A vision-based assistance key differenciator for helicopters automonous scalable missions

Rémi Girard, Sébastien Mavromatis, Jean Sequeira, Nicolas Belanger, Guillaume Anoufa

A vision-based assistance key differenciator for helicopters automonous scalable missions

Rémi Girard[1,2], Sébastien Mavromatis[1], Jean Sequeira[1], Nicolas Belanger[2], and Guillaume Anoufa[3]

[1] Aix Marseille University, CNRS, LIS, Marseille, France
remi.girard@lis-lab.fr
[2] Airbus, Marignane, France
[3] Capgemini, France

Abstract. In the coming years, incremental automation will be the main challenge in the development of highly versatile helicopter technologies. To support this effort, vision-based systems are becoming a mandatory technological foundation for helicopter avionics. Among the different advantages that computer vision can provide for flight assistance, navigation in a GPS-denied environment is an important focus for Airbus because it is relevant for various types of missions. The present position paper introduces the different available SLAM algorithms, along with their limitations and advantages, for addressing vision-based navigation problems for helicopters. The reasons why Visual SLAM is of interest for our application are detailed. For an embedded application for helicopters, it is necessary to robustify the VSLAM algorithm with a special focus on the data model to be exchanged with the autopilot. Finally, we discuss future decisional architecture principles from the perspective of making vision-based navigation the 4th contributing agent in a wider distributed intelligence system composed of the autopilot, the flight management system and the crew.

Keywords: Visual SLAM · Vision-based Navigation · Helicopters · Autonomous · Pose estimation · 3D reconstruction

1 Introduction

Autonomous navigation has become one of Airbus Helicopters' top priorities. Helicopter missions are extremely varied in nature, including freight transport, medical evacuation, search and rescue, logistics support, police operations, and aerial work.

Autonomous helicopter control will enhance the performance and security of these missions. Helicopter operations can take place in urban areas, in mountains, in hostile areas, at low altitude, at night, etc. Most of the time, only helicopters are able to operate in these types of conditions. These missions are often dangerous, and all of the following factors, among others, can contribute to the occurrence of adverse events:

- Loss of visibility during Visual Flight Rules (VFR) navigation
- Loss of GPS during Instrument Flight Rules (IFR) navigation
- Loss of the autopilot
- Loss of engine

Accidents can be caused by difficult flight conditions, the malfunctioning of a flight instrument or undetected drift. If not detected by the crew, such an occurrence may mislead them into improperly guiding the helicopter. Controlled Flight Into Terrain (CFIT) is a common type of accident most often caused by the failure of the pilot to know at all times what the position of his or her craft is and how that actual position relates to the altitude of the surface of the Earth below and immediately ahead along the flight course. Hence, missions could be made more secure thanks to our vision-based piloting system.

2 Vision-based piloting assistance

Our work focuses on the design of a system aiming to improve the safety of operations without directly acting on the control of the helicopter. Our system should have the following capabilities:

- To compute the helicopter's pose in real time considering the environment in which the helicopter is moving. An estimated trajectory should be computed from consecutive poses and compared to the pre-established path.
- To be independent from the helicopter flight instruments. Indeed, a standalone system would add redundancy, resulting in more secure flight.

The computation of the helicopter's pose in real time would enable the detection of an error in the trajectory without the aid of the flight instruments in cases of GPS loss or undetected drift. Being able to detect such errors and make them deterministic could aid in the detection of unplanned trajectories, thus drastically reducing the number of crashes. Our decision-making system should warn the autopilot in cases of trajectory errors. Such an autonomous device embedded in a helicopter could directly influence the mission strategy. The intent is for our device to become a real mission assistance system.

Decisions can be of several types depending on the magnitude of the error and the type of undesirable event: go back, compute a new path, perform an emergency landing, continue the mission, return to base, stop, stay hovering, etc.

The device must operate autonomously and constantly exchange information with the helicopter's autopilot, as our system is not intended as a navigation device but as autopilot assistance. The exchange with the autopilot will need to be reliable. Simultaneous Localization And Mapping (SLAM) is a method for the simultaneous estimation of the state of a system and reconstruction of an environment map. Localization refers to the computation of the system's pose in the reconstructed environment, from which its position in the real environment can be deduced. Mapping refers to the representation of the environment and the interpretation of the data provided by sensors. Our system must be able to simultaneously process these two interdependent phases. The different available state-of-the-art SLAM methods will be detailed below.

In addition to its primary purpose, the system could also be used for environmental reconstruction. Such 3D reconstruction is interesting for the following purposes:

- Identifying terrain characteristics (e.g., slope or ground flatness).
- Identifying eligible landing zones as the mission progresses. These landing zones would be recorded in a database accessible by the autopilot.

Computer vision algorithms make it possible to analyze, process and understand an environment from images acquired by a camera system. The use of a camera is particularly interesting for cases of landing zone identification. However, a disadvantage of using a camera is the lack of a scale factor. This can be corrected by associating the camera with other sensors. Our decision-making system must be able to perceive the environment in which the helicopter is moving up to 1500 meters away. Furthermore, it will be embedded in helicopters; thus, it must be as small as possible. Therefore, our decision-making system will use a camera as its main sensor. The implementation of SLAM with cameras is known as Visual SLAM.

The following section presents an overview of the state of the art with regard to SLAM for vision-based piloting assistance. The end of this paper is devoted to our recommendations for system development.

3 SLAM for autopilot support

In this section, we discuss the literature on SLAM and Visual SLAM.

3.1 History

SLAM first emerged in the 1980s. Randall C.Smith, Peter Cheeseman [20] and Durrant-Whyte [6] defined a relation between the position of a sensor and the structure of the environment. Durrant-Whyte was the representative individual addressing the SLAM problem during the first 20 years. Throughout this period, the issue was seen as a probabilistic and statistical problem. The first approaches to SLAM were based on filters: the extended Kalman filter (EKF), Rao-Blackwell's particle filters, and maximum likelihood estimation. The filter-based approaches were summarized by Durrant-Whyte and Bailey in [5] and [1].

In 1990, Randall C. Smith *et al.* [19] proposed an EKF-based method and presented the concept of a stochastic map. They used the EKF to compute a state vector comprising the positions of the points of interest within an estimated map. The uncertainty of the estimates was represented by a probability density. These methods have several constraints: the state vector increases linearly with the size of the map, and the computational complexity is usually quadratic.

These limitations have led to the development of more advanced SLAM methods, such as the work of Montemerlo *et al.* [14], who proposed the FastSLAM algorithm. FastSLAM is also based on a filtering approach. Maps are generated with the EKF, while the robot's position is represented by distributions of set of particles, where each particle represents a trajectory. This method reduces the complexity of the algorithm, but the position estimates are not accurate, especially for long trajectories.

A graph-based approach has also been used to solve the SLAM problem. In this approach, the landmarks on the map and the poses of the robot are represented by nodes in a graph. Graph-based methods have the advantage of being applicable to much larger maps than EKF approaches. During the initial period of SLAM development (1986-2004), termed the classical age by [3], the sensors used mainly consisted of radars, lidars and sonars.

A new period of development for SLAM algorithms emerged when researchers became interested in information contained in images from cameras. The corresponding approach is known as Visual SLAM (monocular when only one camera is used and stereo when two cameras are used).

3.2 Visual SLAM

The main steps of a feature-based VSLAM algorithm are as follows:

1. extracting a set of salient image features from each keyframe,
2. defining each feature by means of a descriptor,
3. matching the features using the feature descriptors,
4. using epipolar geometry to compute both the camera motion and structure, and
5. using optimization methods, e.g., BA or loop closure, to refine the pose.

An interesting comparison of recents open-sources VSLAM and VO algorithms can be found in table 1 at the end of this section.

Five types of methods can be identified from the Visual SLAM (VSLAM) literature.

Feature based - Filtering methods: The first SLAM system working in real time using a single camera (MonoSLAM) was presented in 2007 by Davison *et al.* [4]. 6 Degree of freedom (DoF) camera motion and 3D positions of feature points are represented as a state vectorin EKF. Further work was inspired by Davison's work. Building on the work of the classical age, the first algorithms for VSLAM were mainly based on filters. These techniques have several disadvantages, such as long computation times, the propagation of linearization errors and the inability to function properly during sudden motion. In large environments, the size of a state vector increases because the number of feature points is large. EKF-SLAM maps are of very poor density, making them suitable for localizing a camera only within a very small environment. Other VSLAM algorithms later emerged that were better suited for operating in real time and in larger environments.

Feature based - Keyframe methods: In 2007, Klein and Murray proposed a new real-time visual SLAM system (PTAM [12]). They introduced the idea of separating the computation of the camera's pose and the mapping of the environment into two different threads. One thread deals with the camera pose estimation and the selection of keyframes, while the other creates and updates a 3D map. This parallelization enabled the use of bundle adjustment (BA) techniques [25] on a set of keyframes. BA techniques are optimization methods based on the minimization of the reproduction error. One of the significant contributions of PTAM is to introduce the use of keyframe. Strasdat *et al.* [21] demonstrated that for the same computation time, a VSLAM algorithm based on keyframes and BA optimization is more accurate than filtering methods. Compared to MonoSLAM, the system can handle thousands of features points by splitting the tracking and the mapping into two threads on CPU. However, PTAM does not detect large loops, and relocalization is based on the correlations between low-resolution thumbnails of the keyframes, resulting in a low invariance to viewpoint. When a loop closure is detected, this information is used to reduce the error drift in both the camera path and the map. Subsequent works improved the PTAM algorithm, particularly for use in large environments [13] and [22]. Strasdat *et al.* [21] have shown that it is necessary to preserve as many points of interest as possible while conserving nonredundant keyframes. To improve an accuracy of VSLAM, it is important to increase the number of feature points in a map. In 2017, Mur-Artal *et al.* proposed ORB-SLAM2 [15] based on their earlier algorithm named ORB-SLAM [16]. This VSLAM algorithm can operate in real time, in any environment, in monocular, RGB-D or stereovision mode. ORB-SLAM2 is divided into 4 modules: tracking, reconstruction, position optimization and loop detection. These four phases are executed in three different threads. In contrast to PTAM, ORB-SLAM2 achieves robustness under challenging conditions by inserting keyframes as quickly as possible and removing the most redundant images. ORB-SLAM2 is based on the main ideas of PTAM, the place recognition work of Galvez-López and Tardos [11], the scale-aware loop closing of Strasdat *et al.* [23], and the use of covisibility information for large-scale operation.

Direct methods: In contrast to feature-based methods, direct methods estimate structure and motion based directly on the pixel-level intensities in images. The Stereo Large-Scale Direct SLAM (LSD-SLAM) method presented by Engel *et al.* [8] is a semidense direct approach that minimizes photometric error in image regions with high gradients. This method is expected to be relatively robust to motion blur or poorly textured environments, they are also called featureless approaches.

Semidirect methods: The Semidirect monocular Visual Odometry (SVO) [9] algorithm is a visual odometry method based on the semidirect approach. It is a hybrid method with a combination of the characteristics of the previous two types of methods. The tracking is done by feature point matching, the mapping is done by the direct method. It inherits some of the drawbacks of direct methods and discards the optimization and loop detection steps.

Visual odometry: The concept of visual odometry (VO) has been summarized by Davide Scaramuzza and Friedrich Fraundorfer in [18] and [10]. The term VO was first introduced by Nister *et al.* [17]. The aim of VSLAM is to compute a global, consistent estimate of the system path, while the goal of VO is to calculate the path incrementally, pose by pose. VO can be used as one part of a complete SLAM algorithm.

$$\text{VSLAM} = \text{VO} + \text{global map optimization}$$

All VSLAM works described in the literature show that detecting and processing all points of interest in the entire image in every frame is not possible, particularly in the case of embedded systems.

Algorithm	Method	Scene type	Map density	GO	LC
MonoSLAM [4] www.doc.ic.ac.uk/~ajd/Scene/index.html	Feature	Small and ind.	Sparse	No	No
PTAM [12] www.robots.ox.ac.uk/~gk/PTAM/download.html	Feature	Small	Sparse	Yes	No
LSD-SLAM [8] www.github.com/tum-vision/lsd_slam	Direct	Small or large	Semi-dense	Yes	Yes
SVO [9] www.github.com/uzh-rpg/rpg_svo	Semi-direct	Repetitive and h.f.t	Sparse	No	No
ORBSLAM2 [15] www.github.com/raulmur/ORB_SLAM2	Feature	Small or large	Sparse	Yes	Yes
DSO [7] www.github.com/JakobEngel/dso	Direct	Small or large	Sparse	No	No

Abreviations: Global Optimizations (GO), Loop Closure (LC), hight-frequency texture (h.f.t) and indoor (ind.).

Table 1. Comparison of recents open-sources VSLAM and VO algorithms.

3.3 Vision-based Navigation

The vast majority of vision-based navigation work has been implemented on micro aerial vehicles (MAVs) or small helicopters. However, these implementations also rely on inertial data most of the time. Vision-based navigation systems for flying vehicles must be capable of estimating an agents pose from aerial views of the ground. Extracting and matching features from images acquired in a large and poorly textured environment at high speed is extremely challenging, especially for an embedded system with low computation capabilities. The authors of [24] and [26] embedded LSD-SLAM and SVO algorithms, respectively, in micro aerial vehicles using monocular cameras. Recently, David S. Bayard *et al.* [2] developed an alternative approach based on velocimetry for navigation on Mars. As noted by [3], we have now entered a new period of SLAM development, which the cited authors term the robust perception age. The robustification of existing SLAM algorithms to enable real applications is the major issue at stake.

4 Discussion and positioning

In this paper, an overview of SLAM methods for vision-based navigation is presented. Full SLAM solutions are challenging for real-time applications embedded in helicopters due to their computation-intensive filter-based state estimations. Our device will be an intermediary system between the pilot, the autopilot and the flight management system (FMS). In this mindset, we do not intend to implement the most powerful possible VSLAM algorithm. Instead, our main contribution will arise from the capability to construct a multiagent distributed intelligence system in which the VSLAM algorithm engages in the best possible dialogue with the other agents to support incremental navigation decisions all along the route. Our VSLAM system will have the ability to provide relevant navigation indicators based on image regions of interest to the autopilot to maintain a low error ratio. At the end of the day, the autopilot must be able to rely on this system as a basis for making decisions. Therefore, our most challenging task will be to robustify and format the output data of our VSLAM application.

References

1. Bailey, T., Durrant-Whyte, H.: Simultaneous localization and mapping (SLAM): part II. IEEE Robotics Automation Magazine **13**(3), 108–117 (Sep 2006). https://doi.org/10.1109/MRA.2006.1678144
2. Bayard, D.S., Conway, D.T., Brockers, R., Delaune, J.H., Matthies, L.H., Grip, H.F., Merewether, G.B., Brown, T.L., Martin, A.M.S.: Vision-Based Navigation for the NASA Mars Helicopter. In: AIAA Scitech 2019 Forum. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2019-1411, https://arc.aiaa.org/doi/abs/10.2514/6.2019-1411
3. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. IEEE Transactions on Robotics **32**(6), 1309–1332 (Dec 2016). https://doi.org/10.1109/TRO.2016.2624754
4. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: Real-Time Single Camera SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(6), 1052–1067 (Jun 2007). https://doi.org/10.1109/TPAMI.2007.1049

5. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part I. IEEE Robotics Automation Magazine **13**(2), 99–110 (Jun 2006). https://doi.org/10.1109/MRA.2006.1638022
6. Durrant-Whyte, H.F.: Uncertain geometry in robotics. IEEE Journal on Robotics and Automation **4**(1), 23–31 (Feb 1988). https://doi.org/10.1109/56.768
7. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE Transactions on Pattern Analysis and Machine Intelligence (Mar 2018)
8. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-Scale Direct Monocular SLAM. In: European Conference on Computer Vision (ECCV) (2014)
9. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: Fast semi-direct monocular visual odometry. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). pp. 15–22 (May 2014). https://doi.org/10.1109/ICRA.2014.6906584
10. Fraundorfer, F., Scaramuzza, D.: Visual odometry: Part II: Matching, robustness, optimization, and applications. IEEE Robotics & Automation Magazine **19**(2), 78–90 (2012). https://doi.org/10.1109/MRA.2012.2182810, https://graz.pure.elsevier.com/en/publications/visual-odometry-part-ii-matching-robustness-optimization-and-appl
11. Galvez-López, D., Tardos, J.D.: Bags of Binary Words for Fast Place Recognition in Image Sequences. IEEE Transactions on Robotics **28**(5), 1188–1197 (Oct 2012). https://doi.org/10.1109/TRO.2012.2197158
12. Klein, G., Murray, D.: Parallel Tracking and Mapping for Small AR Workspaces. In: 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. pp. 225–234 (Nov 2007). https://doi.org/10.1109/ISMAR.2007.4538852
13. Konolige, K., Agrawal, M.: Frameslam: from bundle adjustment to real-time visual mapping. ieee trans. robot. 24, 1066-1077. Robotics, IEEE Transactions on **24**, 1066 – 1077 (11 2008). https://doi.org/10.1109/TRO.2008.2004832
14. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In: In Proceedings of the AAAI National Conference on Artificial Intelligence. pp. 593–598. AAAI (2002)
15. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (Oct 2017). https://doi.org/10.1109/TRO.2017.2705103
16. Mur-Artal, R., Montiel, J., Tardos, J.: Orb-slam: a versatile and accurate monocular slam system. IEEE Transactions on Robotics **31**, 1147 – 1163 (10 2015). https://doi.org/10.1109/TRO.2015.2463671
17. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. vol. 1, pp. I–652–I–659 Vol.1 (Jun 2004). https://doi.org/10.1109/CVPR.2004.1315094
18. Scaramuzza, D., Fraundorfer, F.: Tutorial: Visual odometry. IEEE Robotics & Automation Magazine **18**(4), 80–92 (Dec 2011). https://doi.org/10.1109/MRA.2011.943233, https://graz.pure.elsevier.com/en/publications/tutorial-visual-odometry
19. Smith, R., Self, M., Cheeseman, P.: Estimating uncertain spatial relationships in robotics. In: 1987 IEEE International Conference on Robotics and Automation Proceedings. vol. 4, pp. 850–850 (Mar 1987). https://doi.org/10.1109/ROBOT.1987.1087846
20. Smith, R.C., Cheeseman, P.: On the Representation and Estimation of Spatial Uncertainly. Int. J. Rob. Res. **5**(4), 56–

68 (Dec 1986). https://doi.org/10.1177/027836498600500404, http://dx.doi.org/10.1177/027836498600500404

21. Strasdat, H., Montiel, J.M.M., Davison, A.J.: Real-time monocular SLAM: Why filter? In: 2010 IEEE International Conference on Robotics and Automation. pp. 2657–2664 (May 2010). https://doi.org/10.1109/ROBOT.2010.5509636
22. Strasdat, H., Davison, A.J., Montiel, J.M.M., Konolige, K.: Double Window Optimisation for Constant Time Visual SLAM. In: Proceedings of the 2011 International Conference on Computer Vision. pp. 2352–2359. ICCV '11, IEEE Computer Society, Washington, DC, USA (2011). https://doi.org/10.1109/ICCV.2011.6126517, http://dx.doi.org/10.1109/ICCV.2011.6126517
23. Strasdat, H., Montiel, J.M.M., Davison, A.J.: Scale Drift-Aware Large Scale Monocular SLAM. In: RSS 2010 (2010). https://doi.org/10.15607/RSS.2010.VI.010
24. Stumberg, L.v., Usenko, V., Engel, J., Stückler, J., Cremers, D.: From monocular SLAM to autonomous drone exploration. In: 2017 European Conference on Mobile Robots (ECMR). pp. 1–8 (Sep 2017). https://doi.org/10.1109/ECMR.2017.8098709
25. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle Adjustment A Modern Synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) Vision Algorithms: Theory and Practice. pp. 298–372. Lecture Notes in Computer Science, Springer Berlin Heidelberg (2000)
26. Yang, T., Li, P., Zhang, H., Li, J., Li, Z.: Monocular Vision SLAM-Based UAV Autonomous Landing in Emergencies and Unknown Environments. Electronics **7**(5), 73 (May 2018). https://doi.org/10.3390/electronics7050073, https://www.mdpi.com/2079-9292/7/5/73