

A Vision-based Quadrotor Swarm for the participation in the 2013 International Micro Air Vehicle Competition

Jesús Pestana¹ and Jose Luis Sanchez-Lopez¹ and Paloma de la Puente²
and Adrian Carrio¹ and Pascual Campoy¹

Abstract—This paper presents a completely autonomous solution to participate in the 2013 International Micro Air Vehicle Indoor Flight Competition (IMAV2013). Our proposal is a modular multi-robot swarm architecture, based on the Robot Operating System (ROS) software framework, where the only information shared among swarm agents is each robot's position. Each swarm agent consists of an AR Drone 2.0 quadrotor connected to a laptop which runs the software architecture. In order to present a completely visual-based solution the localization problem is simplified by the usage of ArUco visual markers. These visual markers are used to sense and map obstacles and to improve the pose estimation based on the IMU and optical data flow by means of an Extended Kalman Filter localization and mapping method. The presented solution and the performance of the CVG_UPM team were awarded with the First Prize in the Indoors Autonomy Challenge of the IMAV2013 competition.

I. INTRODUCTION

The motivation of this work is the design of a solution to participate in the 2013 edition of the International Micro Air Vehicle Flight Competition (IMAV2013). The IMAV Flight Competition is the most relevant European competition in the fields of Autonomous Aerial Robotics and Small Remotely Piloted Air Systems (sRPAS). Our research group, the Computer Vision Group (CVG), was awarded³ for its performance in the 2012 edition of the IMAV competition[22] showing the potential of our group in the development of autonomous Unmanned Aerial Systems (UAS). The learning experience obtained from the indoor dynamics competition encouraged us to keep working in the same direction and also to try a swarming approach in the 2013 edition. Our motivation for participating in such competitions is to develop autonomous systems which can be later modified to perform civilian applications. The 2013 edition's rules are significantly different with respect to former edition's. In IMAV2013 there was only one indoor competition (see [3]) which requires a high level of autonomy. The scenario has some fixed and previously known obstacles (a wall and four fixed poles) and several obstacles located at unknown positions (two windows and four pole obstacles). The indoor competition includes

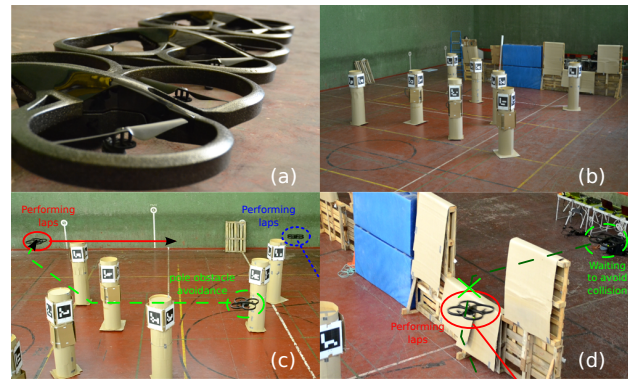


Fig. 1. The presented solution has been designed using AR Drone 2.0 quadrotors, see (a), and a replica of the IMAV2013 indoors challenge environment where the map and the obstacles were marked using ArUco markers [2], see (b). This environment, used during experimental flights, consists of a small window, a big window and 8 poles. The position of the wall is previously known except for the positions of the windows along it are unknown. The positions of the 4 corner poles are previously known, however, the positions of the 4 poles in the middle are unknown. The windows and poles represent obstacles that must be avoided during flight. The subfigure (c) shows a experimental flight where one of the drones is crossing the unknown poles area. The unknown poles are robustly located on previous laps, where the drone performs laps around the known poles, ensuring a good estimation of their positions. Our framework also allows to test partner collision avoidance during experimental flights, as shown in (d), where a drone is waiting until the path to cross the big window is clear. The flights shown in (c) and (d) are both explained in detail on Sec. IV. Videos and more information about our experimental flights can be found in the website <http://www.vision4uav.com/?q=node/386>.

various challenges, including flying through a window, flying through an obstacle zone, target detection and recognition, path following and precision landing, among others.

The second motivation for this paper is that there is a large variety of applications which require a robotic system to densely navigate within a wide area. Such applications can benefit from a swarming approach for the required data gathering of the problem at task, taking benefit from a multi-robot system. For instance, such an approach could be applied to security and surveillance tasks of middle sized areas.

After a deep analysis of the contest characteristics, a vision-based Quadrotor Swarm was selected as the best option to join the IMAV2013 indoor flight competition. A swarm composed by a significant number, 3 or more drones, of relatively simple quadrotors is used to achieve all navigation missions. Additionally, as we decided to work with a vision-based swarm, external visual markers are used to simplify the localization problem. Our swarm is fully

*The authors would like to thank the Consejo Superior de Investigaciones Científicas (CSIC) of Spain for the JAE-Predocctoral scholarships of two of the authors, and the Spanish Ministry of Science MICYT DPI2010-20751-C02-01 for project funding.

¹Computer Vision Group, Centre for Automation and Robotics, CSIC-UPM. {jesus.pestana, jl.sanchez, adrian.carrio, pascual.campoy}@upm.es. www.vision4uav.com

²Institut of Automation and Control, Vienna University of Technology. pdela Puente@acin.tuwien.ac.at

³<http://www.vision4uav.com/?q=imav12>

autonomous, and thus, the level of autonomy is categorized as “Autonomous Mission Control”, requiring a small number of operators to start and monitor the whole system. In case something goes wrong, an operator can stop independent agents of the swarm to prevent further malfunctioning.

The layout of the paper is the following. First, the modules of our architecture are described in sections II & III. Second, two experimental flights in a replica of the competition map are presented in section IV. And lastly, the future work and the conclusions are discussed respectively in the sections V & VI.

II. SYSTEM DESCRIPTION

The system is composed by a swarm of autonomous Unmanned Aerial Vehicles (UAVs), which in the case of our proposal for IMAV2013 are AR Drone 2.0s, see Fig. 1(a). Each drone is autonomous and can complete a previously defined navigation mission avoiding obstacles and collisions with the other drones of the swarm. There is no high-level intelligence that controls or synchronizes the drones. Therefore, the system has a “swarming” or cooperative behavior. All the drones in the swarm share their pose with the rest, so that the partner detection problem is solved. Taking into account the other swarm agents’ positions a free-collision trajectory for each drone is generated.

Thus, in our implementation for the participation in the IMAV2013, the swarm is composed by identical robotic agents, which consist of an AR Drone 2.0 and an instance of the software architecture. The characteristics of the AR Drone 2.0 are thoroughly explained in [4]. This drone has been used by other researchers achieving autonomous navigation in unstructured environments [11], [19]. The drones are commanded via WiFi from a ground station which is achieved using the `ardrone_autonomy` ROS package [1]. The communications between modules and swarm agents are implemented on a single network, where each agent runs several software modules which communicate using the Robot Operating System (ROS) software framework, see [14].

All drones’ ground computers are connected through a Local Area Network (LAN) and communicate using ROS. To keep the desired swarm behaviour the communication between robotics agents is limited to reporting only its own position to the rest of the swarm, which is used to avoid collisions. This approach simplifies the challenging problem of partner detection and localization, allowing the team efforts to focus on other parts of the design.

III. ARCHITECTURE SOFTWARE MODULES

In this section the main modules of each robotic agent in the swarm are described. Previous works of the authors in this architecture are described in [24]. A deeper explanation of the workings of the modules is explained in the partner article [25].

A. Pose Estimator

The “Pose Estimator” module estimates the position of the vehicle based on an altitude measurement, inertial and odometry measurements. This module is explained in the following article and Master’s Thesis [22], [21]. The altitude measurements are yielded by an ultrasound sensor. The inertial measurements that are taken into account are the orientation angles: roll, pitch and yaw. The utilized odometry measurements are the horizontal speed of the vehicle, which is obtained from the altitude measurement and ground optical flow calculations. These measurements are all part of the usual telemetry of the AR Drone 2.0, so that the drone’s do not need to be modified in any manner.

B. ArUco Eye

The odometry localization estimate is corrected with relative position measurements obtained from the usage of visual markers. We use external ArUco visual markers, which are shown in subfigures 1b & 1c, which are open-source, see [2]. There are two kind of ArUco markers defined in our system: the ones for which the position is previously known and the ones for which the position is unknown; which are respectively attached to known and unknown obstacles.

C. Localization and Mapping

Localization in indoor environments is a challenging task for UAVs, especially if a low cost and very lightweight solution is required [18], [23], [16], [12]. In the absence of GPS and laser sensors, visual approaches are very popular [23], [16], [12]. The method presented by Jayatilleke and Zhang [16] requires all the landmark poses to be known a priori and only works in limited areas, making use of quite a simple approach without filtering of any kind. The work by Faig *et al.* presents an interesting approach for local relative localization in swarms of micro UAVs, that requires to keep external markers always visible. Our method was mainly inspired by the work by Rudol [23], but our models and formulation are quite different from those proposed by Conte [6].

We designed and implemented an Extended Kalman Filter (EKF) that allows the complete 6 DOF pose of the drone to be corrected by integrating the odometry data and the information from the visual external markers detection. The localization method benefits from the existence of known landmarks, but it also incorporates unknown detected features, using a Maximum Incremental Probability approach for building a map of 6 DOF poses corresponding to ArUco markers positioned in the environment. Similar methods for ground mobile robots were developed in previous work by de la Puente *et al.*, initially based on the observation of 2D point features with a laser scanner [8] and later based on the extraction of planar features from 3D point clouds generated by a tilting laser scanner [10], [9].

D. Obstacle generator

Once the position of the unknown ArUco landmarks is obtained, they are processed in order to obtain higher level

geometrical features in 2D to be used as obstacles by the trajectory planner. For the IMAV2013, the position of the visual markers located on the following locations was preconfigured in the architecture: the windows and the four corner columns of the obstacle area. The position of the unknown obstacles was unknown and so the corresponding visual markers' position was not preconfigured in the architecture.

E. Trajectory Controller

The “Trajectory Controller” is able to follow yaw commands while executing a trajectory. This module is explained in the following article and Master’s Thesis [22], [21].

F. Trajectory Planner and Collision Avoidance

This module computes a free collision 2D trajectory (horizontal coordinates x and y) to reach the current mission point. Taking into account the other swarm agents’ positions a free-collision trajectory for each drone is generated by using a combination of state of the art trajectory planning algorithms: probabilistic road maps [7], a potential field map algorithm [17] and an A-Star algorithm [20].

The implemented collision avoidance strategy only uses the current position of the other robotics agents, which is a really simple approach. Modern, more involved, approaches which outperform this approach are explained in [13], [5], [26], [15].

G. Yaw Commander

The “yaw commander” module calculates the yaw reference depending on the current swarm agent’s mission. This module could potentially decide in which direction to look in order to explore or to get the most localization information from the environment.

H. Mission Planner

The Mission Planner utilized to participate in the IMAV2013 competition was basically sequential. Thus, the mission was divided in a sequence of tasks. Each task is monitored to determine its completion by this module. The completion event launches the execution of the next task in the sequence.

This module interacts with the trajectory planner module, the localization module and with the hypothalamus module. The hypothalamus module takes care of executing complex high level commands by interfacing with the rest of the architecture.

I. Hypothalamus

This module implements low-level intelligence such as: monitoring the state of the other modules of the swarm agent and presenting a simple interface between the mission planner and other modules. Some of the commands that the mission planner can achieve through the hypothalamus module are: setting up the whole system to an active flying behavior (including the start-up of all other modules and the trajectory controller); and also simpler commands such as take-off, land or hover.

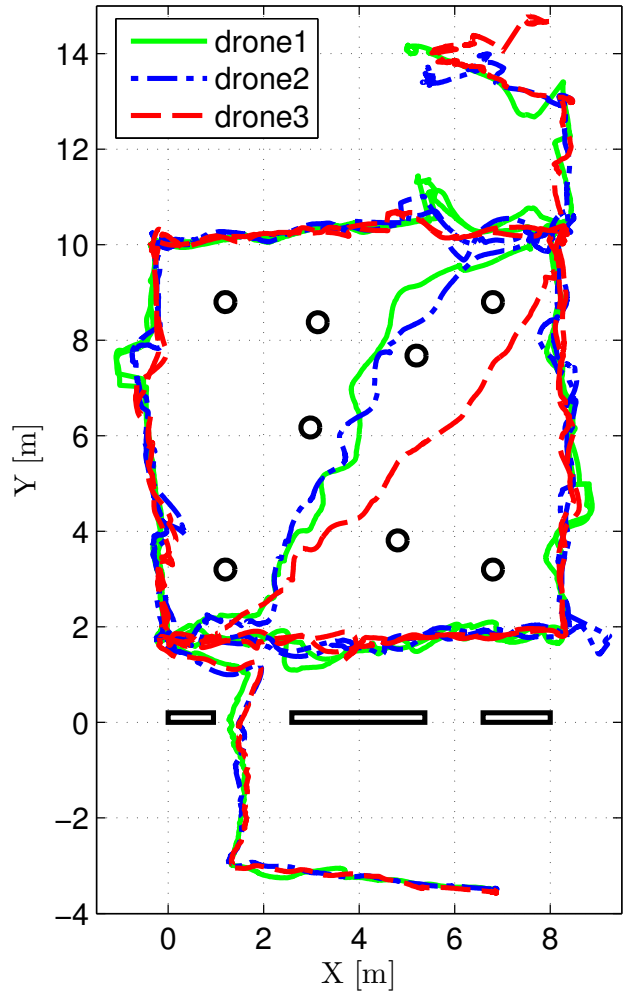


Fig. 2. Experimental flight where three drones flew simultaneously to perform navigation tasks in a replica of the IMAV2013 environment. The same flight is shown in Figs. 2 & 3. The poles are plotted as black circles, and the wall is plotted by three black rectangles, with the two free collision passages represented as windows. The AR Drone 2.0 is too big for the small window, so all the drones have to cross the big window. The trajectory executed by each drone is shown as a distinct line plot, whose color and line style are specified in the figure’s legend. In this experimental flight, all the drones managed to accomplish their respective missions. No navigation conflicts occurred thanks to the 15 seconds timing between the launches of each drone. A video of this flight can be found in the website <http://www.vision4uav.com/?q=node/386>.

This module also implements the communication between its swarm agent and the rest of the swarm. Currently the communication is limited to sharing each drone’s current pose.

IV. EXPERIMENTAL RESULTS

In this section we describe two experimental flights performed to showcase the capabilities of our solution, which was later demonstrated in the IMAV2013 competition. In order to interpret the flight figures, the module’s specifications have to be taken into account, which are described in the Architecture Software Modules section (Sec. II). From them, the overall expected behavior of the drones is:

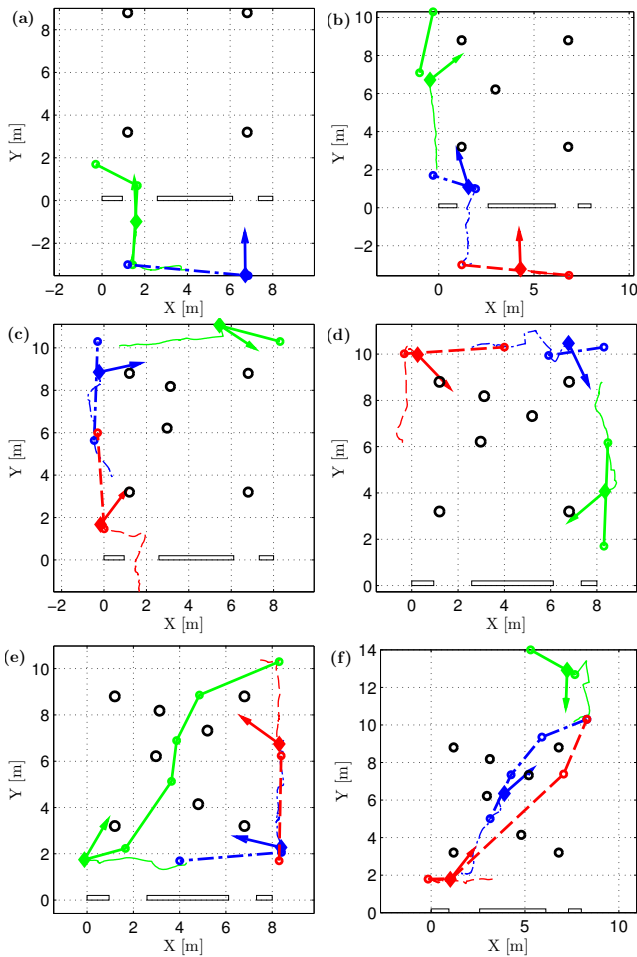


Fig. 3. Experimental flight where three drones flew simultaneously to perform navigation tasks in a replica of the IMAV2013 environment. The same flight is shown in Figs. 2 & 3. These subfigures, labeled a-f, are ordered in time, and each drone's plots have a different color and line style. The drones ordered by their launch times are plotted with: (first drone) green-solid lines, (second) blue-dash dotted lines and (third) red-dashed lines. The planned trajectory is shown with a thicker width than the actual executed trajectory. The dotted lines are the current trajectory references at the end of the plotted time period. The arrows indicate the direction where the drone is looking in the plotted position. As it is shown the drone is usually commanded to look at one of the four known corner columns of the map. (a), (b), (c) and (d) show how the first drone was able to cross the big window, and then performed laps around the poles. The second and third drones, initialized 15 and 30 seconds later in (a) and (b), performed the same navigation missions. The three drones are not synchronized by a high level intelligence/planner and a timed sequence of take-offs is beneficial for the overall mission execution. The unknown poles localized by the first drone are shown in these figures. (e) and (f) show the final stages of the mission where the drones finish their last lap and, afterwards, they have to cross the unknown obstacle area. They have to plan and execute an obstacle-free trajectory through this area. A video of this flight can be found in the website <http://www.vision4uav.com/?q=node/386>.

- Each drone follows a sequence of waypoints given by its mission specification. The mission planner executes it sequentially.
- During execution, the planner will attempt to find trajectories that are collision-free. If it fails, the drone will be controlled to stay in the current position.
- If other swarm agents enter the current trajectory, the

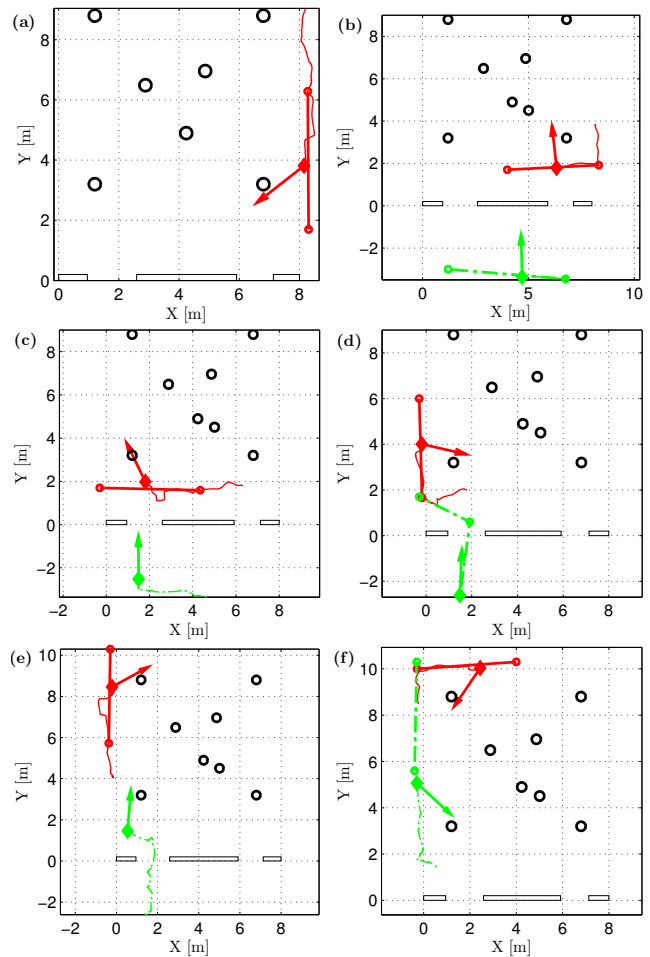


Fig. 4. Experimental flight where three drones flew simultaneously to perform navigation tasks. These subfigures, labeled a-f, are ordered in time, and each drone's plots have a different color and line style. The drones ordered by their launch times are plotted with: (first drone) red-solid lines and (second) green-dash dotted lines. The planned trajectory is shown with a thicker width than the actual executed trajectory. All subplots show how the first drone is performing laps over time. The second drone is launched on subfigure (b) has started the mission late. In Subfigs. (c) & (d) the first drone is blocking the window passage to the second drone, so it has to wait until the way is clear. The planner detects the first drone as an obstacle and thus, does not command a trajectory until the first drone is out of the way. As shown in Subfigs. (e) & (f), when the window passage is clear, the second drone crosses the window and starts performing laps. A video of this flight can be found in the website <http://www.vision4uav.com/?q=node/386>.

planner will detect it and will stop the drone and attempt to find a collision-free trajectory.

- If another swarm agent is on the current goal waypoint, the drone is commanded to stay in the current position. A new trajectory is planned when the goal position is again free.

This intelligence does not allow to solve every possible navigation conflict between the swarm agents. However, if the mission is specified correctly the conflicts will not occur often. For the flights preparation, the drones are ordered by their launch times. The mission specification used to run both missions is the same that was used during the IMAV2013

competition, which is the following sequence of tasks:

- 1) take-off and start the whole architecture,
- 2) move in front of the window and then move to one corner of the poles area (thus, crossing the big window),
- 3) perform laps around the poles area for 5 minutes,
- 4) cross the poles area moving towards the upper right corner of the map,
- 5) move to a final location and land.
- 6) the laps and the crossing through the unknown poles area tasks are specified so that all the drones will probably navigate in the same direction.

This mission specification minimizes the occurrence of navigation conflicts. The figures show the IMAV2013 environment in black. The estimated poses or executed trajectories and the current planned trajectories of each drone are shown with line plots.

The image shown in Subfig. 1(c) corresponds to the first experimental flight, which is shown in Figs. 2 & 3. More specifically the photo corresponds to the part of the flight shown between Subfigs. 3(e) & 3(f), when the drones are attempting to cross the unknown poles area. In the experiment, three swarm agents fly simultaneously performing various navigation tasks in a replica of the IMAV2013 environment. The launch of the drones was timed so that they performed their respective mission without any navigation conflicts. In this case all the drones were able to perform the full mission successfully, including the localization of the unknown elements of the environment and the obstacle avoidance tasks. This experiment was designed to showcase the capability of the swarm agents to fly relatively close to each other and perform the navigation mission when they are launched following the nominal schedule for the realization of the IMAV2013 Indoors Autonomy Challenge. More details of the execution of this mission can be read on the caption of the experiment figures.

The second experiment, shown in Fig.4, showcases the capability of each swarm agent to avoid collisions with other agents. Since there is no central intelligence in our swarm architecture, this capability is what allows the swarm agents to synchronize their navigation around the poles area and successfully accomplish the mission. A photo of this experiment is shown in Subfig. 1(d). As shown in Fig.4, the second drone is launched too late and enters a navigation conflict when it attempts to traverse the big window, because at that time the first drone is performing a lap just in front of the window. Then, the trajectory planner of the second drone determines that there is no free-collision path to cross the window and commands the drone to stay in position. Finally, when the first drone is no longer on the way, the second drone can proceed with mission execution when its planner determines a new collision-free trajectory to traverse the window. This intelligence is not able to perform well in every situation, however, the mission was specified so that the swarm agents would be able to perform the mission successfully most of the time.

Our experiments show that the swarm is able to navigate in areas with known obstacles, cross windows and navigate near obstacles. In addition, the swarm agents can cross parts of the environment where unknown poles were located, more specifically the middle area of the IMAV2013 environment, see Figs. 1 & 3. Also, as showcased in the experiment shown in Fig. 4, the drones know each other's position, and they are capable of mutual obstacle avoidance without the requirement of a high-level synchronization.

V. FUTURE WORK

Some ways to improve the presented architecture are the following:

- Design interface modules for other multirotors, which could carry more sensors other than a front-facing camera. This module would have to comply with part of the interface offered by the AR Drone such as its flying modes: take-off, taking-off, hovering, flying, landing, landed and emergency.
- Solve the localization problem using other visual-based techniques, such as stereo vision, mono-camera SLAM or SLAM based on the utilization of laser rangefinders.
- Design, implement and test different swarming behaviors.
- Design, implement and test more complex mission planners, which could detect better solutions to navigation conflicts during mission execution or that could handle different missions other than pure navigation.

VI. CONCLUSIONS

This paper presented an overview of a whole swarm system designed to autonomously complete the Indoors Autonomy Challenge of the IMAV2013 competition. The system is low-cost, employing Parrot ArDrone 2.0 quadrotors without any extra sensors, and the deployment and setup are quite easy and straightforward due to the fact that only a limited number of known external ArUco visual markers has to be put in place.

The ArUco markers are used for localization and mapping, improving the pose estimation obtained from IMU data and optical flow by means of an EKF based method. The resulting map of ArUco markers is converted to higher level 2D geometrical obstacles used by a trajectory planner combining probabilistic roadmaps, the potential field map algorithm and the A-Star algorithm. All the drones have access to the global position of every other drone in the team. The corresponding obstacles are incorporated to obtain a safe trajectory. A robust mid-level controller employs the target global position given by the trajectory planner and the corrected pose of each drone in order to drive them to their respective goals, defined by a mission planner module. The system design and implementation is based on ROS, which makes code sharing and module reuse easier.

This paper has two main contributions. The first contribution is to present our vision-based quadrotor swarm solution for the 2013 International Micro Air Vehicle Indoor Flight Competition, which was awarded with the First Prize

in the Indoors Autonomy Challenge. Our solution allowed to simplify the complexity of some modules such as the localization and partner detection capabilities. It was robust enough to work properly during the competition, but encountered problems due to its dependency on the WiFi links of the AR Drones 2.0. The second contribution is that the presented architecture has been made publicly available in the website: http://www.vision4uav.com/?q=quadrotor_stack. The modularity of the architecture allows to focus on the development of specific functionalities and test them along with the rest of the architecture. The authors hope that the public stack will benefit other developers in their research on swarming behaviors for small UAS among other topics.

REFERENCES

- [1] ardrone autonomy ros stack. https://github.com/AutonomyLab/ardrone_autonomy/.
- [2] Aruco: a minimal library for augmented reality applications based on opencv. <http://www.uco.es/investiga/grupos/ava/node/26>.
- [3] Imav 2013 flight competition rules. <http://www.imav2013.org/index.php/information>.
- [4] *The Navigation and Control Technology Inside the AR.Drone Micro UAV*, Milano, Italy, 2011.
- [5] Jur Berg, Stephen J. Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In Cdric Pradalier, Roland Siegwart, and Gerhard Hirzinger, editors, *Robotics Research*, volume 70 of *Springer Tracts in Advanced Robotics*, pages 3–19. Springer Berlin Heidelberg, 2011.
- [6] G. Conte. *Vision-Based Localization and Guidance for Unmanned Aerial Vehicles*. PhD thesis, Linkopings universitet, 2009.
- [7] R. Motwani D. Hsu, J.C. Latombe. Path planning in expansive configuration spaces. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 27192726, 1997.
- [8] P. de la Puente, D. Rodriguez-Losada, L. Pedraza, and F. Matia. Robot goes back home despite all the people. In *Proc. 5th. Conference on Informatics in Control, Automation and Robotics ICINCO 2008 Funchal, Portugal*, pages 208–213, 2008.
- [9] P. de la Puente, D. Rodriguez-Losada, and A. Valero. 3D Mapping: testing algorithms and discovering new ideas with USARSim. In *USARSim workshop, IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.
- [10] P. de la Puente, D. Rodriguez-Losada, A. Valero, and F. Mata. 3D feature based mapping towards mobile robots enhanced performance in rescue missions. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.
- [11] J. Engel, J. Sturm, and D. Cremers. Camera-based navigation of a low-cost quadrocopter. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.
- [12] J. Faigl, T. Krajník, J. Chudoba, M. Saska, and L. Přeučil. Low-Cost Embedded System for Relative Localization in Robotic Swarms. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2013.
- [13] Paolo Fiorini and Zvi Shillert. Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, 17:760–772, 1998.
- [14] Willow Garage. Ros: Robot operating system. www.ros.org/.
- [15] Daniel Hennes, Daniel Claes, Wim Meeussen, and Karl Tuyls. Multi-robot collision avoidance with localization uncertainty. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 147–154. International Foundation for Autonomous Agents and Multiagent Systems, 2012.
- [16] L. Jayatileke and N. Zhang. Landmark-based localization for unmanned aerial vehicles. In *IEEE International Systems Conference (SysCon'13)*, pages 448–451, 2013.
- [17] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic, 1991.
- [18] G. Mao, S. Drake, and B. D. O. Anderson. Design of an Extended Kalman Filter for UAV Localization. In *Information, Decision and Control, 2007 (IDC'07)*, pages 224–229, 2007.
- [19] Christian Mostegel, Andreas Wendel, and Horst Bischof. Active monocular localization: Towards autonomous monocular exploration for multirotor mavs. In *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA 2014)*, 2014.
- [20] B. Raphael P. E. Hart, N. J. Nilsson. A formal basus for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [21] Jesús Pestana. On-board control algorithms for Quadrotors and indoors navigation. Master's thesis, Universidad Politécnica de Madrid, Spain, 2012.
- [22] Jesús Pestana, Ignacio Mellado-Bataller, Jose Luis Sanchez-Lopez, Changhong Fu, Iván F Mondragón, and Pascual Campoy. A general purpose configurable controller for indoors and outdoors gps-denied navigation for multirotor unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, 73(1-4):387–400, 2014.
- [23] P. Rudol. Increasing autonomy of unmanned aircraft systems through the use of imaging sensors. Master's thesis, Linkoping Institute of Technology, 2011.
- [24] Jose Luis Sanchez-Lopez, Jesús Pestana, Paloma de la Puente, Adrian Carrio, and Pascual Campoy. Visual quadrotor swarm for the imav 2013 indoor competition. In Manuel A. Armada, Alberto Sanfeliu, and Manuel Ferre, editors, *ROBOT2013: First Iberian Robotics Conference*, volume 253 of *Advances in Intelligent Systems and Computing*, pages 55–63. Springer, 2013.
- [25] Jose Luis Sanchez-Lopez, Jesus Pestana, Paloma de la Puente, Ramon Suarez-Fernandez, and Pascual Campoy. A system for the design and development of vision-based multi-robot quadrotor swarms. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS 2014)*.
- [26] Jur van den Berg, Ming C. Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *IEEE International Conference on Robotics and Automation 2008 (ICRA 2008)*, pages 1928–1935. IEEE, 2008.