



Geocaching-Inspired Navigation for Micro Aerial Vehicles with Fallible Place Recognition

Michel Barbeau¹, Joaquin Garcia-Alfaro²(✉), and Evangelos Kranakis¹

¹ Carleton University, Ottawa K1S 5B6, Canada

² Institut Polytechnique de Paris, SAMOVAR, Telecom SudParis, Palaiseau, France
jgalfaro@ieee.org

Abstract. This paper extends an existing decisional framework for the navigation of Micro Aerial Vehicle (MAV) swarms. The work finds inspiration in the geocaching outdoor game. It leverages place recognition methods, information sharing and collaborative work between MAVs. It is unique in that a priori none of the MAVs knows the trajectory, waypoints and destination. The MAVs collectively solve a series of problems that involve the recognition of physical places and determination of their GPS coordinates. Our algorithm builds upon various methods that had been created for place recognition. The need for a decisional framework comes from the fact that all methods are fallible and make place recognition errors. In this paper, we augment the navigation algorithm with a decisional framework resolving conflicts resulting from errors made by place recognition methods. The errors divide the members of a swarm with respect to the location of waypoints (i.e., some members continue the trip following the proper itinerary; others follow a wrong one). We propose four decisional algorithms to resolve conflicts among members of a swarm due to place recognition errors. The performance of the decisional algorithms is modeled and analyzed.

Keywords: Micro aerial vehicle · Drone formation control · Drone swarm · Goal location · Quadcopter · Information sharing · Localization · Path planning · Navigation · Place recognition

1 Introduction

Recent developments in MAV technologies enable several applications such as inspection of infrastructures [1], parcel delivery [2] and search and rescue operations [3]. Modern MAVs are equipped with cameras, GPS and various sensors. We focus our attention on the navigation problem for a swarm of MAVs. Navigation involves the planning of a path. Existing path planning approaches use results from the robotics literature such as artificial potential function [4], random tree [5] and Voronoi diagram [6]. Specific related issues have been addressed such as obstacle avoidance [7] and embedding of computational intelligence [8].

© Springer Nature Switzerland AG 2020

L. A. Grieco et al. (Eds.): ADHOC-NOW 2020, LNCS 12338, pp. 55–70, 2020.

https://doi.org/10.1007/978-3-030-61746-2_5

In this paper, we extend a decentralized path planning algorithm for MAV swarms, originally published in [9]. This algorithm is one of a kind. At the start, the MAVs do not know the trajectory they have to follow. Each MAV is given some information that enables the recognition of places along the way. While they proceed along a path, the MAVs cooperate and determine the GPS coordinates of a series of landmarks. The MAVs use their on-board cameras and other sensors, such as microphones, to collect observational data to recognize places and determine their GPS locations. They exchange together these locations to solve simple geometric problems, that is, determination of circle foci and line and perimeter intersections. The recognition of places and resolution of geometric problems determine the waypoints and how they are related. There are numerous methods available for place recognition, including navigation systems, visual recognition, environmental sound and thermal infrared imaging (cf. [10–14] for further details). However, none of them is perfect. They all make errors. The presence of errors misleads the members of a swarm with respect to the waypoint locations. In this paper, we address the problem of conflict resolution due to place recognition errors in the context of the swarm navigation algorithm originally presented in [9].

We create a decisional framework to resolve conflicts due to errors that occur during waypoint recognition. We propose four conflict resolution algorithms assembling the replies returned by place recognition queries executed by the MAVs. In four different ways, they apply the majority rule. The first algorithm establishes a baseline and formalizes what is done originally in [9]. The second algorithm leverages the availability of multiple place recognition methods. It is a two-level process. Within each MAV, the available place recognition methods are applied. The majority’s decision of the methods applied by an individual MAV is selected. Individual decisions are exchanged among the members of a swarm. The final decision is the majority’s decision of the swarm. The third algorithm is a flat process. Each MAV applies all available place recognition methods. All results are exchanged with all members of a swarm. The final result is the majority’s decision of all individual results obtained by all applications of the methods in a swarm. The fourth algorithm is similar to the second one, but it captures with a threshold the idea that some place recognition methods might not be even capable of producing results due to non favorable conditions (e.g., due to a high number of failures). We demonstrate through error probabilistic models that the second, third and fourth algorithms improve the performance of the baseline algorithm.

Section 2 surveys related work. Section 3 reviews the algorithm originally presented in [9]. Section 4 describes the decisional framework details of our algorithms and provides simulation results. Section 5 concludes the paper.

2 Related Work

Path planning for a swarm of MAVs involves collective movement of the ensemble from one waypoint to another, while at the same time avoiding intermediate obstacles and possibly according to a pattern. There is extensive scientific literature on the subject. Radmanesh et al. [7] made a survey on path planning

with obstacle avoidance. Otto et al. [15] are concerned with path planning with a coverage goal, while Zhao et al. [8] surveyed path planning involving computational intelligence. Several path planning approaches are inspired from ideas developed for classical robotics. For example, some of these use artificial potential functions [4], random trees [5] and Voronoi diagrams [6]. Path planning may be combined with team work and formation control [16]. Some approaches have been adapted to quadcopters [17]. Similarly to other path planning algorithms, the MAVs have a common goal, i.e., a location or coverage of an area. In contrast to the others, in our algorithm the MAVs do not know what the exact goal is, i.e., the covered area or final location, until the very end.

For landmark recognition, several authors use machine learning. Maravall et al. [18,19] use probabilistic knowledge-based classification and learning automata for the recognition of patterns associated to visual landmarks. In [20], classification rules are associated to probability weights that are adapted dynamically using supervised reinforcement learning. There is an adaptation process that is conducted using a two-stage learning procedure. In the first stage, a series of variables are associated to each rule, e.g., the variables associated to the construction of a landmark recognition classifier are constructed using images' histogram features, such as standard deviation, skewness, kurtosis, uniformity and entropy. In the second stage, a series of weights is associated to every variable whereby the weights are obtained by applying a reinforcement algorithm, i.e., incremental R-L [19,20], over a random environment. This results in a specific image classifier for the recognition of landmarks, which is then loaded to the MAVs.

Related to MAV formation control, [21] assumes that the signal propagation model of a MAV has the shape of a sphere and analyzes network capacity allocation in a MAV-based network infrastructure. Furthermore, it proposes a formation algorithm that determines the 3D geographic location of each MAV. In [22], it is shown how to operate a swarm by human piloting a MAV (the leader) while the remaining followers are autonomous. A solution is proposed to synchronize and orchestrate a swarm of MAVs, based only on ad hoc communications to position MAVs. Finally, in [23], the authors formulate the multi-UAV formation reconfiguration problem as an optimal control problem with dynamical and algebraic constraints. They provide a hybrid particle swarm optimization and genetic algorithm.

The approach presented in [9] uses an information sharing path planning algorithm for MAV swarms. The swarm is expected to conduct a mission, whose final destination and entire path are unknown to the MAVs in the swarm. In order to complete the mission, MAVs are requested to cooperate and exchange information to compute and determine a series of intermediate steps by resolving a discrete localization problem. At each step, the MAVs execute a geocaching-like algorithm using navigational devices (e.g., GPS receivers) and visual references to landmarks to unravel trajectory waypoints. Together, the MAVs build a shared-information path planning process. The information sharing approach in [9] has been extended in [24] by adding research ideas inspired from search with advice over graphs [25], rings and complete network [26].

3 Incremental Path Construction Algorithm

We have introduced a navigation algorithm for a swarm of n MAVs, where n is a positive integer greater than or equal to three [9]. The algorithm is one of its kind in that it adopts an information sharing and collaborative model. It also finds inspiration in the geocaching outdoor recreational activity. We describe in the sequel its main characteristics.

3.1 Constructing a Path

A navigation path is defined by a sequence of waypoints. At the start, each individual MAV does not know what the waypoints are and what the path consists of, although every MAV in the swarm is seeded with some information. While, they progress from the origin to the destination, each MAV in the swarm has specific tasks to achieve. The completion of each of these tasks provides complementary information. The MAVs share their information with the other members of the swarm. Together, they uncover the location of each waypoint of the path.

Formally, a path consists of k waypoints p_0, p_1, \dots, p_{k-1} , where k is a positive integer. The swarm starts from the point p_0 . Intermediate and the final waypoints p_1, p_2, \dots, p_{k-1} are initially unknown and determined one by one. For $i = 1, \dots, k - 1$, the MAVs collectively resolve a localization challenge. For each instance of this challenge, for $j = 1, \dots, n$, the j th MAV finds the position of a reference point q_j . Determining the reference point requires finding the precise location of a place. For instance, such a place can be a building with specific architectural characteristics, e.g., a church with a double bell tower. Before starting, the MAV is seeded with visual clues about that place. Using its camera, observational data is collected. Using this data, place recognition methods and a GPS, the precise position of the place is computed. The resulting position is a point q_j . Every MAV exchanges the location of q_j with the other members of the swarm.

Assuming the recognition of the places has been correctly conducted, all the reference points sit on the perimeter of an imaginary circle, see Fig. 1(a). The centre of this circle is set as the origin of a unit vector \mathbf{v} . The waypoint p_i is determined by the intersection of the supporting line of \mathbf{v} and perimeter of the circle, see Fig. 1(b). Using waypoint p_i as reference, the MAVs head in a direction defined by a direction vector \mathbf{d} to find the reference points of the next waypoint, see Fig. 1(c). A path is a succession of “chained” waypoints, see Fig. 1(d). Before the start of the procedure, for every waypoint, every MAV receives information that will be used to recognize its reference points, such as physical characteristics. The MAV is also given the vectors \mathbf{v} and \mathbf{d} , for every waypoint. At each step, there is a unique circle determined by three reference points or more. The representation of vector \mathbf{v} finds its inspiration in an outdoor recreational game called geocaching. It is hidden within the circle. An example is pictured in Fig. 2. The origin of \mathbf{v} is located at the focus of the circle. In this

example, the orientation of \mathbf{v} is revealed by the upstream flow of the river, a clue given beforehand to the MAVs.

This navigation algorithm requires recognition of places, such as buildings and rivers. For this aspect, we depend on the research about place recognition [12, 13]. Several methods have been developed, including approaches specific to building recognition [27] and applicable to recognition from the air [11, 28]. While many techniques are resting on visual data, place recognition can also rely on environmental sound [10] such as flowing water, vehicular traffic or school yard. Each of the place recognition methods is fallible and has a degree of precision. It is determined empirically through testing and experimentation. In [12], precision is defined as the ratio of true positives over the sum of true positives plus the false positives. For a given method, we define the error probability as one minus its precision. This leads to a fault model for our algorithms detailed in the next sub-section.

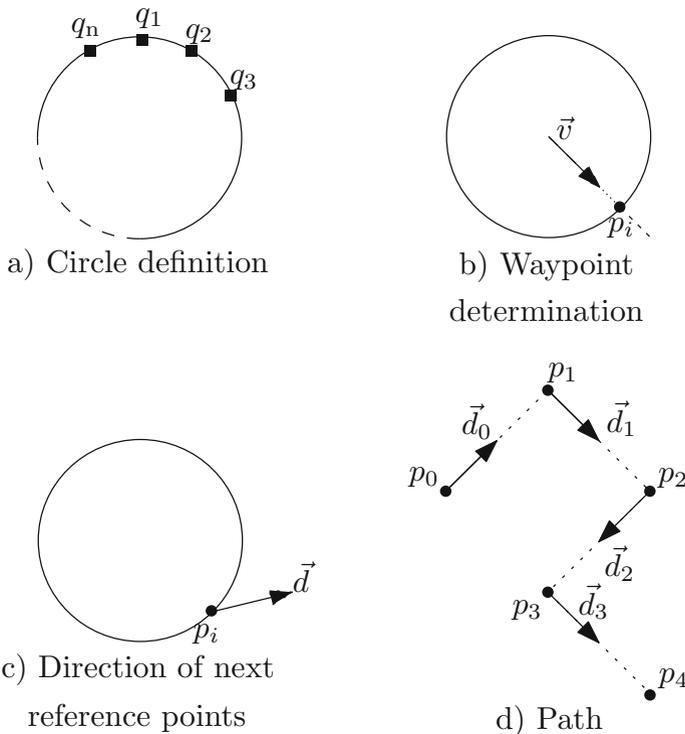


Fig. 1. Incremental path construction algorithm illustrated. Pictures (a), (b), and (c) constitute steps required to determine the next waypoint, while picture (d) depicts a resulting path.

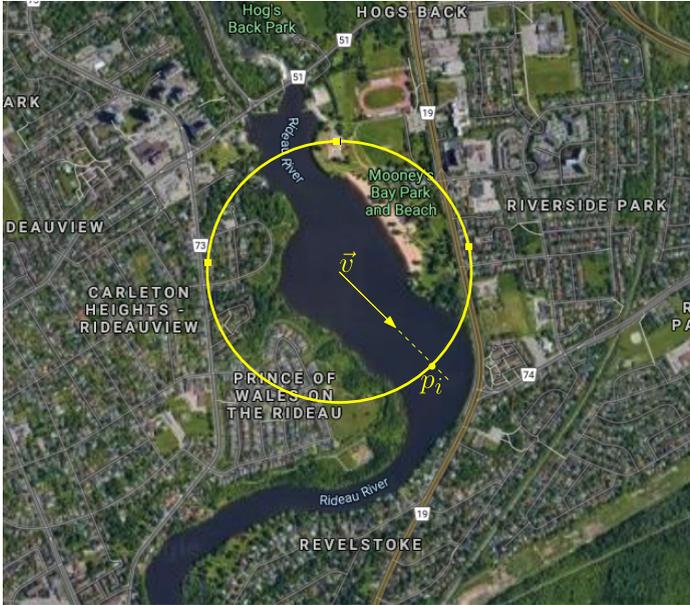


Fig. 2. Representation of vector v .

3.2 Fault Model

The algorithm originally presented in [9] solves failures in a partial manner. For instance, assume the two scenarios depicted in Fig. 3. Figure 3(a) consists of an arrangement of 11 MAVs with three faulty MAVs. Black dots represent reliable MAVs and black squares faulty MAVs. In this situation, the two non-faulty MAVs at the intersection of the two circles wrongly determine circle S' instead of circle S . If the decision to pick S or S' is according to the majority rule

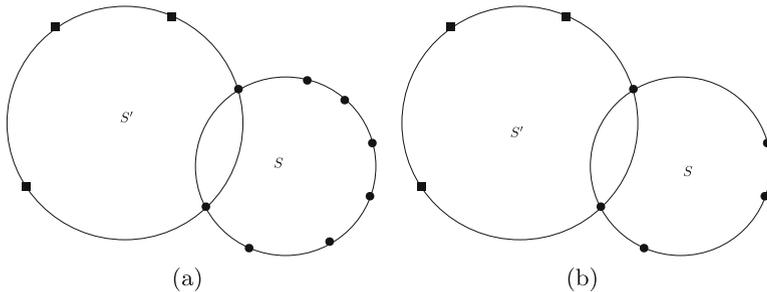


Fig. 3. Black dots represent reliable MAVs and black squares unreliable MAVs. (a) An arrangement of $n = 11$ MAVs with $f = 3$ faulty MAVs. (b) An arrangement of $n = 8$ MAVs with $f = 3$ faulty.

(six MAVs), then the whole swarm makes a correct decision (circle S). Figure 3(b) consists of an arrangement of eight MAVs with three faulty MAVs. Two circles are determined S and S' . The former being correct while the latter is incorrect. Two non-faulty MAVs, at the intersection of the two circles, are deceived by the three faulty MAVs. They participate to the determination of the incorrect circle (S' instead of S). If the decision to pick S or S' is according to the majority rule (five MAVs), then the whole swarm makes an incorrect decision (S').

In this paper, we address situations such as the ones depicted in Fig. 3. The goal is to increase the probabilities that a swarm of MAVs makes correct choices, accomplishes its mission and reaches the final destination. Errors are due to failures to recognize places correctly by each of the different methods. We assume that MAVs have the capability to concurrently apply several place recognition methods. The results of the applications of the different methods by the MAVs can be assembled together through communications and improved the decisional procedure. Building upon the algorithm introduced in [9] and place recognition methods, we introduce new algorithms. The recognition methods can be faulty. Hence, the MAVs can make errors and pick wrong reference points. For a given waypoint, they must compute together the center of a circle. In multiple-circle situations, ambiguity is resolved using a voting procedure and majority rules. Once a circle is determined and chosen, the swarm computes the center and waypoint. When a waypoint is correctly computed, the MAVs in the swarm makes a right move. In the following section, we formalize these ideas.

4 Decisional Framework

The MAVs of a swarm co-operate to accomplish a mission that consists of following a path with several waypoints. The discovery of each waypoint is achieved using a combination of visual and sound clues, observational data, place recognition methods and GPS information. This means that the MAVs process queries about place locations. The obtained replies are exchanged among the swarm members. Due to place recognition errors, replies may be inconsistent. Based on majoritarian rule, the replies are assembled together to determine a waypoint. This procedure is repeated for each step until the destination is reached. The resulting sequence of waypoints forms a path. In the sequel, we describe details of the decisional framework taking into account place recognition errors and disagreement between among swarm members.

4.1 Queries and Indicators

Let $N = \{1, 2, \dots, n\}$ denote the set of MAVs navigating a terrain. Let $I = \{1, 2, \dots, m\}$ be a set of indicators. Every indicator $i \in I$ may correspond to a place recognition method.

The MAVs need to navigate a terrain. They query available indicators. At a given location within the terrain, and for each MAV-indicator pair (u, i) a

query $Q(u, i)$ is made by MAV u . The obtained reply may be the precise GPS coordinates of a place. Besides, each reply may be wrong with probability p or correct with probability $1 - p$. The probability p may depend on the MAV u and/or indicator i being used. Further, we assume that the queries, and replies, are independent of each other, across the MAVs.

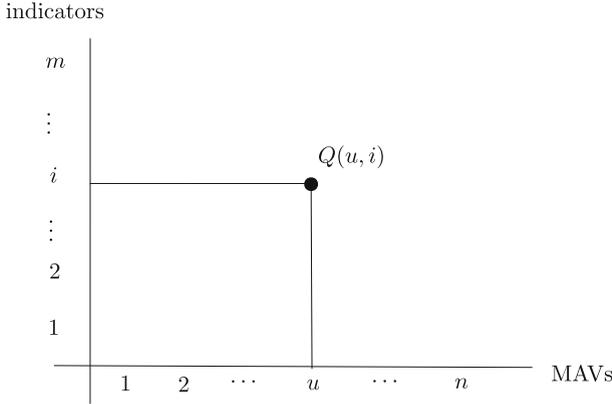


Fig. 4. Query $Q(u, i)$ made by a MAV u to indicator i .

As shown in Fig. 4, it is convenient to depict the query system in the (u, i) coordinate system with the u -axis representing the MAVs $u \in N$ and the i -axis the indicators $i \in I$. The place recognition methods associated with indicators may vary and depend on sensing and computing capabilities of the MAVs. In the sequel, it is assumed that all MAVs $u \in N$ may query exactly the same set of indicators, I .

4.2 Distributed Majority

There are two basic ways to reach a decision. Either every MAV processes queries individually or MAVs process queries together as a group. In the first way, all MAVs query a single indicator, say $i \in I$, corresponding to a single place recognition method. The MAVs exchange their replies, i.e., the reference points q_1, q_2, \dots, q_n , resulting from the application of the place recognition method. The replies may be inconsistent and define several circles. To resolve conflicts, the MAVs use a majoritarian rule to select a single circle focus. This decisional process is captured in Algorithm 1.

Algorithm 1. Single Indicator

- 1: Every MAV $u \in N$ queries a single indicator, resulting into points q_1, q_2, \dots, q_n .
 - 2: MAVs exchange reference points q_1, q_2, \dots, q_n
 - 3: All MAVs adopt the outcome of the swarm majority, resulting into a circle focus.
-

Let $p_n(i)$ denote the error probability of Algorithm 1, for n MAVs all using a single common indicator $i \in I$. It is given by the following formula.

$$p_n(i) = 1 - \sum_{l=\lceil n/2 \rceil}^n \binom{n}{l} [1 - p(i)]^l p(i)^{n-l}, \quad (1)$$

where $p(i)$ is the probability that a reply to a query is correct and $1 - p(i)$ that it is wrong, for the indicator i , for all MAVs $u \in N$. Algorithm 1 corresponds to the decisional logic in [9]. However, one can go beyond a single indicator. When all MAVs use all indicators in I , then they may exchange their individual majority decisions obtained across all indicators and follow the resulting swarm majority.

In a second individualistic approach, every single MAV in N uses all available indicators in I . It applies the standard majority rule to select a reply, i.e., a reference point. The reference points are exchanged among the MAVs. The group selects the circle determined by the majority of the MAVs. The decisional process is captured in Algorithm 2.

Algorithm 2. Multiple Indicators Hierarchical

- 1: Every MAV $u \in N$ queries all its indicators.
 - 2: Every MAV determines the indicator majority point q_u , $u \in 1, 2, \dots, n$.
 - 3: MAVs exchange reference points q_1, q_2, \dots, q_n
 - 4: All MAVs adopt the outcome of the swarm majority, resulting into a circle focus.
-

Let p_m denote the error probability of the majority rule for a single MAV using all m indicators,

$$p_m = 1 - \sum_{k=\lceil m/2 \rceil}^m \binom{m}{k} (1 - p)^k p^{m-k}, \quad (2)$$

where p is the probability that a reply to a query is correct and $1 - p$ that it is wrong, assuming that it is the same across all place recognition methods. We can also derive an equation without this assumption, indexing the probabilities over the indicators. Each MAV makes a decision independently querying all available indicators. The MAVs exchange their decisions and follow the resulting group majority. Assuming independence, the error probability of Algorithm 2 is given by the following formula.

$$p_{m,n} = 1 - \sum_{l=\lceil n/2 \rceil}^n \binom{n}{l} (1 - p_m)^l p_m^{n-l}, \quad (3)$$

In retrospect, there are advantages and drawbacks in Algorithms 1 and 2. On the one hand, in either case they are flexible in that one may select the indicators used in the majority decision and which are more suited to the terrain being

navigated or available communication possibilities and thus save in energy and communication costs, say. On the other hand, it should be emphasized that some indicators may not be available for querying to all MAVs because of environmental conditions, say, throughout the entire waypoint navigation process. This in turn may cause additional overhead on the MAVs that are required to maintain a database of active indicators per waypoint depending on environmental conditions.

Algorithm 3 combines Algorithms 1 and 2 in a comprehensive manner in that it adjusts and updates its set of MAVs and set of indicators employed in each waypoint. As such, it may yield more accurate results. A selected subset $N' \subseteq N$ of MAVs may query a selected subset $I' \subseteq I$ of indicators. They exchange all their replies. Because place recognition methods may disagree, a single MAV may generate several reference points. The circle determined by the majority of the replies is chosen.

Algorithm 3. Multiple Indicators Flat

- 1: All MAVs in N' query all indicators in I' , resulting into points $q_1, q_2, \dots, q_{|N'| \cdot |I'|}$.
 - 2: MAVs exchange reference points $q_1, q_2, \dots, q_{|N'| \cdot |I'|}$.
 - 3: All MAVs $u \in N'$ pick the majority among all outcomes, result is a circle focus.
-

Let $n' = |N'|$ and $m' = |I'|$. Let $p_{m',n'}$ denote the error probability of the majority rule for MAVs in N' and indicators in I' . It is given by the following formula.

$$p_{m',n'} = 1 - \sum_{r=\lceil m'n'/2 \rceil}^{m'n'} \binom{m'n'}{r} (1-p)^r p^{m'n'-r}, \quad (4)$$

where p is the probability that a reply to a query is correct and $1-p$ that it is wrong, for all the MAVs and all indicators. Note that Eq. (4) is the most general, for a fixed p . It subsumes Eqs. (1) and (2), as this can be seen by choosing $N' = N, I' = \{i\}$ and $N' = \{u\}, I' = I$, respectively.

An interesting point to make here is that the quality of the assembled answers depend on the reliability of the communications between the MAVs. Algorithm 3 is the most complete but also the most demanding in communication overhead while Algorithms 1 and 2 depend on more local conditions, to a certain extent.

4.3 Majority with a Threshold

The validity of Algorithms 1 and 2, and their generalization presented as Algorithm 3, hinges on the assumption that all the participating MAVs get replies to their queries. It may be the case that replies to queries are not obtained by a MAV due to harsh signal conditions (see Fig. 5). In this approach, a MAV participates in the majority algorithm when the number of obtained replies exceeds a certain predetermined threshold value.

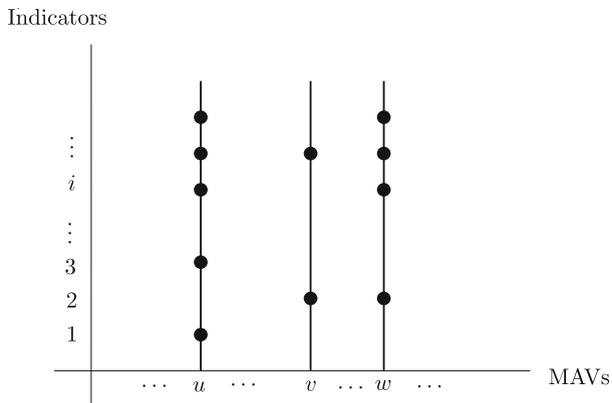


Fig. 5. Certain MAVs may not necessarily get replies for all the queries made. The MAVs u, v, w depicted are the ones for which the number of obtained replies exceeds a certain predefined threshold value.

More formally, for each MAV u let I_u be a random variable that counts the number of indicators in I that replied to a query made by u . It is natural to assume that I_u is a binomial random variable $B(q, m)$ with parameters q ($0 < q < 1$) and m , the number of indicators. The value of q may well depend on conditions of the environment as the MAVs are navigating waypoints. The binomial random variable $B(q, m)$ could be approximated with a Poisson random variable having arrival rate $\lambda = mq$ and probability of k arrivals equal to $\frac{\lambda^k}{k!} e^{-\lambda}$, where $m \rightarrow \infty, q \rightarrow 0$ and $mq \rightarrow \lambda$.

We now require that the vote of a MAV u counts towards the computation of the majority if $I_u > t$, where t is a fixed predetermined threshold value. Evidently in this counting, MAVs that obtain low numbers of replies are not taken into account in the final computation of the majority. The algorithm parametrized with a threshold value t generates a set of *VOTERS*, namely the set of MAVs that pass the threshold t . It is formalized as Algorithm 4.

Algorithm 4. Threshold Majority (t)

- 1: All MAVs in N query all indicators in I .
 - 2: MAVs with number of received replies exceeding threshold t become *VOTERS*.
 - 3: *VOTERS* exchange reference points.
 - 4: Majority decision is decided among *VOTERS*.
-

Algorithm 4 has the advantage that it takes into account only the MAVs that have the potential to provide the most accurate majorities based on the number of reliable indicators. Ultimately, it is a MAV that judges the appropriateness of a reply or even if a reply by a particular indicator is feasible given geometric and other communication factors. However, the choice of the threshold value t may be

critical. If it is too high, then the opinion of a large number of MAVs is excluded while if it is set too low the majority is cluttered with inaccurate answers. We can derive estimates on the threshold t using Chernoff bounds (see [29]). For example, we know that for $t > \lambda$, then we may have the formula.

$$\Pr[I_u > t] \leq \left(\frac{e\lambda}{t}\right)^t \cdot e^{-\lambda} \tag{5}$$

4.4 Numeric Results

Next, we discuss the numeric evaluation of Algorithms 1, 2, 3, and 4. Figure 6 have been obtained with implementation of the equations associated to each algorithm, Eqs. (1) to (5), using representative ranges of parameters. The x -axis represents a range for the indicator error probability. The y -axis plots the resulting error probability, for every algorithm. Variable m corresponds to the number of indicators. Variable n indicates the number of MAVs in the swarm. For Algorithm 4, variable v indicates the number of MAVs participating to the voting

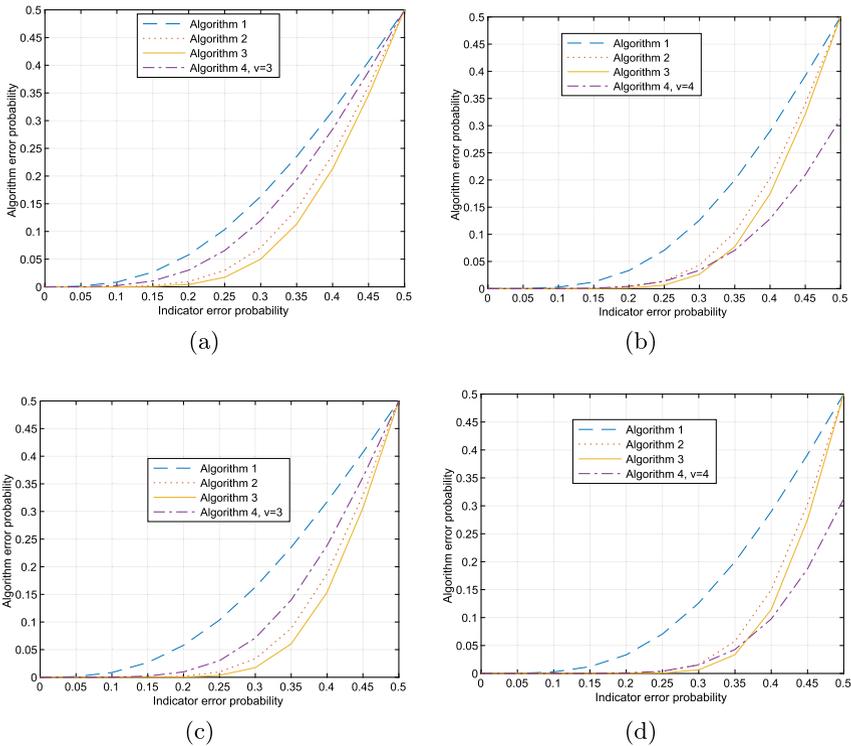


Fig. 6. Numeric results. Plots depict error probabilities, assuming (a) $m = 3, n = 5$; (b) $m = 3, n = 7$; (c) $m = 5, n = 5$; and (d) $m = 3, n = 5$.

process. The parameters associated to the indicators of each equation are based on experimental values reported in the literature, such as failure ratios expected from navigation systems, visual recognition, environmental sound and thermal infrared imaging [10–14]. As we can appreciate with the results, Algorithm 1 is always worst than all the others. For low indicator error probability, Algorithms 2, 3, and 4 have similar performance. The curves also indicate that a higher number of participating MAVs reduces the probability of errors. The same is true for the number of indicators.

4.5 Simulation Results

The numeric results reported in the previous section are extended in this section. We have implemented our decisional framework and algorithms in Java and conducted Monte Carlo simulations. In Fig. 7, the x -axis represents the number of members in a swarm. We show simulation results in which all four algorithms involve an incremental number of MAVs (from 10 to 90 MAVs), randomized failure ratios reported in the literature of navigation systems, visual recognition, environmental sound and thermal infrared imaging. The simulation aims at evaluating the chances of mission success of a swarm. Each simulation consists

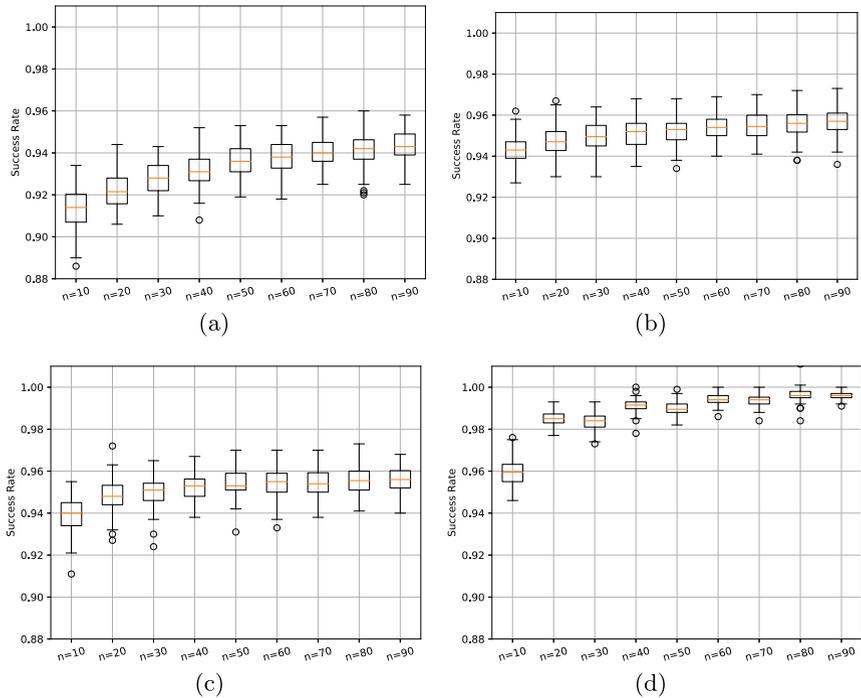


Fig. 7. Monte Carlo simulations results. (a) Algorithm 1, $m = 1$. (b) Algorithm 2, $m = 5$. (c) Algorithm 3, $m = 7$. (d) Algorithm 4, $m = 10$.

of 1000 iterations per x -axis value. The y -axis represents the success rate, i.e., number of successful iterations per x -axis value. Additional results are available in a companion website, online at [github](#).¹

5 Conclusion

This paper has extended an existing path planning algorithm for MAV swarms. The revisited approach assumes that MAVs collectively work to determine and solve a discrete location problem, in order to discover initially unknown intermediate landmarks. By collectively determining and sharing the landmark locations, the MAVs are able to complete their mission and reach their final destination. We assume that the process is prone to random failures. The original algorithm fails at handling situations where a series of faulty MAVs disrupt the collective procedure. It may affect the remaining MAVs to complete their mission. The new algorithms presented in this paper improve the original work. They are resilient to random failures.

The methodology proposed offers potential for additional research, exploration and testing more sophisticated majority strategies based on probability distributions that are realistic in that they are sensitive, for example to geographic location, proximity to base station and/or landmarks and available energy of the drones.

Acknowledgments. We acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC), the European Commission (H2020 SPARTA project, under grant agreement 830892), and the SAMOVAR research laboratory of Télécom SudParis.

References

1. Aerial Robots for Sewer Inspection, “ARSI”. http://echord.eu/essential_grid/arsi/. Accessed 2019
2. DHL: “DHL’s Parcelcopter: changing shipping forever”. <http://bit.ly/2WV7KcO>. Accessed 2019
3. Altigator: “Drones for search and rescue missions”. <https://altigator.com/drones-for-search-rescue-missions/>. Accessed 2019
4. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* **5**(1), 90–98 (1986)
5. LaValle, S.M.: Rapidly-exploring random trees: A new tool for path planning, Department of Computer Science, Iowa State University, Technical report (1998)
6. LaValle, S.: *Planning Algorithms*. Cambridge University Press, New York (2006)
7. Radmanesh, M., Kumar, M., Guentert, P.H., Sarim, M.: Overview of path-planning and obstacle avoidance algorithms for UAVs: a comparative study. *Unmanned Syst.* **06**(02), 95–118 (2018)

¹ <https://github.com/jgalfaro/mirrored-geomav>.

8. Zhao, Y., Zheng, Z., Liu, Y.: Survey on computational-intelligence-based UAV path planning. *Knowl. Based Syst.* **158**, 54–64 (2018)
9. Barbeau, M., Garcia-Alfaro, J., Kranakis, E.: Geocaching-inspired resilient path planning for drone swarms. In: *IEEE Conference on Computer Communications (IEEE INFOCOM 2019) Workshops*, pp. 620–625, April 2019. <https://doi.org/10.1109/INFCOMW.2019.8845318>
10. Chu, S., Narayanan, S., Kuo, C.J.: Environmental sound recognition with time-frequency audio features. *IEEE Trans. Audio Speech Lang. Process.* **17**(6), 1142–1158 (2009)
11. Gehrig, M., Stumm, E., Hinzmann, T., Siegwart, R.: Visual place recognition with probabilistic voting. In: *IEEE International Conference on Robotics and Automation (ICRA 2017)*, pp. 3192–3199 (2017)
12. Lowry, S., et al.: Visual place recognition: a survey. *IEEE Trans. Robot.* **32**(1), 1–19 (2016)
13. Galvez-López, D., Tardos, J.D.: Bags of binary words for fast place recognition in image sequences. *IEEE Trans. Robot.* **28**(5), 1188–1197 (2012)
14. Spaan, D., et al.: Thermal infrared imaging from drones offers a major advance for spider monkey surveys. *Drones* **3**(2), 34 (2019)
15. Otto, A., Agatz, N., Campbell, J., Golden, B., Pesch, E.: Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: a survey. *Networks* **72**(4), 411–458 (2018)
16. Turpin, M., Michael, N., Kumar, V.: Capt: concurrent assignment and planning of trajectories for multiple robots. *Int. J. Robot. Res.* **33**(1), 98–112 (2014)
17. Rizqi, A.A.A., Cahyadi, A.I., Adji, T.B.: Path planning and formation control via potential function for UAV quadrotor. In: *2014 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, pp. 165–170, June 2014
18. Maravall, D., de Lope, J., Fuentes Brea, J.P.: A vision-based dual anticipatory/reactive control architecture for indoor navigation of an unmanned aerial vehicle using visual topological maps. In: Ferrández Vicente, J.M., Álvarez Sánchez, J.R., de la Paz López, F., Toledo Moreo, F.J. (eds.) *IWINAC 2013*. LNCS, vol. 7931, pp. 66–72. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38622-0_7
19. Maravall, D., de Lope, J., Fuentes, J.: Navigation and self-semantic location of drones in indoor environments by combining the visual bug algorithm and entropy-based vision. *Front. Neurobot.* **11**, 46 (2017)
20. Narendra, K., Thathachar, M.: *Learning Automata: An Introduction*. Courier Corporation (2012)
21. Park, S., Kim, H., Kim, K., Kim, H.: Drone formation algorithm on 3d space for a drone-based network infrastructure. In: *IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–6, September 2016
22. Shrit, O., Martin, S., Alagha, K., Pujolle, G.: A new approach to realize drone swarm using ad-hoc network. In: *2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pp. 1–5, June 2017
23. Duan, H., Luo, Q., Shi, Y., Ma, G.: Hybrid particle swarm optimization and genetic algorithm for multi-UAV formation reconfiguration. *IEEE Comput. Intell. Mag.* **8**(3), 16–27 (2013)
24. Barbeau, M., Garcia-Alfaro, J., Kranakis, E., Santos, F.: Quality amplification of error prone navigation for swarms of micro aerial vehicles. In: *2019 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6. IEEE (2019)

25. Kranakis, E., Krizanc, D.: Searching with uncertainty. In: 6th International Colloquium on Structural Information & Communication Complexity, SIROCCO 1999, Lacaunau-Ocean, France, 1–3 July 1999, pp. 194–203 (1999)
26. Kirousis, L.M., Kranakis, E., Krizanc, D., Stamatiou, Y.C.: Locating information with uncertainty in fully interconnected networks. In: Herlihy, M. (ed.) DISC 2000. LNCS, vol. 1914, pp. 283–296. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-40026-5_19
27. Farfan-Escobedo, J.D., Enciso-Rodas, L., Vargas-Muñoz, J.E.: Towards accurate building recognition using convolutional neural networks. In: 2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), pp. 1–4 (2017)
28. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source slam system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robot.* **33**(5), 1255–1262 (2017)
29. Mitzenmacher, M., Upfal, E.: *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, New York (2017)