time constraints. To save memory loop closure detection method is useful for identifying the previously detected location using differential reference [12]. It saves processing time once it is marked as previous visited location. By combining the graph-SLAM and global loop closure detection we can get an efficient and accurate map online. However the optimization techniques used can impede the process due to the online constraints which is decided by the size of the environment. By applying the graph slam and global loop closure detection with open source g2o optimization technique [16] to minimize the error, we showed the efficient online mapping from the small UAV. For representation of 3D model of the map we used the open source library octomap [13] which is proven to be memory efficient to get the known, unknown and free spaces inside the map.

Here in this paper, we focuses on the real time experiments for outdoor large scale as well as cluttered environment. While doing the long term operations we demonstrated the computational time processing of the onboard computer.

III. Methodology

In the present work the forward looking stereo camera is used as main sensor and the sift algorithm[19] is applied to detect and extract features from the environment. Figure 3 shows the stereo coordinate system.

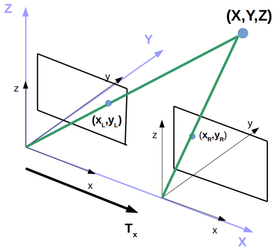

Fig. 3. geometry of the stereo camera setup

Equation 2 shows the calibration matrix,

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_c = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_w \qquad (2)$$

where, (X,Y,Z) are the 3D point in the world coordinate system. f_x , f_y are the focal lengths. Next to that c_x , c_y are the principal point in the center of the image and u , v are points in the camera coordinate system. Here, the disparity map d is calculated by,

$$d = X_L - X_R \qquad (3)$$

The 3D point from each pair of images are calculated using equation 4,

$$Z = fB/d \qquad (4)$$

where B, d, f are the baseline between two cameras, disparity and focal length respectively. More details found in [5]

The calculation of the pose and map have been done using graph-SLAM with global loop closure detection. The graph-SLAM is independent of size and time and it stores the pose information in nodes at each frame acquired. Here, the time t is discrete. The UAV pose associated to the stereo camera system and the individual sensor measurements are represented by x_t and z_t respectively. The UAV pose and measurement at time t is represented by x_t and z_t respectively. The individual sensor measurement for the image is denoted by z_t.

A. Graph-SLAM

In the graph-SLAM the pose of the UAV is stored in the form of nodes which are connected by edges, according to the increment of the measurements as shown in Figure 4. In this Figure $x_1, x_2, x_3, ...x_t$ represents the nodes. Each node stores the absolute pose with respect to the first node, which are calculated from the stereo images.

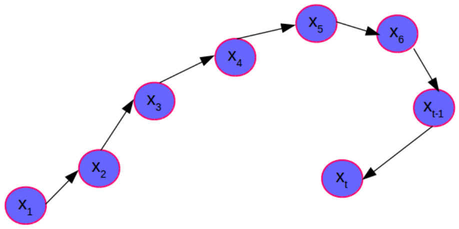

Fig. 4. Node representation in graph slam

The estimation of the pose to store in the i^{th} node at time t is based on the raw sensor data z_t^i. This data is processed and generates a new i^{th} node based on the detection of the features present in that image captured at time t. This information can contain some noise that can generate an error in the pose estimation between consecutive nodes. This error is defined by,

$$e_{i,j}(x) = f_{i,j}(x) - c_{i,j} \qquad (5)$$

where, x_i and x_j denotes the absolute pose of i^{th} and j^{th} consecutive nodes respectively. $f_{i,j}(x)$ is the function that measures the transformation from node i to node j and $c_{i,j}$ denotes the constraints of the odometry displacement for the mentioned nodes.

Here the cost function $F_{i,j}(x)$ is defined as the transformation from the i^{th} to the j^{th} node as it is shown in equation 6 where $\Omega_{i,j}$ is the information matrix contains the measurement of the respective nodes.

$$F_{i,j} = e_{i,j}^T \Omega_{i,j} e_{i,j} \qquad (6)$$

Equation 7 is to find the optimized value of x^*

$$x^* = \arg\min_x F(x) \qquad (7)$$

B. Global loop closure detection

The global loop closure detection is and algorithm used to find and match data from previously visited locations. For example, if the UAV reach a location that is already stored, then the mentioned algorithm will recognize the location and identify the represented node. This algorithm also used to match the detection done in previous frame with the following one. In that way an object which is detected in various frames is detected always as one single object and as new object in each frame. Besides it also have been proven for memory efficient [1] in large size scenarios. This loop closure detection is determined using the bag of words method [28].

In general, the bag of words approach is a method of image classification. With this method we index the node information based on the image information. When new image information is processed, it is compared with the stored data using bayesian filters. Equation 8 shows the mentioned filter.

$$p(S_t|I^t) = \eta p(I^t|S_t)p(S_t|I^{t-1}) \qquad (8)$$

Where $I^t = I_0, I_1, ...I_t$, I_t is the current image, S_t is the random image information obtained from the bag of words database, which conform the loop closure hypothesis. Figure 5 illustrate the loop closure procedure based on the probability event .

Equation 9 shows the previous equation rewritten in terms of measurements z_t.

$$p(S_t|z^t) = \eta p(z_t|S_t)p(S_t|z^{t-1}) \qquad (9)$$

Here I^t is replaced by the sequence of sensor measurement represented as z^t

The loop closure detection can be erroneous if the same features of the images are perceived as different. This will have influence in the nodes containing the poses related to these images and its edges. To optimize this problem we used the open source c++ framework called *g2o* [16].

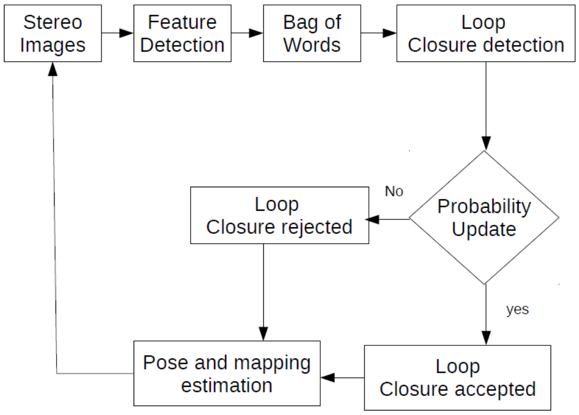

Fig. 5. Methodology

IV. Experiments

To validate the implemented approach we conducted a series of experiments in outdoor environments with a real quadrotor. The areas for the two experiments were selected based on the characteristics of them. First experiment scenario was set on a low density forest (cluttered environment). The purpose of this specific scenario is to check how the implemented approach performs in an environment where objects are very similar and features from different trees can be wrongly matched to one single tree or trees detected in the past. The second scenario was, on the other hand, set on an open field with one single obstacle. This one was selected to evaluate the approach in a scenario with a reduced number of features.

A. Experiment setup

The quadrotor used in this work is the DJI matrice 100 [2] shown in figure 6. All the processing and computations are done in the onboard computer (arm processor) [22]. The onboard computer has installed ubuntu 14.04 and the Indigo version of the Robotic Operating System (ROS). The customized software was installed in this processor.

We use a stereo camera system with two ueye cameras separated by a baseline of 19cm as shown in figure 6. The maximum resolution of the camera is 752 x 480 pixels, however to reduce the computational cost and to increase the number of frames per seconds we reduce the resolution by half pf the maximum resolution i.e 376 x 240 at 30 frames per second. It has global shutter and the sensor model is named as MT9V032C12STC. The stereo camera have been calibrated previously. The main advantage of this setup is that it is easy to build, lightweight and the distance between the camera can be adjusted. This stereo camera is not hardware triggered. The delay between images (left and right) are approximated every 15 frames. This simple setup is mounted on the front of quadrotor platform as shown in figure 6.

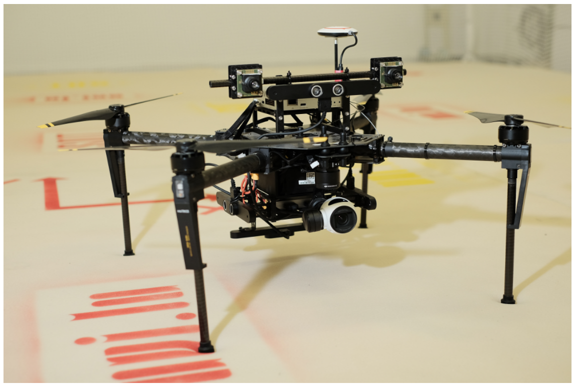

Fig. 6. UAV mounted with the stereo camera

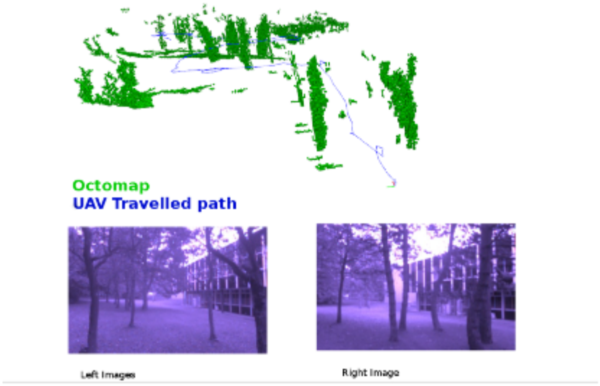

Fig. 7. Octomap and UAV trajectory

B. Case 1 : Cluttered environment

To test our approach, we choose a cluttered environment with varying illumination condition and also with similar features to check the robustness of the loop closure algorithm. This environment consists of trees located in narrow space as shown in figure10. The ground surface is not even and the illumination condition change when the unstructured trees creating sparse shadow. We fly manually the UAV inside the unstructured, unknown environment at an altitude of 2 meters. The algorithm started to compute the initial odometery after 6.278 milliseconds. Even though, the high similarity between the tress present in the environment,the algorithm predicted the keypoint features and process the odometry correctly. Figure 11 shows the keypoint features in yellow and the blue color represents the odometry.

The odometry data started to generate at 1.5 m from the staring point. The average and maximum time taken to process the node indexes were 9.691 ms and 28.43 ms as shown in figure 15. In figure 7it is shown the octomap map in green color and the trajectory of the UAV in blue color. The odometry of the UAV is shown in figure 8. The GPS information along with top view of the environment is shown in figure 9.

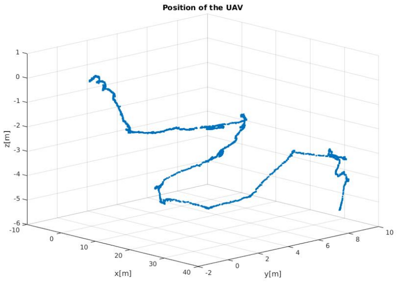

Fig. 8. Position and Orientation of the UAV (odometry)

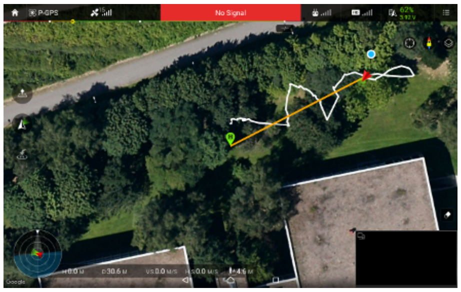

Fig. 9. Top view and gps information of the environment

Fig. 10. Unstructured and illumination condition

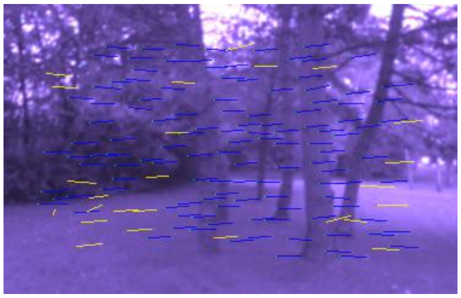

Fig. 11. Odometry and keypoint features

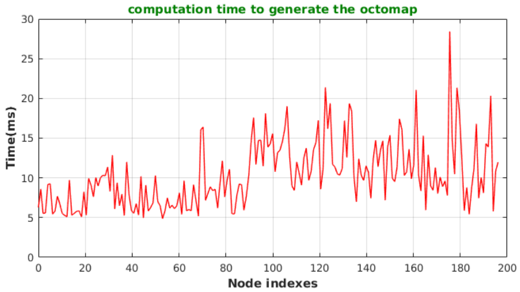

Fig. 12. Total time response to process all the nodes

C. Case 2 : Open Field

In the second experiment, we fly manually the UAV in wide outdoor space with a single large structure. There are no other objects and features except the grass in the ground. The computational cost of the odometry algorithm during this test is shown in figure 15, with and average value of 17.789 ms and a maximum time of 38.37 ms. Here the complexity occurs, when no obstacles or objects were inside the image frame reducing the number of features to almost null. To increase the number of features, we keep the altitude fixed on 1.5m allowing the system to take ground features to generate the map. It have to be taken into account that the features extracted from the grass (features) was affected by the airflow created by the quadrotor.However, the system was able to generate the map and localize the UAV successfully. Totally 78 nodes were processed in this experiment. In figure 13 it is shown the generated map and trajectory of the UAV in green and blue color respectively.

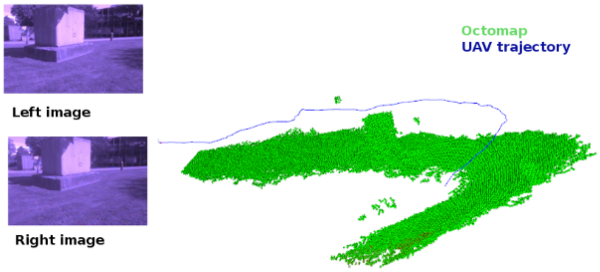

Fig. 13. Octomap and UAV trajectory

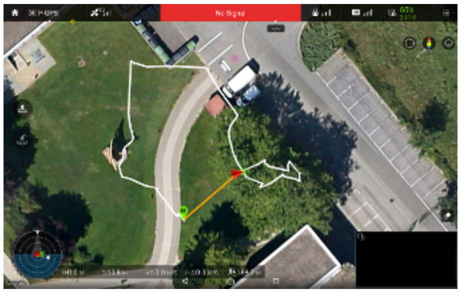

Fig. 14. top view and gps information of the environment

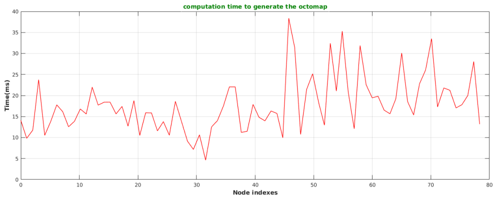

Fig. 15. Total time response to process all the nodes

V. Conclusions

The presented work shows a real time online map generation based on graph-SLAM with loop closure detection optimized using g2o which is applied on small UAVs. Two different experiment have been done in different real world scenarios. One in a low tree density forest (cluttered environment) and in a outdoor open space with reduced features. The obtained results satisfied the real time constraints generated by the stereo camera and the environment, with an average of 17.78 and 9.691 ms. The odometery lost occurs, when there was a completely lack of features for only 100ms. The total number of node indexes processed were 190 and 78 in the cluttered environment and in open field respectively. Though the outdoor scenario is mixed with uncertainties of illumination and varying features, the experimental results shows that our approach satisfy the real time constraints. As a future work, the presented approach will be tested with path planning algorithm along with suitable control system approach to attain the collision avoidance task .

References

[1] A. Angeli, D. Filliat, S. Doncieux, and J. A. Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics*, 24(5):1027–1037, Oct 2008.

[2] D. Assistant. http://www.dji.com/de/product/matrice100/info.

[3] A. J. Barry and R. Tedrake. Pushbroom stereo for high-speed navigation in cluttered environments. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3046–3052, May 2015.

[4] A. Ben-Afia, L. Deambrogio, D. Salos, A. C. Escher, C. Macabiau, L. Soulier, and V. Gay-Bellile. Review and classification of vision-based localisation techniques in unknown environments. *IET Radar, Sonar Navigation*, 8(9):1059–1072, 2014.

[5] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*. O'Reilly Media, Inc., 2nd edition, 2013.

[6] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard. Simultaneous localization and mapping: Present, future, and the robust-perception age. *CoRR*, abs/1606.05830, 2016.

[7] J. Castellanos, R. Martinez-Cantin, J. Tards, and J. Neira. Robocentric map joining: Improving the consistency of ekf-slam. *Robotics and Autonomous Systems*, 55(1):21 – 29, 2007. Simultaneous Localisation and Map Building.

[8] J. A. Castellanos, J. Neira, and J. D. Tards. Limits to the consistency of ekf-based slam 1.

[9] J. Courbon, Y. Mezouar, N. Gunard, and P. Martinet. Vision-based navigation of unmanned aerial vehicles. *Control Engineering Practice*, 18(7):789 – 799, 2010. Special Issue on Aerial Robotics.

[10] G. N. Desouza and A. C. Kak. Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, Feb 2002.

[11] G. Grisetti, C. Stachniss, and W. Burgard. Nonlinear constraint network optimization for efficient map learning. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):428–439, Sept 2009.

[12] K. L. Ho and P. Newman. Loop closure detection in {SLAM} by combining visual and spatial appearance. *Robotics and Autonomous Systems*, 54(9):740 – 749, 2006. Selected papers from the 2nd European Conference on Mobile Robots (ECMR 05)2nd European Conference on Mobile Robots.

[13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[14] S. Huang and G. Dissanayake. Convergence analysis for extended kalman filter based slam. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 412–417, May 2006.

[15] M. Jalobeanu, G. Shirakyan, G. Parent, H. Kikkeri, B. Peasley, and A. Feniello. Reliable kinect-based navigation in large indoor environments. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 495–502, May 2015.

[16] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3607–3613, Shanghai, China, May 2011.

[17] J. Li, Y. Bi, MengluLan, H. Qin, M. Shan, F. Lin, and B. M.Chen. Real-time simultaneous localization and mapping for uav: A survey. *International Micro Air Vehicle Conference and Competition (IMAV)*, 2010.

[18] M. Liu, S. Huang, and G. Dissanayake. Feature based slam using laser sensor data with maximized information usage. In *2011 IEEE International Conference on Robotics and Automation*, pages 1811–1816, May 2011.

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004.

[20] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 1997.

[21] D. Magree, J. G. Mooney, and E. N. Johnson. Monocular visual mapping for obstacle avoidance on uavs. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 471–479, May 2013.

[22] D. manifold manual. https://dl.djicdn.com/downloads/manifold/en/ manifold usermanual env 1.0.pdf.

[23] L. Marchetti, G. Grisetti, and L. Iocchi. *A Comparative Analysis of Particle Filter Based Localization Methods*, pages 442–449. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[24] R. Martinez-Cantin, N. de Freitas, and J. A. Castellanos. Analysis of particle methods for simultaneous robot localization and mapping and a new algorithm: Marginal-slam. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2415–2420, April 2007.

[25] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 2, pages 2051–2058 vol.2, 2001.

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

[27] S. Thrun and M. Montemerlo. The graph slam algorithm with applications to large-scale mapping of urban structures. *Int. J. Rob. Res.*, 25(5-6):403–429, May 2006.

[28] J. Uijlings, A. Smeulders, and R. Scha. Real-time bag-of-words, approximately. In *CIVR*, 2009.