Simultaneous Localization and Mapping for micro Unmanned Aerial Vehicles

P. Tsilivis[1], G. Nikolaou[2]

[1] Dpt. of Automation, Piraeus University of Applied Sciences, Athens, Greece
Tel: +30 6979670375, E-mail: p.tsilivis10@gmail.com

[2] Dpt. of Automation, Piraeus University of Applied Sciences, Athens, Greece
Tel: +30 2105381314, E-mail: nikolaou@teipir.gr

Abstract

The aim of this paper is to present and analyze the basics of the simultaneous localization and mapping (SLAM) algorithm, implemented in micro Unmanned Aerial Vehicles – UAVs. In particular, two of the most essential elements for autonomous drone navigation are the localization and mapping. For a UAV, the ability to predict its position and recognize the surrounding environment is of the utmost importance. Hence, there are numerous techniques that rely on various sensors, such as navigation based on camera or localization based on GPS and laser beacons. These methods improve the performance of the self-navigating drone regarding its stability during the flight and its response in unknown environments. Owing to the advanced level of difficulty, these algorithms were developed with the view of operational consistency, not for perfection. This paper will initially present the fundamentals of SLAM, along with a detailed description of the importance of SLAM algorithms in regards to the autonomous navigation of a UAV. Secondly, the most widely adopted SLAM algorithms will be presented and explained. In the last part of this paper, an implementation of SLAM on a low cost commercial drone, will be presented.

1. Description of the Simultaneous Localization & Mapping in regards the UAVs

The development of the software part of an Unmanned Aerial Vehicle (UAV) is divided in two main subsystems: (1) the control system and (2) the navigation system. **Simultaneous Localization and Mapping (SLAM)** is the master feature of the navigation subsystem, due to its real-time influence in the entire performance of the drone.

In particular, the SLAM is defined as a problem of continuous localization and mapping in an unknown environment. The absence of SLAM capabilities in a drone's navigation system, may cause potential "blindness" to the drone, due to its space ignorance. The reason for this are the inaccurate measurements produced by the drone's sensors along with the increased computing power needed for pose estimation. In addition, the combination of these along with other major external or not parameters, such as wind turbulences or suboptimal tuned control system, set the drone in need for advanced localization and mapping algorithms, so as to maintain a stable autonomous flight.

The *Simultaneous Localization and Mapping* task consists of two main operations, named *Localization* and *Mapping*. **Localization** is the term that refers to the continuous estimation of the drone's current position, based on features taken from the map that the SLAM algorithm itself has created. The second term, **mapping**, refers to the process of creating a feature-based map containing essential key points of the environment, which can be detected by the drone's sensors. These elements cooperate to produce a detailed map that can provide a real-time position estimation of the drone in a previously-unknown environment. The problem of SLAM is being solved by advanced mathematical algorithms, which detect features from sensor data and translate them into valuable information that compose a coordinate system.

Description of the main mindset regarding the SLAM algorithms

The fundamental mindset of the SLAM algorithms is based on approaches of **probability** and **scan matching** techniques. Sensors such as stereo or monocular cameras or laser beacons, mounted on the drone, provide these algorithms with essential information / inputs (images, videos, etc.). The ***Probability*** techniques are based on the **Bayes Filter** [1], **Kalman Filter** [2] and also, when the non-linear nature of the drone is considered, on the **Extended Kalman Filter (EKF)** [3]. The state estimation process involved is the most essential feature in this technique, since it is the core procedure for the real-time localization. Hence, during this process an estimated belief is calculated based on the previous states of the drone and the measurements data. In particular, a current state of the drone, which describes its velocity, acceleration and pose at every time instant, is the output of a two-step method: (1) the **Prediction Step**, (2) and the **Update Step**. Starting from the prediction step, the initial belief of the state of the drone is calculated based on the measurements and control data, in addition to the motion model. Afterwards, the update step is responsible for the update of the initial belief, taking into account the initial measurements and the sensor model; combining these, produces the final state estimation outcome. Last but not least, these steps are an iterative process so as to continually estimate the drone's position. The ***Scan Matching*** techniques vary among feature to feature methods, point to point or even feature to point. These techniques are based on the analysis of the image in features or points, such as corners, edges and lines, or simple points. The main difference between the features and the points is that *features* are more accurate but more difficult to extract them from the image, while *points* can be easily extracted but also contain lower information quality. Hence, an **F2F** (feature to feature) method scans the captured image of the environment and matches the features of the first image with the features of the next image. In contrast, a **P2P** (point to point) method uses the Iterative Closest Point (ICP) [4] technique in order to match each initial point with the closest final belief point. The outcome of these methods is the estimation of the relative position of the drone on the map.

2. Motivation

2.1 Importance of SLAM Algorithms

While robots are developed to move independently in environments unknown to them, the navigation algorithms constitute the most essential tool for autonomous motion. Every robotic machine, such as a simple commercial drone or even a complex planetary rover, presupposes the existence of a navigation system able to meet the requirements of each desirable performance of a robot. Hence, as the most basic features of a navigation system are the real-time localization along with the accurate mapping of a given environment, algorithms which are part of the SLAM mindset are located high in the pyramid of importance, with regards to the effect on the performance of the robot. In particular, and as it described above, SLAM algorithms provide the robot with the ability to continually map the environment that surrounds it, along with the feature of tracking its position regarding its own map, continually; something which is quite demanding in regards the real-time process of the resulting data in order to both map and localize. Thus, SLAM algorithms provide the robots with essential benefits concerning its simultaneous navigation performance, which are difficult to be replaced by other simpler methods.

2.2 Correlation to the Autonomous Navigation

In the field of Autonomous Navigation and especially with the use of a **Micro Aerial Vehicle (MAV)**, the SLAM algorithms are of the utmost importance for the stability and the performance of every drone. According to the above-mentioned, the Simultaneous Localization and Mapping is the basic tool that renders the navigation of a drone as completely autonomous. It enables not only the drone to know its location every moment, but also predicts the drone’s next movement and provides supplementary navigation methods with valuable information. Hence, implementing a SLAM algorithm into the development of a drone, offers also a significant aid to the control system providing valuable feedback in a way of tuning the control parameters to their optimum. All in all, the effect of the SLAM algorithms into the whole system is directly linked to the control system, since both are sharing the same information and both are responsible for the stability and motion of the drone; in other words, they are both vital systems for the performance of a micro UAV.

3. SLAM Essentials

3.1 Technical Description of SLAM techniques based on micro UAVs

Owing to the fact that the Kalman Filter and the Extended Kalman Filter are vital methods for the control system, they are commonly integrated into both the control system and the navigation. One approach to solve the SLAM problem is to use vision sensors. The techniques used for the identification and description of the environment that the drone operates, rely on the detection of features extracted from the captured images of the drone and differ depending on the variation of the features. Specifically, the feature-based SLAM algorithms follow two main steps as a procedure of feature extraction, description and setup parameterization so as to reach the desired outcome. The first step of the process is the application of the feature extraction and description methods, so as to extract sets of features from the image and describe them into the system as observations. The second and more important step is the estimation of the exact location of the hardware sensor, which in most cases of a commercial MAV is a monocular camera, along with the geometry of the environment. These estimations are made in function of the first step's observations and the combination of these leads to the final outcome. The interim procedure of the estimation is also known as the method of the **Inverse Depth Parametrization** [5]. This particular method solves the essential problem of the close distance features that cannot be identified, while using the standard EKF framework. In other words, the depth of the initial observed features is parametrized to an inverse depth, relative to the camera position. With this method, the features can be identified "even up to infinity" depths [5] and it is used exclusively in the **Monocular SLAM** algorithms.

3.2 SLAM Algorithms

The second half of the SLAM algorithms, in other words the vision part of each algorithm, is consisted of the feature extraction and feature description techniques. These techniques detect every feature in the drone's captured images and analyse them in sets, following the system's standards.

Feature Extraction

The feature extraction methods are the initialisation of the SLAM algorithm. While the drone flies around its environment, the monocular camera captures multiple images that describe its view; these images are the input of the feature extraction algorithms. Due to the diversity of these features, the extraction techniques are categorised based on the most common features and they are divided into more than one classes. In particular, there are two main categories that differentiate these techniques; namely, (1) the **edge detectors** and (2) the **corner detectors**.

The ***edge detectors*** are responsible for tracking the edges of an image and selecting the most distinguishable of them. The process of such a method begins with the application of a **Gaussian filter** in order to remove the noise of the image. The Gaussian filter applies smoothing and filtering into the resolution of the image, thereby cleans the image from unnecessary noise. Afterwards, the **intensity**

gradients of the image are analyzed so that the method to focus at certain parts of the image. This part of the technique helps improve the speed of the procedure by decreasing unnecessary processing time of useless image content, such as large non-graduating intensity parts of an image. Finally, after the completion of the above steps, a **double threshold binarization technique** [6] is being applied for determining the potential edges, in addition to the detection of the edges with weak presence and poor connection with others; these are being suppressed in order not to confuse the system. A very effective method is the **Canny edge detector** [7], while methods such as the **Deriche edge detector** [8], or the **differential edge detector** are improvements of the original Canny method.

The ***corner detectors*** form the second category of the feature extraction techniques, correspond to the detection of sharp corners inside an image. These techniques analyze the image pixel by pixel and compare each detail of a pixel in order to detect the shape of a corner. Specifically, the **Moravec corner detection** [9] method, a very essential technique, compares every pixel of the image with its neighbor, so as to spot a difference among their density. The final output of this technique contains the variable called “**similarity”**, which is the sum of the squared differences **(SSD)** among the respective pixels; the smaller this output is, the more likely is that the corner is detected. The improvement of the Moravec method, also called **Harris & Stephens corner detection** [9] differs in the way that the algorithm estimates the similarity. In particular, the algorithm uses the **differential** of the respective pixels instead of summing the squared differences among themselves, while taking into account the direction of the pixel that tend. This differential logic is called **“autocorrelation”** and implements a faster and more consistent way for detecting corners.

Feature Description

After the fundamental section of the feature extraction, follows the feature description. This task contains techniques that not only describe the detected features into the system, but spot the most important key points of the total. These algorithms are based on the **Points of Interest – PoI**, which are used as means of defining and characterizing the features of every detected object in an image. The most applied algorithms in this section are: (1) the **Histogram of Oriented Gradients – HOG** [11] method, (2) the **Scale-invariant Feature Transform – SIFT** [12] method and (3) the **Speeded Up Robust Features – SURF** [13] method.

The **HOG** method is one of the most widespread algorithms for object and face detection. The basic logic behind this algorithm is based on the separation of the image in many small portions, called **cells**. The figures that are present within the image along with their appearance, is described in the system as **intensity gradients** and **edge variations**. Hence, every cell contains a part of the image, defined from the above, in which the summary of the pixels inside the cell forming a **histogram** of intensity gradients. Thus, the final output is described by the total of the histograms contained in the image, in which case is called **descriptor**. Moreover, and besides the above basic operation, this method aims to improve the quality of the recognition by applying a contrast normalization; in other words, estimates the volume of the intensity in a larger area called **block**. In this way, the estimated value becomes a **stabilizer** among the diversity of the intensities and it is applied to each histogram inside the block. This technique benefits in a way of smoothing the illumination and the shadowing.

On the other hand, the **SIFT** method uses the **PoI** technique in order to cope with the common problem of the incorrect mapping of features from image to image. In particular, this algorithm detects points of interest **(key points)** from a set of images and stores them in a database, so as to constitute the basic data for the comparison of the objects. Hence, a new object will be detected when the key points from the database coincide with the new features, based on their Euclidean distance; these new features are called **"good matches"** depending on the matching of their magnitude, orientation and position. Last but not least, the final output of these algorithms consist of **clusters** of "good matches", while make use of the **Hough transform** in order to untwist the outliers.

Finally, the **SURF** method is an essential improvement of the SIFT algorithm, in a way of speed and performance stability. While the above algorithms make use of the corner and edge detection methods, SURF uses **blob detectors** instead. Specifically, this method compares the image in terms of density and color variations, through mathematical estimations, and uses the **"multi-resolution Pyramid"** [14] technique in order to convert the points of interest into coordinates. As a result, a copy of the image is created and with the implementation of the Laplacian and Gaussian Pyramid techniques, the new image has reduced bandwidth but same size. Hence, the outcome of this method is the application of a custom filter, based on the standards of each image and called "**scale-space"**, in which the points of interest do not affect the total feature comparison in regards the magnitude.

The above techniques are applied in combination into the navigation system of a micro UAV, due to the complexity and the importunity nature.

4. SLAM Applications in micro UAVs

4.1 Monocular and Direct SLAM

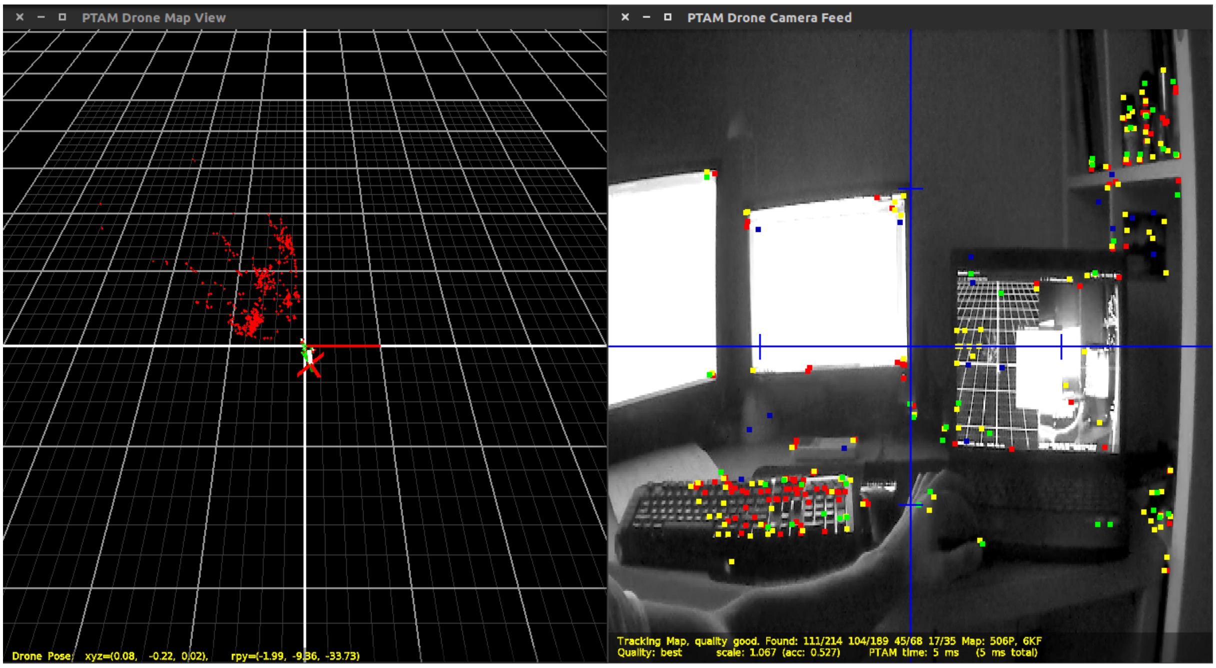
The navigation system of a drone is based on its vision sensors in cooperation with the SLAM algorithms. However, in the case of the micro unmanned aerial vehicles their vision system is usually consisted of one monocular camera, for commercial drones, or with the addition of laser beacons, in military applications, for optimal tracking. In such case, the problem that a drone equipped with a monocular camera has to cope with, is the depth parameterization. In other words, the particular drone has to observe the environment in 3 dimensions, adding the depth of the scene into the equations of the SLAM algorithm, with the use of a 2-dimensional vision sensor. This technique is called **Monocular SLAM** and its fundamental logic is about the tracking of the depth in a scenery with the use of a 2-dimensional camera, and not with stereo cameras. Such algorithms are divided in two basic categories: (1) the **feature-based SLAM**, which is described above, and (2) the **direct SLAM**. The difference between them, is that the feature-based algorithms are based on the feature extraction and description techniques, while the direct SLAM algorithms skip these steps and analyze the entire image pixel per pixel.

In particular, the **direct SLAM** is developed exclusively for drones equipped with just a monocular camera, while the exact name of this method is **Large-Scale Direct Monocular SLAM** or **LSD-SLAM** [15]. In contrast to the feature-based methods, this technique uses **image-to-image alignment** in order to track the exact location of the drone. Specifically, the alignment is performed by a **coarse-to-fine** algorithm; this algorithm analyses a small set of data by triangulation and after the completion of the each calculation it adds another small set of data, until the error between the last triangular result and the total to be within a bandwidth. Moreover, this algorithm includes the **Huber loss** equation, which provides the same benefits as the **SSD**, but with more resilience in the extreme values. Additionally, the depth estimation on the environment is performed by the inverse depth parameterization with a use of a small number of image sets. This method makes use of the **semi-dense** technique in order to create a depth map by the information in the image boundaries. Hence, the main difference between the direct SLAM and the common feature-based SLAM algorithms is that the particular method is based on a **texture tracking** technique and not on the characteristics of each image.

4.2 PTAM – Parallel Tracking & Mapping

This particular algorithm is developed for commercial micro UAVs, in order to track in real-time the position of the drone in space. Specifically, the **Parallel Tracking & Mapping – PTAM** [16] algorithm is based on the monocular SLAM logic; the navigation system tracks the position of the drone in real-time and also creates a 3-dimensional map of the environment. This method is divided in two main procedures: (1) the **tracking** and (2) the **mapping**. The ***tracking*** process is performed by the combination of feature-based algorithms and stereo initialization techniques. In particular, the position of the drone is estimated via the triangulation of the observations received by the feature extraction and description methods, along with stereo initialization techniques; in which the drone is able to observe the environment from multiple points of view. In addition, this algorithm makes use of the pyramid technique, which divides the captured scenery in multiple layers consisted of various video frames. With this process the system is able to track the relative position of the drone in real-time and with great stability in terms of blurring or fast movements. The ***mapping*** process, on the other hand, is highly linked with the tracking due to the information exchange between them. This process is described by the collection of sets of data, called **key frames**, which include information about the environment. The procedure of the particular collection is essentially filtered, in order to achieve stability in the estimation of the errors and also to enable the ability of reviewing each frame so as to improve the quality. The result of this process is the continuous improvement of the map quality along with the position tracking.

The *Picture 1*, below, illustrates the practical application of the PTAM algorithm with the use of a **Parrot AR Drone 2.0** [17]**,** which was developed within the context of laboratory application in our teams lab at Piraeus University of Applied Sciences. On the left side of the picture, it is indicated the map of the environment, based on the key frames received from the tracking process, while on the right side it is illustrated, via key points, the essential features detected from the extraction and description techniques. This constitutes a legit example of a SLAM algorithm implementation into a micro Unmanned Aerial Vehicle.



Picture 1: PTAM illustration with the use of the Parrot AR Drone [18]

5. Discussion

The above illustration summarizes the SLAM algorithm essentials described in the particular paper. Specifically, the application developed in our laboratory was focused on the indoor autonomous navigation and its main aim was the inspection of a desired object. The results of the particular application showed the importance of the SLAM algorithm in many ways. One of the most significant factors was the choice of the Extended Kalman Filter algorithm, which was an essential element for both the vision part and the stability of the drone due to the prediction information. In addition, the advanced technique of the PTAM algorithm enabled the drone to track numerous features, regardless of the illumination and the shadowing of the scenery, that always constituted a sufficient input for the creation of the 3-dimensional map. For instance, *Picture 1* was captured in a dark environment with the on-board camera of the drone. Nevertheless, as it is illustrated, the system had enough information for the initialization of the localization and mapping. Moreover, as it is depicted on the right side of the *Picture 1*, the majority of the detected features are consisted by corners and edges. The reason is that PTAM makes use of basic and reliable methods for the extraction and description of the features, such are the above mentioned techniques. Last but not least, *Picture 1* demonstrates the efficiency of the depth detection from a monocular camera with the use of the Inverse Depth Parametrization method; this is illustrated in the left side of the *Picture 1*, with the perspective view being from above. All in all, SLAM algorithms constitute perhaps the most essential element in any drone navigation system, due to their direct influence in every part of its performance.

6. References

1] S. M. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE TRANSACTIONS ON SIGNAL PROCESSING,* vol. 50, FEBRUARY 2002.

2] R. E. KALMAN, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME – Journal of Basic Engineering,,* pp. 35-45, 1960.

3] S. J. JULIER and J. K. UHLMANN, "Unscented Filtering and Nonlinear Estimation," *PROCEEDINGS OF THE IEEE,* vol. 92, pp. 401-422, MARCH 2004.

4] Y. Chen and G. Medioni, "Object Modeling by Registration of Multiple Range Images," in *IEEE Int. Conf. On Robotics and Automation*, 1992.

5] J. Civera, A. J. Davison and J. M. Martinez Montiel, "Inverse Depth Parametrization for Monocular SLAM," *IEEE TRANSACTIONS ON ROBOTICS,* vol. 24, October 2008.

6] Q. Chen, Q.-s. Sun, P. A. Heng and D.-s. Xia, "A double-threshold image binarization method based on edge detector". China/Hong Kong Patent Pattern Recognition 41, 1254 – 1267, 10 September 2007.

7] J. Canny, "A Computational Approach To Edge Detection," in *Pattern Analysis and Machine Intelligence, IEEE Trans*, 1986.

8] R. Deriche, "Using Canny's criteria to derive a recursively implemented optimal edge detector," in *Computer Vision, Int. J.*, 1987, pp. 167-187.

9] H. P. Moravec, "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover," Pennsylvani, 1980.

10] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, United Kingdom, 1988.

11] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Conference on Computer Vision and Pattern Recognition*, France, 2005.

12] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision*, 1999.

13] H. Bay, A. Ess and T. Tuytelaars, "Speeded Up Robust Features (SURF)," ETH Zurich, Belgium.

14] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt and J. M. Ogden, "Pyramid Methods in Image Processing," RCA Engineer, 1984.

15] J. Engel, T. Schops and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," Technical University Munich, Munich, 2014.

16] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *6th IEEE and ACM International Symposium on. IEEE*, 2007.

17] "Parrot," Parrot SA, [Online]. Available: http://www.parrot.com/uk/products/ardrone-2/..

18] J. Engel, "Computer Vision Group," Informatik IX, Computer Vision Group, [Online]. Available: http://vision.in.tum.de/data/software/tum_ardrone.