SwarmCity Project: Can an Aerial Swarm Monitor Traffic in a Smart City?

Juan Jesús Roldán, Pablo Garcia-Aunon, Elena Peña-Tapia and Antonio Barrientos
Centre for Automation and Robotics (UPM-CSIC)
Technical University of Madrid
José Gutiérrez Abascal, 2, 28006, Madrid, Spain
jj.roldan@upm.es, pablo.garcia.aunon@upm.es, elena.ptapia@alumnos.upm.es and antonio.barrientos@upm.es

***Abstract*—Smart Cities have emerged as a strategy to solve problems that current cities face, such as traffic, resource management, waste, pollution, etc. Most of the Smart City proposals are based on placing sensors in fixed locations of the city or, at the most, in public transportation systems. These strategies can produce blind zones, given that the sensors are fixed or their movement cannot be controlled. The SwarmCity Project proposes the use of an aerial swarm to monitor the city state, including traffic, crowds, climate and pollution. This paper is focused on traffic and tries to answer the question: "Can an aerial swarm monitor the traffic in a Smart City?". The work presents a data processing algorithm developed and optimized to fuse the data provided by the drones and build maps of traffic in real time. The proposed method is integrated with a surveillance algorithm and tested under different conditions in a city simulator. The results demonstrate the viability of SwarmCity Project and the potential use of aerial swarms as tools to collect data and model traffic in Smart Cities.**

***Index Terms*—Smart City, Robot swarm, Traffic monitoring, Data processing.**

I. Introduction

The growth of cities worldwide is an undeniable fact: the population of urban areas will increase from the current 54% to a 66% in 2050, according to United Nations reports [1]. This growth of cities can be accompanied by an improvement of efficiency in the management of resources, such as the distribution of goods, transportation and consumption of energy. Nevertheless, it may also cause traffic jams, security issues, pollution, noise, waste management issues and infrastructure deterioration. Acquiring, processing and visualizing data correctly will be key to achieve higher levels of efficiency and safety.

Smart Cities present a relatively new approach to deal with these challenges [2]. This idea involves the use of information and communication technologies to ensure a sustainable development of urban areas, optimize the management of available resources, improve the life quality of citizens and promote civic participation. A prototypical Smart City consists of many subsystems devoted to monitoring the city, understanding its state and predicting its evolution [3].

The most common solution for Smart Cities is based on the Internet of Things (IoT) and involves the deployment of a sensor network throughout the city [4]. Those sensors are normally fixed in specific places and, therefore, only able to collect data in their locations. This feature introduces bias in the data and reduces the robustness of the whole system. An explored solution for this issue is the integration of sensors in the public transportation systems, which allows to take measures in more locations, but the downside is that these sensor's positions cannot be controlled.

The SwarmCity Project proposes the use of a swarm of aerial robots to monitor the state of a city, collecting relevant data about traffic, pedestrians, climate and pollution. This project is being developed by the Robotics and Cybernetics Research Group of the Centre for Automation and Robotics, which is formed by the Technical University of Madrid and Spanish National Research Council. Why aerial robots? Because they are fast, agile and practical, and they have a limited impact on the daily life of citizens. Why a swarm? Because a group of simpler and lighter robots can perform a greater amount tasks, covering wider areas and consuming less time.

This paper explores the viability of the SwarmCity Project, trying to answer the following question: "Can an aerial swarm monitor the variables of a Smart City?". The work is focused on traffic monitoring, since this task is one of the most important challenges in Smart Cities and a relevant environment to validate the developed algorithms. However, it is expected that the results of traffic monitoring can be applied to other tasks, such as crowds monitoring, free parking space search, garbage detection and pollution monitoring.

In previous works, we developed a city simulator with realistic traffic, pedestrians, climate and pollution [5], as well as a behavior-based algorithm that allows the swarm to carry out search and surveillance tasks [6], [7]. In this work, a new algorithm has been developed and optimized to fuse the data collected by the drones and build a traffic map of the city. This algorithm has been integrated with the previous contributions, allowing for the first time to compare the information obtained from the swarm with the real state of the city. The results show the potential of aerial swarms as tools for collecting data and model traffic in Smart Cities.

Some works found in the literature propose to monitor cities using aerial robots, each one focusing on a certain part of the system: drones, sensors, infrastructures and communications. A framework for monitoring a Smart Cities with drones is presented in [8]. The work described in [9] is focused on planning the routes of aerial robots to optimize the task of traffic monitoring. An application of drones for traffic monitoring with experiments that test communications and video streaming is reported in [10]. Finally, the work described in [11] considers an aerial swarm for monitoring multiple variables of cities, focusing on the communications between the agents and the data processing.

The remainder of the paper is organized as follows: Section II describes the city simulator, focusing on the traffic model and its main variables. Section III explains the algorithm that allows the swarm to cover the city and monitor traffic. Section IV presents the algorithm developed and optimized for data fusion and information discovery. Section V reports the simulations performed to estimate the performance of this algorithm in various scenarios. Finally, Section VI summarizes the main conclusions of the work and the future steps of the project.

II. City simulator

The city simulator, also known as SwarmCity, reproduces a small European city with a central district, two residential neighborhoods, an industrial area, a big park and several public infrastructures (airport, train station...), as shown in Figure 1. Additionally, this simulator includes models of traffic, population, climate and pollution with variable behaviors. For instance, the temperature and humidity depend on the date and time, the levels of pollution are higher in industrial areas than in residential neighborhoods, and there are agglomerations of people in work places in weekdays and in leisure places during the weekends. SwarmCity has been developed using Unity Game engine and City Adventure, Road & Traffic System and Population Engine assets.

The most relevant part of SwarmCity for this work is the traffic model. The city consists of 20 streets with different lengths (from 50 to 800 meters) and shapes (straight and curves) and 22 intersections with different shapes (T intersections, cross intersections and roundabouts) and rules (priority and semaphores). At the beginning of the simulation, a certain number of cars, which can be set by the user, are spawned randomly on the roads. During the simulation, every car moves throughout the city at a certain speed that depends on the situation of traffic (light, heavy or jam), the type of section (straight, curve or intersection), and the speed limit of the road (set by the user). When a car arrives to an intersection, it decides randomly where to continue, and, depending on the situation, it moves immediately or waits until it has priority. As in real cities, the intersections tend to accumulate cars, which can lead to traffic jams. In addition, the user can generate traffic jams just by setting their locations and times.

Traffic models study the relationships between vehicles and infrastructures, seeking to understand transport systems in order to optimize their designs. These models combine theoretical and empirical techniques and usually use three variables: density, speed and flow [12]. In order to study the traffic, the roads of SwarmCity have been discretized in square cells of $S \cdot S$ m^2. Every detected car will be assigned to any of these cells, so all the traffic variables here estimated will be referred to these cells.

(a)

(b)

Fig. 1: SwarmCity: (a) Bird's eye view of the city, (b) A drone monitoring traffic in an intersection.

Traffic density is the number of vehicles per length unit that occupy a certain section of road at a given moment. As described by Equation 1, this variable (k) can be computed in our scenario as the number of cars detected in a cell (N) divided by the length of the side of the cell (S) and the number of lanes in the road (L).

$$k = \frac{N}{L \cdot S} \tag{1}$$

Traffic speed represents the distance covered per unit of time. There are two methods to obtain this variable from the instantaneous speeds of the cars: time mean speed and space mean speed. The first one takes into account the cars that go through a certain point in the road during a certain period of time. The second one takes into account the cars that are located in a certain section of the road at a certain moment. This second approach is easier to implement in our scenario, since the drones are continuously covering sections of roads.

The Equation 2 shows its computation: traffic speed (V) is the average value of the individual speeds (v_i) of the vehicles detected in a cell (N).

$$v = \frac{1}{N} \cdot \sum_{i=1}^{N} v_i \tag{2}$$

Traffic flow is the number of vehicles that crosses a section of road per unit of time. This variable (q) can be computed as the product of traffic density (k) and speed (v) as described by Equation 3.

$$q = k \cdot v \tag{3}$$

Figure 2 shows the traffic variable measurements after simulating in SwarmCity for eight hours: the pairs of flow and density values of multiple cars in different places and moments, together with the curves that represent the maximum and average values of flow for every density. This data is coherent with previous works of traffic modeling [13], since it includes multiple roads with different features. For instance, the flow in intersections will be less than in streets and in curved sections less than in straight ones, since the vehicles have to reduce their speed in these sections.

III. Swarm intelligence

The team of drones is led by an algorithm originally developed for search tasks in open environments [6], and later adapted for traffic monitoring in SwarmCity [5]. In order to organize the surveillance, the area is divided into cells, and the drones move between their centers. Similarly to the previous works, we consider that the drones fly at a height of 20 meters and, therefore, they can detect all the cars within a circle with a radius of 10 meters. Given that the amount of energy available in the batteries of the quadcopters is limited, 5 recharging bases are placed throughout the city. Each agent visits those bases with a period of 5-10 minutes and recharges its batteries during 60 seconds.

The algorithm works in a distributed fashion, that is, every drone shares specific information with the other agents and individually decides the next cell to visit. The algorithm is based on a network of 6 behaviors, see Figure 3, each of which was designed for a specific purpose and counts with a set of variables to be chosen.

Based on previous experiences optimizing the original algorithm [7], the surveillance adapted version was optimized with a genetic algorithm for the case in which there are 150 cars in the city, and the team of agents is made up with 10 drones. We observed that high variations of cars and drones do not impact on the performance of the algorithm.

IV. Traffic modeling

Let us consider N_D drones moving throughout the city and measuring traffic density and speed. Every drone will provide measurements of density k_d and speed v_d every time t at its location (x, y). The objective of the proposed algorithm is to fuse these measurements ($k_d(x, y, t)$ and $v_d(x, y, t)$) obtained by the drones ($d = \{1, ..., N_D\}$) and building maps ($K(t)$ and $V(t)$) as similar as possible to the ground truth measurements ($K_{ref}(t)$ and $V_{ref}(t)$). For this purpose, we propose three methods to create a map M at the time T from a set of measurements m_d obtained in previous instants $t < T$, where M and m_d can represent densities, speeds or any other variable of interest.

- Method 1: This method builds the map taking into account the most recent measurements of the drones. Initially, the map is a matrix with all the elements equal to -1. Then, when a drone obtains a measurement at a certain location, it is added to the map replacing the previous value. The result is a map that contains the most recent data for every cell, but may be vulnerable to the noise produced by traffic disturbances and sensing errors.
- Method 2: This method builds the map M computing the mean of the measurements M_t in a certain time window before the current time $t = T - W, ..., T$, as defined by:

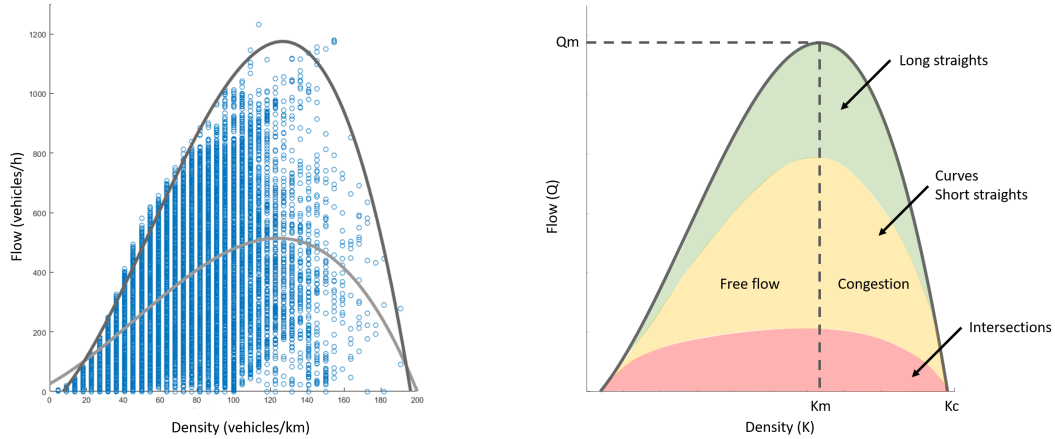
  $$M = \frac{1}{W} \cdot \sum_{t=T-W}^{T} M_t \tag{4}$$

  If the drones take some measurements at a point and, during the time window, they do not come back there, the mean of these last measurements remains in the map. The time window W is a parameter to be tuned in order to minimize the error between the estimated and real maps. This method provides more stability and robustness, but may take a long time to detect changes in traffic. As it can be checked, the method 1 is a particular case of the method 2 when $W = 1$.
- Method 3: As the previous one, this method builds the map M taking into account the measurements M_t in a certain time window W, and keeps the most recent measurements when there are not new ones in this period. However, this method applies a weighted mean to give more importance to recent measurements. As shown by Equation 5, the weights are generated through a negative exponential function whose behavior depends on a time constant T_c:

  $$M = \frac{\sum_{t=T-W}^{T} e^{-\frac{T-t}{T_c}} \cdot M_t}{\sum_{i=T-W}^{T} e^{-\frac{T-t}{T_c}}} \tag{5}$$

  In this way, two parameters (W and T_c) must be tuned to minimize the error between the estimated and real maps. This method is still robust against noise and it may adapt better to the changes. Method 2 is a particular case of method 3 with infinite T_c.

Eight one-hour long simulations were performed to obtain data to test the three methods with multiple parameters. In these simulations, a fleet of 10 drones monitored the city with 150 cars. We chose these values because the surveillance algorithm was optimized for them, and we wanted to optimize the data processing algorithm in the best scenario. Nevertheless, the surveillance algorithm showed good performance in other scenarios [5], and the data processing method will be also tested with other configurations, in Section V.



(a) (b)

Fig. 2: Flow-Density relationship: (a) Data obtained from simulations (blue circles), maximum (dark gray) and mean curve (light gray), (b) Explanation of the different areas below the curve.

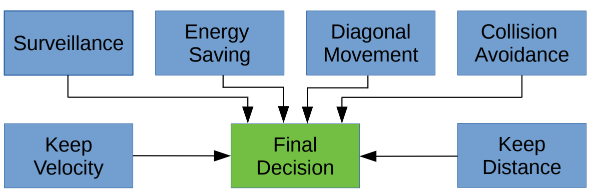


Fig. 3: Control scheme based on a set of behaviors presented in [6], originally developed for search tasks and later adapted to surveillance ones.

As it can be seen in Figure 4, four maps are obtained every second: two that show the real traffic densities and speeds in the city, and two that collect and process the values measured by the drones. As shown in Equation 6, we can define the error e as the difference between real and estimated maps (M_{ref} and M):

$$e = \sqrt{\frac{1}{N_P} \cdot \sum_{(i,j) \in R : M(P) \geq 0} (M_{ref}(P) - M(P))^2} \quad (6)$$

Only the points that belong to the roads and have been measured ($P \in R | M(P) \geq 0$) are taken into account, being N_P the number of points that meet these conditions. This variable is useful to measure the performance of data processing algorithms, but may produce disturbances when evaluating swarm control algorithms. For instance, a configuration that leads the drones to be static at certain points could produce less error than another one that allows a wider exploration of the map.

Two types of maps can be used as ground truth during the optimization: one with instantaneous and other with mean values. As shown in Figure 5, the first ones practically show the location and speed of every car or group of cars in the city at a given moment. However, the second ones are more useful to characterize the traffic in the city, since they reveal not only instant phenomena (such as traffic jams), but also permanent situations (e.g. average use of streets, bottlenecks, fastest and slowest routes...). Therefore, we use a time window of 30 seconds for the real maps hereinafter.

We applied the three methods with different parameters to the data obtained simulating 8 hours. Method 2 was tested with $W = \{10, 20, 30, 40, 50, 60, 90, 120, 150, 180, 210, 240, 300, 360, 420, 480, 540, 600\}s$, whereas Method 3 was checked with the same time windows W and $T_c = \{10, 50, 100, 500, 1000\}s$. The errors for densities and speeds obtained with the proposed methods and their dependence on the time windows are shown in Figure 6. As it can be observed, method 3 with $W > 150$ and $T_c = 10$ provided better results than the rest of methods and configurations. Hereafter, $W = 150s$ and $T_c = 10s$ are therefore selected.

V. EXPERIMENTS

Twenty simulations were performed to analyze the performance of the algorithm in multiple scenarios, that is, modifying the number of cars and drones. We considered fleets of 5, 10, 15 and 20 drones and cities with 50, 100, 150, 200 and 250 cars. As shown in Figure 7, the errors slightly increase with the number of drones, whereas their dependence on the number of cars is not clear. As mentioned above, visiting regularly a few points can be better than exploring the whole map taking into account the metric of error used in this work. This fact is consistent with the similarities between the graphics of covered area and density and speed errors. In future works, we are going to study both the application of other error metrics and normalization of the traffic variables to avoid this problem.

Additionally, four simulations were conducted to check if the algorithm was able to detect traffic jams. A traffic jam increases the density and, at the same time, reduces the speed in a certain area (see Figure 4). Thus, a simple method to detect

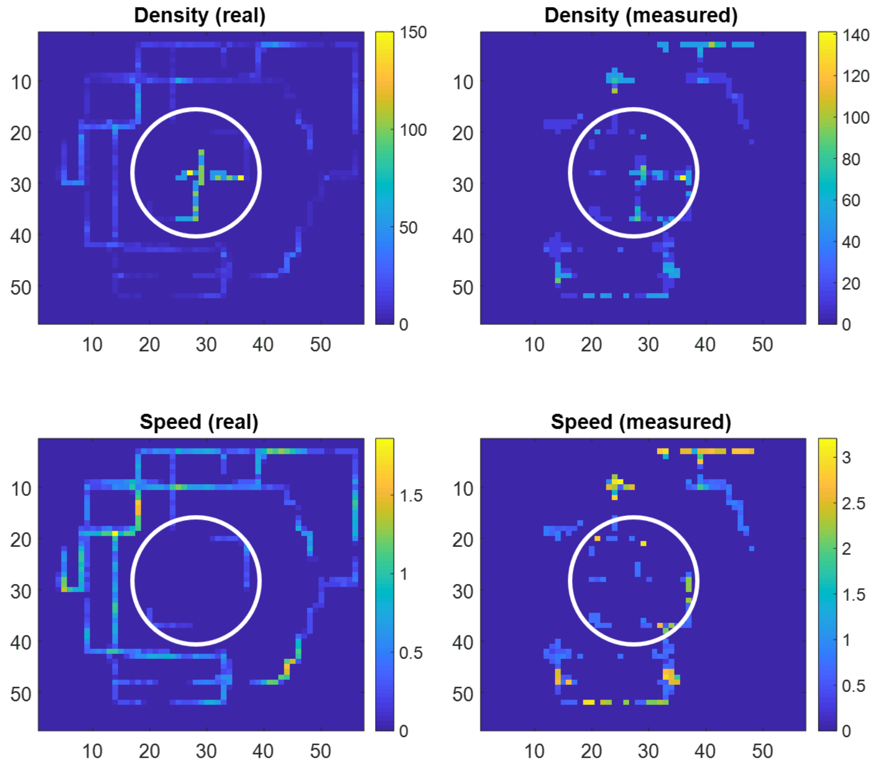


Fig. 4: Real and measured maps of traffic density and speed obtained during one of the simulations. The white circumference surrounds the area where a traffic jam is occurring.

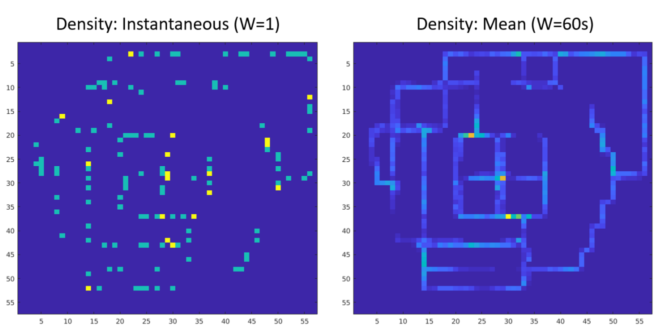


Fig. 5: Effects of applying a time window on a density map.

it is to compute these variables by regions and compare them. In the simulations, an accident is triggered at 120 seconds, the traffic jam can be observed in the real map at 232 seconds, and the algorithm can discover it at 538 seconds.

VI. Conclusions

This work analyzes the feasibility of aerial swarms as tools for monitoring traffic in Smart Cities. For this purpose, a new algorithm has been developed to fuse the data collected by the drones and build the traffic maps of the city. This algorithm has been integrated in a city simulator with a realistic traffic model and an aerial swarm controlled by a behavior-based surveillance algorithm. Then, a set of experiments have been performed to optimize the algorithm and test the whole system under different conditions.

The results show that the best strategy for data processing is computing a weighted mean of the last measurements and, specifically, using a time window $W = 150s$ and a time constant $T_c = 10s$. This algorithm provides similar results in the simulations with different numbers of cars, but its performance decreases when higher numbers of drones are managed. This fact can be explained because the bigger fleets explore wider areas, which is desired when monitoring cities, but increases the probability that the information in some points is not updated.

In future works, we are going to study the application of models that consider input noise such as Gaussian processes for data fusion and traffic modeling. Additionally, we want to adapt the algorithms to work with crowds, climate and pollution monitoring. Finally, we will develop an adaptive& immersive interface to allow users to monitor the city. The final goal of SwarmCity Project is to integrate the city simulator, swarm intelligence, data processing and advanced interface, and perform a real-time demonstration in which an operator commands the swarm and monitors the city.

Acknowledgments

This work is part of the “SwarmCity project: monitoring future cities with intelligent flying swarms”, developed by the

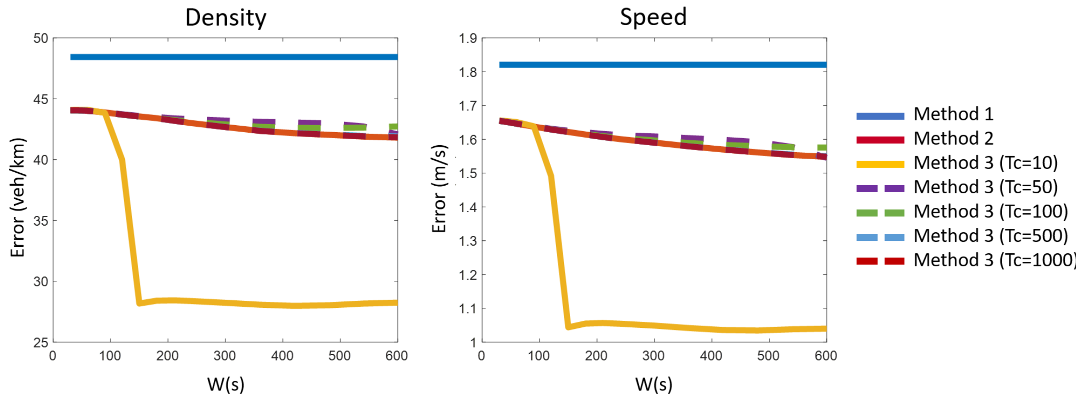


Fig. 6: Errors obtained between real and estimated maps using the three methods.

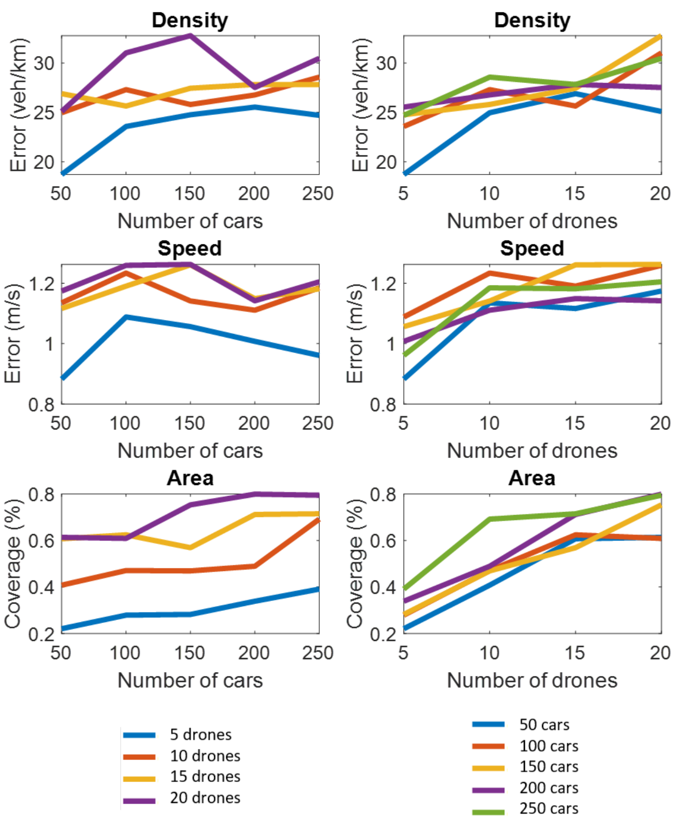


Fig. 7: Performance of the algorithm with different numbers of drones and cars.

Robotics and Cybernetics Research Group of the Centre for Automation and Robotics (UPM-CSIC).

The research leading to these results has received funding from the SAVIER (Situational Awareness VIrtual EnviRonment) project of Airbus Defence & Space; RoboCity2030-III-CM project (Robótica aplicada a la mejora de la calidad de vida de los ciudadanos. Fase III; S2013/MIT-2748), funded by Programas de Actividades I+D en la Comunidad de Madrid and cofunded by Structural Funds of the EU; and from the DPI2014-56985-R project (Protección robotizada de infraestructuras críticas) funded by the Ministerio de Economía y Competitividad of Gobierno de España.

References

[1] U. Desa *et al.*, "World urbanization prospects, the 2011 revision," *Population Division, Department of Economic and Social Affairs, United Nations Secretariat*, 2014.

[2] D. McLaren and J. Agyeman, *Sharing cities: a case for truly smart and sustainable cities*. MIT Press, 2015.

[3] L. Anthopoulos, "Smart utopia vs smart reality: Learning by experience from 10 smart city cases," *Cities*, vol. 63, pp. 128–148, 2017.

[4] R. Petrolo, V. Loscri, and N. Mitton, "Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 1, p. e2931, 2017.

[5] P. Garcia-Aunon, J. J. Roldán, and A. Barrientos, "Monitoring traffic in future cities with aerial swarms: developing and optimizing a behavior-based surveillance algorithm," *Cognitive Systems Research*, 2018.

[6] P. Garcia-Aunon and A. Barrientos Cruz, "Comparison of heuristic algorithms in discrete search and surveillance tasks using aerial swarms.," *Applied Sciences (2076-3417)*, vol. 8, no. 5, 2018.

[7] P. Garcia-Aunon and A. B. Cruz, "Control optimization of an aerial robotic swarm in a search task and its adaptation to different scenarios," *Journal of Computational Science*, 2018.

[8] H. Kim, L. Mokdad, and J. Ben-Othman, "Designing uav surveillance frameworks for smart city and extensive ocean with differential perspectives," *IEEE Communications Magazine*, p. 99, 2018.

[9] J. Y. Chow, "Dynamic uav-based traffic monitoring under uncertainty as a stochastic arc-inventory routing policy," *International Journal of transportation science and technology*, vol. 5, no. 3, pp. 167–185, 2016.

[10] S. Srinivasan, H. Latchman, J. Shea, T. Wong, and J. McNair, "Airborne traffic surveillance systems: video surveillance of highway traffic," in *Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks*, pp. 131–135, ACM, 2004.

[11] D. Wu, D. I. Arkhipov, M. Kim, C. L. Talcott, A. C. Regan, J. A. McCann, and N. Venkatasubramanian, "Addsen: Adaptive data processing and dissemination for drone swarms in urban sensing," *IEEE transactions on computers*, vol. 66, no. 2, pp. 183–198, 2017.

[12] F. Aparicio, B. Arenas, A. Gómez, F. Jiménez, J. López, L. Martínez, and F. Páez, "Ingeniería del transporte," *CIE Dossat: Madrid, Spain*, 2008.

[13] F. van Wageningen-Kessels, H. Van Lint, K. Vuik, and S. Hoogendoorn, "Genealogy of traffic flow models," *EURO Journal on Transportation and Logistics*, vol. 4, no. 4, pp. 445–473, 2015.