Image-based Road Network Clearing without Localization and without Maps using a Team of UAVs

Matteo Gagliardi, Giuseppe Oriolo, Heinrich H. Bülthoff, Antonio Franchi

Abstract— We address the problem of clearing an arbitrary and unknown network of roads using an organized team of Unmanned Aerial Vehicles (UAVs) equipped with a monocular down-facing camera, an altimeter, plus high-bandwidth short-range and low-bandwidth long-range communication systems. We allow the UAVs to possibly split in several subgroups. In each subgroup a leader guides the motion employing a hier-archical coordination. A feature/image-based algorithm guides the subgroup toward the unexplored region without any use of global localization or environmental mapping. At the same time all the entry-points of the the explored region are kept under control, so that any moving object that enters or exits the previously cleared area. Simulative results on real aerial images demonstrate the functionalities and the effectiveness of the proposed algorithm.

I. INTRODUCTION

The use of coordinated teams of robots allows to require less from each hardware device and at the same time achieve higher performance and robustness. For such reasons the design of controllers for multiple robots gained great relevance in the literature, see, e.g., [1], [2] and references therein. Particular interest has been paid to the exploration of a given region [1], [3], [4], allowing the execution of several subtasks: guaranteed clearing for pursuit-evasion [4], [5], models for forest area coverage [6], autonomous or semi-autonomous control for rescue purposes [7], mine detection [8], exploration models for underwater regions [9], just to mention a few. Along this line, in this work we present an algorithm for the guaranteed clearing of a network of roads using a team of UAVs.

The clearing problem is well established in the research on robotics: so far, however, there have been still few results suitable for real-world applications. In fact, a large fraction of such research work has developed models using assumptions far from those suitable for realistic applications, e.g., sensors with infinite range, a priori knowledge of the environment or target model, or perfect localization. Furthermore, From a theoretical point of view, important results have been achieved in this field using three main approaches: graph-based, visibility-based and sweeping research. In graph-based approaches the team is typically supllied with a

M. Gagliardi, H. H. Bülthoff are with the Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany. E-Mail: {matteo.gagliardi,hhb}@tuebingen.mpg.de. H. H. Bülthoff is additionally with the Department of Brain and Cognitive Engineering, Korea University, Seoul, 136-713 Korea.

G. Oriolo is with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Via Ariosto 25, 00185 Roma, Italy. E-mail: oriolo@dis.uniroma1.it

A. Franchi is with the Centre National de la Recherche Scientifique (CNRS) Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), 7 Avenue du Colonel Roche, 31077 Toulouse CEDEX 4, France. antonio.franchi@laas.fr
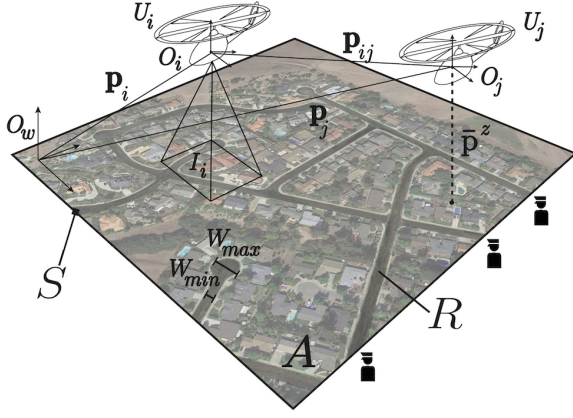
complete or partial knowledge of the environment to be explored [10]–[12]. In [11] the authors improve their previous works by proposing a clearing strategy that does not require a map of the environment. Using visibility-based approaches one can cope with the lack of a priori information about the environment, as, e.g., in [13] where a kind of communication chain through limited-range sensors with a perfect localization in the environment is created in order to report all the data acquired to a command center. An important contribution to the visibility-based approach is proposed in [5] in which the clearing is performed by terrestrial agents using sensors with limited range capabilities. Solutions using Simultaneous Localization and Mapping (SLAM) make possible to avoid a priori assumptions about the environment, to the detriment of the memory space and computational load. Another relevant work is [14], where the searching problem is addressed using a *variable resolution* data structure and field experiments are also presented. In [4], [15] all the previously adopted assumptions are removed at once: each agent uses limited-range planar sensors and shares information about its own local exploration frontiers with its neighbors.

Almost all the aforementioned works developed algorithms that are suited for ground robots. However, using UAVs instead of ground robots makes possible the inspection, monitoring and search of large areas that may be inaccessible by terrestrial vehicles. Multi-rotor UAVs have, in particular, gained a lot of attention in the community [16], [17]. Along this line, [18] presents an algorithm to search for ground targets using multiple UAVs employing the technique of sweep lines.

Here, we propose a novel vision-based outdoor clearing algorithm for a group of UAVs, equipped with limited-field-of-view down-facing cameras, that relies neither on global localization nor on a mapping of the area to be cleared. The method call for each UAV to store only the current image and the previous one in order to work properly, thus requiring limited computational capabilities and memory space, a feature that is mandatory, e.g., in the so called nano-UAVs.

The paper is organized as follows. In Sec. II a formal definition of the scenario is provided. A theoretical explanation of the clearing strategy is exposed in Sec. III. Sec. IV describes the image processing part of the algorithm and Sec. V presents the control architecture adopted for the coordination of the whole team. Finally, Sec. VI shows simulation results with a group of 8 quadrotor UAVs and Sec. VII concludes the paper with some final remarks.

Fig. 1: Main formalism for our scenario.

II. PROBLEM SETTING

Consider a limited ground area A and denote with $R \subset A$ the set of points of A that belong to the roads present on A (see Fig. 1 for a graphical representation of the scenario). The boundaries of A and R are denoted with ∂A and ∂R respectively. Consider then a team of n UAVs denoted with $U = \{U_1, U_2, \dots, U_n\}$. At t_0 all the UAVs are placed behind a starting line located on the intersection between ∂A and ∂R and denoted with $S \subset \partial A \cap \partial R$. Denote with $R_S \subset R$ the largest connected component of R that includes S. Consider an ENU (East-North-Up) inertial reference frame, denoted with $\mathcal{W}: O_w - \vec{x}_w, \vec{y}_w, \vec{z}_w$. We assume that the aerial occlusions caused by the buildings can be neglected, i.e., the ground environment, as seen from the UAVs, is 'sufficiently flat'.

Each UAV is equipped with a rigidly attached down-facing monocular camera having finite resolution $L \times L$ and both the vertical and horizontal field of view equal to $\alpha \in (0, \pi)$. An NED (North-East-Down) frame, denoted with $\mathcal{R}_i$: $O_i - \vec{x}_i, \vec{y}_i, \vec{z}_i$, is attached to the camera of each UAV U_i. The origin O_i is located at the focal center of the camera and the $\vec{z}_i$ axis is directed along the focal axis. The position of O_i in $\mathcal{W}$ is denoted by $\mathbf{p}_i = (p_i^x, p_i^y, p_i^z) \in \mathbb{R}^3$ and $\mathbf{R}_i^{\mathcal{W}}$ denotes the rotation matrix representing the orientation of $\mathcal{R}_i$ with respect to $\mathcal{W}$. We assume that each UAV can store only its current and previous image (see the discussion in Sec. I).

The UAV system is also endowed with a decentralized communication infrastructure that provides two different channels. A High-Bandwidth Short-Range (HBSR) channel is used to transmit the acquired image to every robot that is closer than D_{max}^{HBSR}. A Low-Bandwidth Long-Range (LBLR) channel is used to transmit brief messages to every robot that is closer than $D_{max}^{\text{LBLR}} \gg D_{max}^{\text{HBSR}}$.

Each UAV is equipped with a flight controller that guarantees the following conditions during the whole mission:

- altitude of the camera is kept at a given constant desired value $\bar{p}^z$, i.e., $p_i^z = \bar{p}^z$;
- the focal axis of the camera $\vec{z}_i$ is kept parallel to $\vec{z}_{\mathcal{W}}$;
- the $\vec{x}_i$ axis is kept parallel to $\vec{x}_{\mathcal{W}}$;
- the horizontal velocity $(\dot{p}_i^x, \dot{p}_i^y)$ is regulated to a value $\mathbf{u}_i \in \mathbb{R}^2$ that is assigned by the clearing algorithm described in the following section.

The four requirements on the flight controller can be easily achieved using, e.g., a barometer for the regulation of the altitude, the IMU measurements and a magnetometer for the regulation of $\vec{z}_i$ and $\vec{x}_i$, respectively, and a velocity sensor (e.g., based on the optical flow plus the barometer to recover the scale) for the regulation of the horizontal speed.

Being the image acquisition a discrete process, we consider a discretized time, denoting with k a generic time step and with I_i^k the image acquired by the i-th UAV camera at k. We assume that a specific algorithm is available to extract the road in I_i^k that is designed and tuned for the environment at hand. Such algorithm is able to extract the roads in I_i^k provided that the road width, expressed in pixels, is always contained in $[W_{min}^L, W_{max}^L]$, where $W_{min}^L > 0$ and $W_{max}^L < L$.

In order to obtain a well-posed problem setting, it is needed that:

- the altitude, the image resolution, and the road width allow a reliable segmentation, i.e.:

$$\frac{W_{min}}{2\bar{p}^z \tan(\alpha/2)} \geq \frac{W_{min}^L}{L}, \qquad \frac{W_{max}}{2\bar{p}^z \tan(\alpha/2)} \leq \frac{W_{max}^L}{L}, \quad (1)$$

where W_{min} and W_{max} are maximum and minimum width of the roads in A;
- D_{max}^{LBLR} is larger than the *diameter* of A, i.e., the maximum distance between any two points of A.

The first assumption guarantees a successful road extraction while the second one guarantees that two UAVs can always communicate using the LBLR channel. Finally we assume that the lines where the road network crosses ∂A apart from the starting line S are constantly patrolled by some *guards*. In the case that S is the only access to A this assumption is not needed.

Notice the following facts:

- the set R is unknown (no map);
- each UAV can use only the current and previous image;
- each UAV ignores both its latitude and its longitude (no localization).

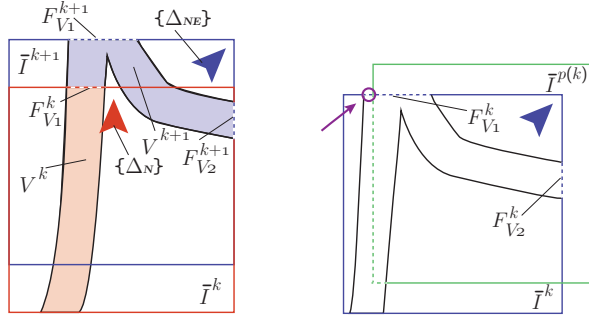We can now formally define our objective.

Clearing Problem: An unknown number of moving agents (e.g., vehicles) travels on the roads of R. Their motion is continuous but their initial position is unknown and their speed is unpredictable and unbounded. An agent is assumed to be detected only if it falls in the field of view of a camera or if it is spotted by a guard (i.e., if it crosses ∂A).

The goal for the UAV team is to move and deploy over the area A at constant altitude $\bar{p}^z$ so as to guarantee that any moving agent that starts moving in R_S at t_0 is, in a finite time, detected. The UAVs enter the scene from the same starting line, and in a way such that S is covered by the camera of at least one of them.

III. CLEARING STRATEGY

A. Preliminary Results

In order to precisely describe our clearing strategy we start by formalizing the problem from a set theory point of view. Define as $U_C^k \subset U$ the subset of UAVs whose images at time step k are actually used for the clearing task in a way to be described below. We call these UAVs *cleaners*. The set U_C^k might be the whole U or just one UAV. The reason for

(a) Selection of a motion candidate: at step k the primitive Δ_N is selected because the frontier F_V^k is located on the north side of the image $\bar{I}^k$; at step $k+1$ the motion Δ_{NE} is instead selected since the frontier F_V^{k+1} spans both the north and the east sides

(b) Motion candidate validation: Δ_{NE} has been selected as possible motion at step k since F_V^k spans both the north and the east sides. However, by prediction it is clear that such a movement will not ensure $F_V^k \subset \bar{I}^{k+1}$, and it is therefore not executed.

Fig. 2

introducing U_C^k is that at time step k not all the UAVs provide an image which is used to clean the environment. Denote with $\bar{I}^k$ the union of all the points of the plane $<\vec{x}_w, \vec{y}_w>$ that are visible by the cleaners at time step k, i.e., $\bar{I}^k = \cup_{i \in U_C^k} \bar{I}_i^k$.

Consider a ground agent, and denote with T^k its *trail*, i.e., the area swept by the agent during its motion from time step $k-1$ to time step k. The ground agent is *detected* at k if

$$T^k \cap \left(\partial A \cup \bar{I}^k \right) \neq \emptyset, \qquad (2)$$

i.e., if T^k either crosses ∂A or is visible (either entirely or partially) by one of the cleaners at time step k.

A region $R_S' \subset R_S$ is *cleared* at time step k if each agent in that region has been detected, i.e., if for every agent

$$T^k \cap R_S' \neq \emptyset \quad \Rightarrow \quad \exists l \leq k : T^l \cap \left(\partial A \cup \bar{I}^l \right) \neq \emptyset,$$

The *clearing problem* can then be formulated as clearing the whole set R_S.

Instrumental to the description of the clearing strategy is the introduction of V^k, the *connected visited subset* of R_S, defined as

$$V^k = Conn \left\{ S, R_S \cap \left(\cup_{l=1 \ldots k} \bar{I}^l \right) \right\} \qquad (3)$$

i.e., the maximal connected subset of R_S that includes S and has been covered by the cleaners up to time k. We define the *frontier* of V^k as

$$F_V^k = \partial V^k \setminus \partial R_S = \partial V^k \cap R_S^\circ, \qquad (4)$$

i.e., the points of the boundary of V^k that are not contained in the boundary of R_S (i.e., are part of the interior of R_S). Figure 2a depicts the sets defined so far in a typical case. Notice that the symbol $\circ$ denotes the *interior* of a set.

The entire F_V^k is in general the union of several connected components, i.e., $F_V^k = \cup_{j \in F_V^k} F_{V\,j}^k$, where each component $F_{V\,j}^k$ is referred to as a *sub-frontier* of F_V^k.

The following result (based on the "pushing back frontiers" concept [4], [15]) is the main principle around which our strategy is built.

Proposition 1. *If V^k is cleared and*

$$F_V^k \subset \bar{I}^{k+1} \qquad (5)$$

then V^{k+1} is also cleared and $F_V^{k+1} \subset \partial \bar{I}^{k+1}$. If, additionally, $F_V^{k+1} = \emptyset$, then R_S is cleared at time $k+1$.

The formal proof of the previous proposition is omitted here for brevity. Proposition 1 implies the following results:

Corollary 1. *Any strategy ensuring that*
1) *condition (5) holds at every $k \geq 1$, and*
2) *there exists a finite K such that $V^K = R_S$,*

solves the clearing problem.

The second property of Corollary 1 implies that if $F_K^1 \neq \emptyset$ then

$$F_V^k \subset \bar{I}^{k+1^\circ} \qquad (6)$$

holds a number of times $m > 1$.

B. Image-based Clearing Strategy

Assume that by suitable image processing we are able to identify the portion of V^k in each $\bar{I}^k$ and its frontier F_V^k. We shall give more details on how to implement such image processing in Section IV. Furthermore, in order to ease the description of the clearing algorithm, we temporarily assume that the commanded motion of the UAVs is perfectly executed. In Section IV we will show how this last assumption can be relaxed.

Our clearing method is based on Corollary 1. Assume that each cleaner can either stay still or move, at constant speed, along the 8 cardinal and ordinal directions, thus obtaining 9 motion primitives on the horizontal plane. Denote with D the set of 9 possible horizontal displacements that are obtained staying still or moving in one of the possible directions between two consecutive instants, i.e., $D = \{\Delta_0, \Delta_N, \Delta_{NE}, \Delta_E, \Delta_{SE}, \Delta_S, \Delta_{SW}, \Delta_W, \Delta_{NW}\}$ where Δ_0 denotes the null displacement.

At time step $k=1$ the group starts with $U_C^1 = \{U_1\}$ and U_1 located on top of the starting line S. Let $D_1^1 \subset D$ be the set of all the displacements of U_1 such that (5) holds at time step $k=1$. At the same time, it is easy to see that the following properties hold

$$S \in \bar{I}_1^1, \quad V^1 \subset \bar{I}_1^1, \quad F_V^1 \subset \bar{I}_1^1.$$

Proposition 2. *If U_1 moves in $D^1 \setminus \{\Delta_0\}$ then*

$$F_V^1 \subset \bar{I}^{2^\circ} \qquad (7)$$

A motion primitive candidate $\hat{\Delta} \in D$ is selected taking into account the location of F_V^1 on the 4 sides composing $\partial \bar{I}^1$ as follows:

- if F_V^1 is entirely contained in one of the sides, the motion candidate is the displacement in the cardinal direction of the side itself, e.g., if F_V^1 is entirely contained in the North side of the image then the candidate will be Δ_N (see Fig. 2a);
- if F_V^1 spans three sides of the image, the candidate motion is the displacement corresponding to the middle
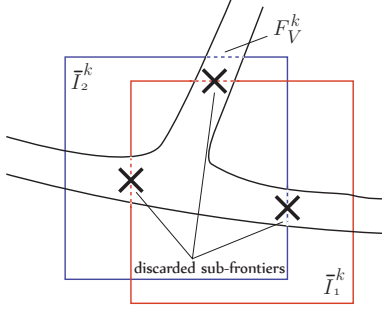
Fig. 3: Encounter between two cleaners U_1 and U_2: each cleaner deletes from its frontier the potions that fall into the other's image. U_1 remains without frontiers and the frontier of U_2 is reduced to the only northern part.

(a) Main notation of the image-based process.

(b) Image-processing for a real image using the margin box ∂M.

Fig. 4

direction, e.g., if F_V^1 is contained in the West, North, and East sides then the candidate will be Δ_N;

- if F_V^1 spans two contiguous sides, the motion candidate is the displacement toward the ordinal direction between the two sides; e.g,, if the F_V^1 is contained in the North and East sides the candidate will be Δ_{NE} (see Fig. 2a);
- if F_V^1 spans only two non-contiguous sides or all four sides of the image, the candidate is Δ_0.

In order to check the feasibility of the motion candidate U_1 performs a prediction of the location of F_V^1 in $\bar{I}^{p(1)}$ after the execution of the motion candidate. The motion candidate is *feasible* if it does not violate (5). In case of feasibility U_1 actually executes the candidate primitive.

If the motion candidate is not feasible (see Fig. 2b) then the null motion primitive Δ_0 is selected. Such situation implies that one or more UAVs have to be included in U_C^2 to continue the clearing. To this end, the frontier F_V^1 is shrunk and partitioned in sub-frontiers. Then U_1 assigns some of the sub-frontiers to other UAVs until its motion candidate results feasible. The UAVs who receive an assignment are then promoted to cleaners and included in U_C^2. This event is referred to as *splitting*. A splitting decision can be taken at each step k by any cleaner.
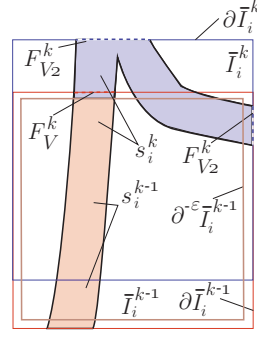
In the case, referred to as an *encounter*, where two or more cleaners get so close to each other that their images overlap, each cleaner deletes from its frontier the portions that are contained in the others' images. Figure 3 illlustrates this case.

Finally, if at a certain step the frontier of a UAV becomes empty, its clearing phase is temporarily over and it becomes a *helper*. The behavior of the helpers is described in Sec. V.
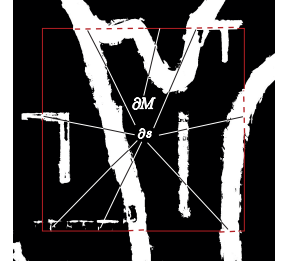
IV. IMAGE PROCESSING

In Section III the basic clearing strategy has been described under the assumption that an image processing scheme is available for reconstructing the set F_V^k at each step k. In this section we want to justify this premise by outlining a suitable algorithm that works also if a certain level of noise is present in the segmentation process.

Consider the segmented image $\bar{I}_i^k$ and denote with $\partial \bar{I}_i^k$ the boundary of $\bar{I}_i^k$ formed by the four lines of pixels on its four sides. Denote with s_1^k the segmented points in $\bar{I}_1^1$ and define

$\partial s_i^k = s_1^k \cap \partial \bar{I}_1^1$. Consider now the case $k = 1$, where only the current image $\bar{I}_1^1$ is available. However, since by assumption S belongs to $\bar{I}_1^1$ we have that F_V^1 can be computed as

$$F_V^1 = Conn\left\{S, s_1^k\right\} \cap \partial s_1^k. \tag{8}$$

Consider now the case $k > 1$. After a registration of the two images $\bar{I}_i^k$ and $\bar{I}_i^{k-1}$, it is possible to obtain a compound image that includes both, see Fig. 4a. Then consider a 'shrunk' version of $\partial \bar{I}_i^{k-1}$ that is at distance ε from $\partial \bar{I}_i^{k-1}$ and denote this internal stripe with $\partial^{-\varepsilon} \bar{I}_i^{k-1}$. The new frontier can then be computed as

$$F_V^k = Conn\left\{F_V^{k-1}, (s_i^k \cup s_i^{k-1}) \cap \partial^{-\varepsilon} \bar{I}_i^{k-1}\right\} \cap \partial s_i^k. \tag{9}$$

Loosely speaking, $\partial^{-\varepsilon} \bar{I}_i^{k-1}$ represents a virtual barrier that avoids a backward propagation of the frontier computation.

Regarding the validation of candidate primitives, this can be easily done by checking if F_V^k lies in the interior of a frame displaced with the nominal commands. Furthermore, in order to deal with noise on the motion velocity input, which would prevent the aforementioned algorithm to work, the frontier is evaluated on a shrunk virtual boundary of the image, see Fig. 4b. In this way, even if the actual motion is off by the shrinking amount the frontier will still lie in the next image, thus allowing a correct execution of the algorithm. Such a solution requires a trade-off between two opposite demands:

- amount of shrinking as small as possible to make the clearing process faster;
- amount of shrinking as large as possible to achieve robustness to imprecise motions of the camera (i.e., of the UAV).

V. FORMATION CONTROL

Dealing with a network of many roads, the usage of a single robot would not be able in general to clear all the paths belonging to the network. As already mentioned, during the *splitting* event, each cleaner can entrust a subset of F_V to another agent that automatically joins the set U_C. The objective of the formation control is to provide, whenever possible, each cleaner with a sufficient number of nearby agents, not belonging to U_C, with which is possible to
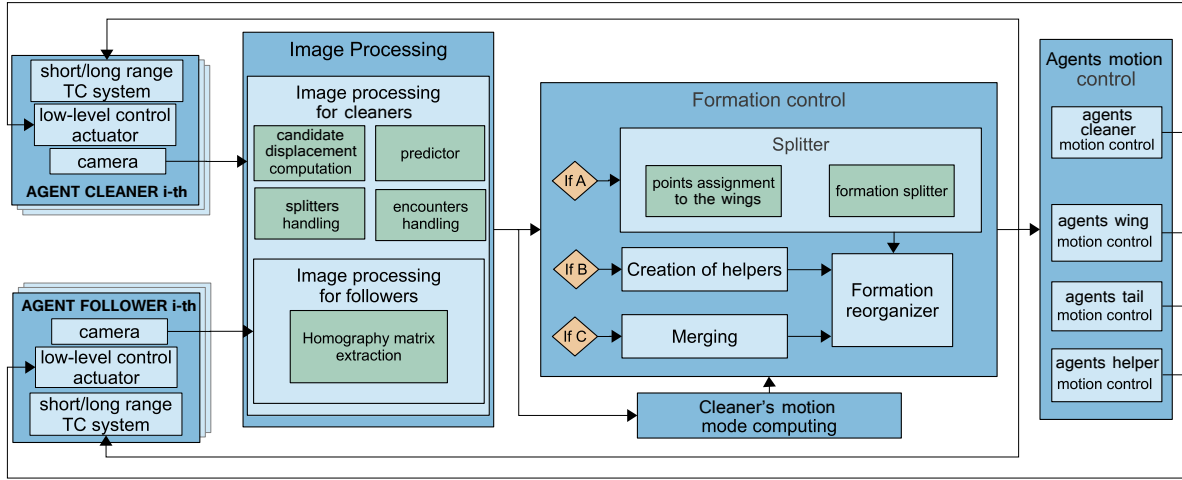
Fig. 5: Block scheme of the main operative steps of our algorithm. Starting from the left side: UAVs (input: velocities and data coming from other element of the team; output: current images); image-processing block (output: selected displacement for the cleaners and relative positions for the followers); formation control (A= splitting event; B=end of clearing; C=integration of helpers; output - desired velocities); the motion controllers represent the final step of the system.

share the frontier if necessary. This feature requires the implementation of a collaboration strategy among the agents.

Our approach is to organize the team into a variable number of formations whose agents perform specific roles. In general each formation is composed by: one *cleaner*, up to two *wing* agents denoted with U_W (called left and right wing) and up to $n-3$ *tail* agents, denoted with U_T. The formation configuration chosen is shown in Fig. 6. An additional role is the *helper*: the number of helpers can be up to n and they are not part of a formation. Such roles are not fixed for each agent but change during the clearing process. The above formation choice ensures that all communication required is feasible by only using *visual odometry* and the HBSR telecommunication system.

During the clearing process each agent, except the cleaner, has an agent, among the members of its own formation, called *target*, with respect to which it has to maintain a certain relative position. In particular:

- Cleaner: it does not have any target and it is controlled directly by the motion strategy of Sec. III-B, that uses the image processing of Sec. IV;
- Wings: their target is the cleaner (dashed horizontal red connections in Fig. 6). During the normal clearing phase they have to maintain a position such that their images always overlap two opposite halves of the image of their own target. The wings are the only agents that can be called to share, if necessary, a portion of F_V of their cleaner during a splitting event.
- Tails: the tail agents are arranged as shown in Fig. 6. The target of each tail agent is the previous tail agent (dashed green connections), except for the first whose target is the cleaner (dashed vertical red connection). After a splitting event the tail agents must fill all the empty positions left by the wing agents promoted to cleaners, and become wing agents themselves.

At $k = 1$ all the team is organized in an unique formation

containing all the n agents[1]; the formation follows U_1 executing its task until a splitting event occurs. In this case, always communicating through the HBSR channel, all the tail agents with *odd* positions abandon the formation in order to follow the new cleaner composing a new, independent formation as shown in Fig. 7. The same mechanism can be extended to each formation present at the generic time step k. In the case that the tail of the formation is empty the possible vacant wing roles cannot be replaced.

If a cleaner concludes its clearing process, then all the agents belonging to the associated formation become helper UAVs. When a UAV becomes a helper it sends a signal of *availability* on the LBLR channel; all the cleaners then reply on the same channel providing a suitable signal in order to let the helper understand the direction where to move to approach them. Following a suitable selection strategy, the helpers elect a target among the cleaners and move toward it following the direction of origin of the LBLR signal. As soon as the the distance between the helpers and their target gets sufficiently small (=i.e., HBSR communication is available) a *merging* event occurs: all the helpers become tails of the selected cleaner. Note that the whole process is completely decentralized.

A summary block-scheme representing the main operative steps of the proposed algorithm is in Fig. 5.

VI. SIMULATIONS

In order to validate the proposed algorithm we have performed a wide range of simulations. Many different kind of road networks have been considered by using real satellite images of urbanized environments, taken by the Google Maps service[2], and imported on the ground of our simulated world in the V-REP simulator[3]. A group of realistically simulated quadrotors has been used as UAV platform. Each UAV

---

[1]For reasons of clarity we assume $n > 3$, so that at least a tail is present.
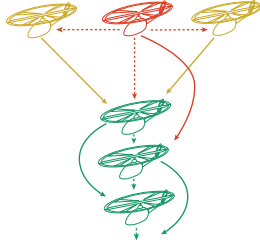[2]https://maps.google.com/
[3]http://www.coppeliarobotics.com/

Fig. 6: Red: cleaner; orange: right and left wings; green: tails. All lines represent HBSR communication needed during the clearing and splitting phases; the dashed represent in particular data coming from the *targets*.
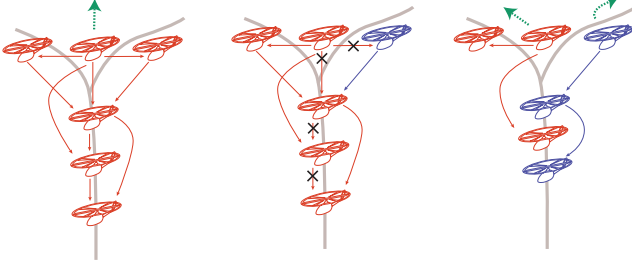

Fig. 7: Splitting strategy.

Fig. 8: Trajectories of the cleaners and stroboscopic view of the covered areas. The dashed orange line represents the boundary of the region *A*; the white line represents the starting line *S*. A video of the simulation is available at https://www.youtube.com/watch?v=ESlVS40h1XE

has a rigidly attached camera which is therefore subject to the same tilting rotations that the quadrotor performs in order to change its velocity. Image processing has been implemented using the OpenCV[4] library, the formation control through Matlab[5] and communication is handled in ROS[6].

We report the results of a representative simulation which illustrates the main features of the algorithm, see Fig. 8. A video of the simulation can be seen at https://www.youtube.com/watch?v=ESlVS40h1XE. For this simulation we selected a map of an area which is about $500\,\text{m}^2$ wide, located near Lancaster (California). All the roads are approximately $6\,\text{m}$ wide and visible from an aerial point of view. All the most frequent possible characteristics of a network of roads are represented: dead-ends, loops and junctions. In this simulation we used a team of 8 UAVs, each of them is equipped with a camera having an field of view of $\alpha = 80°$ and a resolution $L = 256$. The desired altitude is $\bar{p}^z = 70\,\text{m}$; the clearing algorithm runs at a frequency of $f = 8\,\text{Hz}$ and the absolute value of the nonzero displacement is $|\Delta| = 0.125\,\text{m}$. The flight control loop that regulates the speed of the quadrotor is based on the controller described in [19] and runs at $100\,\text{Hz}$.

Segmentation of the street has been performed incrementing the contrast of the roads with respect to the background and then using the OpenCV library. As it can be noticed from the video, in spite of a certain amount of noise and outliers in the segmentation the clearing process is executed correctly by the team. The trajectories executed by the cleaner UAVs are represented in Fig. 8 as continuous lines with a different color for each UAV; the colored squares
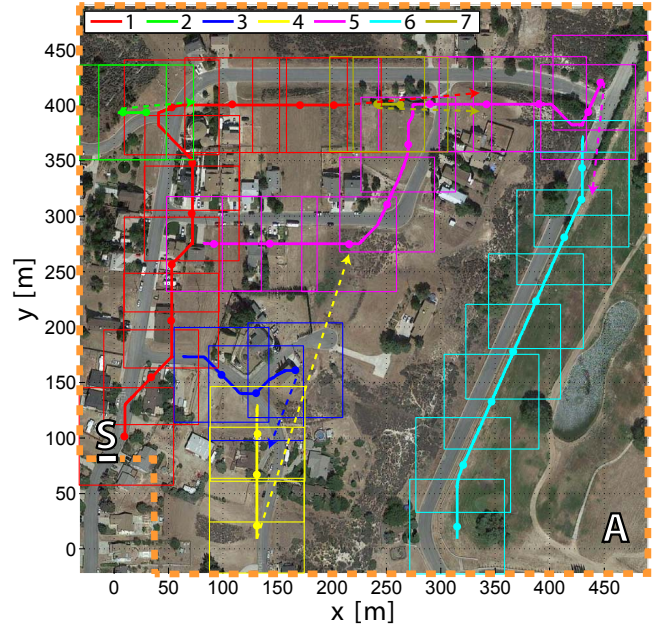
[4]http://opencv.org/
[5]http://www.mathworks.com/
[6]http://wiki.ros.org/

represent a stroboscopic view of the portions of the area visible by the cleaners along their trajectories every $10\,\text{s}$. The dashed lines represent the approximate trajectories executed by each cleaner after it becomes a helper. Notice that in this particular case one of the 8 UAVs never assumes the role of cleaner.

Note that the UAVs do not the street profile precisely; this is because each cleaner tries to push back all its current sub-frontiers in order to clear as much area of the street as possible. Note also how the whole network contained in *A* is covered by the union of squares representing the fields of view, thus demonstrating that the clearing is successfully achieved.

Figure 9 shows some magnified views of the salient phases of the clearing process summarized in Fig. 8.

In Fig. 10a we report the cumulative percentage of the road cleared by each cleaner at each step. At the first step a high increment is present because in the first frame acquired by the first cleaner all the visible road area contributes to the clearing. During the following steps only the new areas actually matters. Whenever a UAV is promoted to cleaner a new steep increment is present. The cleared area for each robot is obviously constant during the time intervals in which the robot is not a cleaner. The plot shows a remarkably uniform distribution of the workload among all the UAVs used as cleaners in the group.

In Fig. 10b the distances covered by each agent over time are shown: the small peaks along the curves represent the accelerations performed by the helpers going toward a formation to merge with it. Note how velocities are approximately constant, showing that the UAVs do not stop
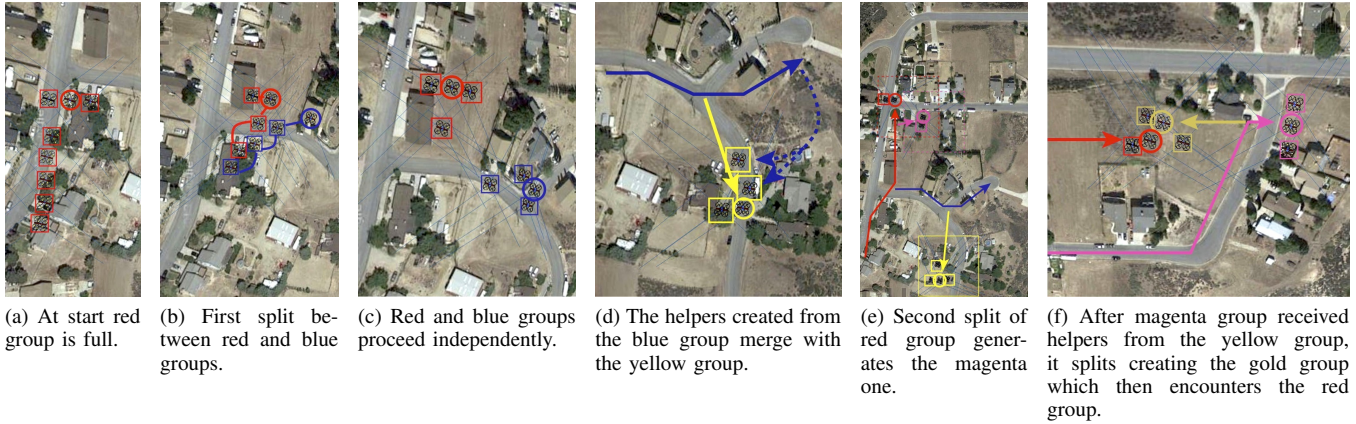
(a) At start red group is full.

(b) First split between red and blue groups.

(c) Red and blue groups proceed independently.

(d) The helpers created from the blue group merge with the yellow group.

(e) Second split of red group generates the magenta one.

(f) After magenta group received helpers from the yellow group, it splits creating the gold group which then encounters the red group.

Fig. 9: Salient phases of the simulation with quadrotor UAVs performing the clearing task on a real network of roads.

(a) Percentage of cleared road area: total (black) and for each UAV (colored).
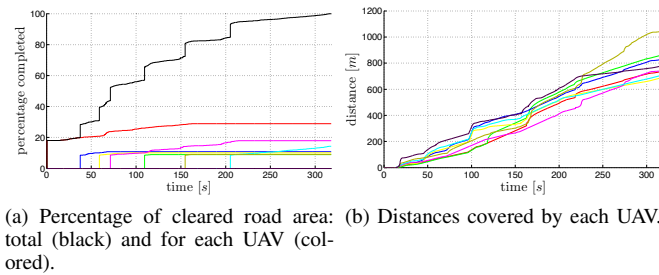
(b) Distances covered by each UAV.

Fig. 10

on average.

VII. CONCLUSION

We have presented an algorithm for aerial clearing of a completely unknown network of roads. Under a limited number of reasonable assumptions the algorithm allows a team of UAVs to complete the task by organizing themselves into formations, without any help from mapping or localization systems. Thanks to these features the robots involved in such a process use a very small amount of memory space. Through a series of simulations we determined the effectiveness of the proposed algorithm.

An interesting prospect for future works is the experimental implementation of the proposed clearing strategy with real aerial robots using, as a preliminary experiment, a reconstructed urban scenario in an indoor environment.

REFERENCES

[1] A. Howard, L. E. Parker, and G. S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 431–447, 2006.

[2] F. Pasqualetti, A. Franchi, and F. Bullo, "On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms," *IEEE Trans. on Robotics*, vol. 28, no. 3, pp. 592–606, 2012.

[3] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "The sensor-based random graph method for cooperative robot exploration," *IEEE/ASME Trans. on Mechatronics*, vol. 14, no. 2, pp. 163–175, 2009.

[4] J. W. Durham, A. Franchi, and F. Bullo, "Distributed pursuit-evasion without global localization via local frontiers," *Autonomous Robots*, vol. 32, no. 1, pp. 81–95, 2012.

[5] B. P. Gerkey, S. Thrun, and G. Gordon, "Visibility-based pursuit-evasion with limited field of view," *The International Journal of Robotics Research*, vol. 25, no. 4, pp. 299–315, 2006.

[6] X. Zheng, S. Jain, S. Koenig, and D. Kempe, "Multi-robot forest coverage," in *2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nice, France, Aug. 2008, pp. 3852–3857.

[7] B. Doroodgar and G. Nejat, "A hierarchical reinforcement learning based control architecture for semi-autonomous rescue robots in cluttered environments," in *2010 IEEE Int. Conf. on Automation Sciences and Engineering*, Toronto, Canada, Aug. 2010, pp. 948–953.

[8] V. Kumar and F. Sahin, "Cognitive maps in swarm robots for the mine detection application," in *2003 IEEE Int. Conf. on Systems, Man, and Cybernetics*, Washington, DC, Oct. 2003, pp. 3364–3369.

[9] R. Rathnam and A. Birk, "Distributed communicative exploration under underwater communication constraints," in *2011 IEEE Int. Symp. on Safety, Security and Rescue Robotics*, Kyoto, Japan, Nov. 2011, pp. 339–344.

[10] D. Portugal and R. Rocha, "Decision methods for distributed multi-robot patrol," in *2012 IEEE Int. Symp. on Safety, Security and Rescue Robotics*, College Station, TX, Nov. 2012, pp. 1–6.

[11] A. Kolling and S. Carpin, "Pursuit-evasion on trees by robot teams," *IEEE Trans. on Robotics*, vol. 26, no. 1, pp. 32–47, 2010.

[12] ——, "Multi-robot surveillance: an improved algorithm for the GRAPH-CLEAR problem," in *2008 IEEE Int. Conf. on Robotics and Automation*, Pasadena, CA, May 2008, pp. 2360–2365.

[13] J. de Hoog, S. Cameron, and A. Visser, "Role-based autonomous multi-robot exploration," in *Computation World 2009*, Athens, Greece, Nov. 2009, pp. 482–487.

[14] S. Carpin, D. Burch, N. Basilico, T. H. Chung, and M. Kölsch, "Variable resolution search with quadrotors: Theory and practice," *Journal of Field Robotics*, vol. 30, no. 5, pp. 685–701, 2013.

[15] J. W. Durham, A. Franchi, and F. Bullo, "Distributed pursuit-evasion with limited-visibility sensor via frontier-based exploration," in *2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, pp. 3562–3568.

[16] N. Michael and V. Kumar, "Opportunities and challenges with autonomous micro aerial vehicles," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1279–1291, 2012.

[17] A. Franchi, C. Secchi, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs," *IEEE Robotics & Automation Magazine, Special Issue on Aerial Robotics and the Quadrotor Platform*, vol. 19, no. 3, pp. 57–68, 2012.

[18] A. Kleiner and A. Kollig, "Guaranteed search with large teams of unmanned aerial vehicles," in *2013 IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013, pp. 2962–2968.

[19] D. J. Lee, A. Franchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano, "Semi-autonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles," *IEEE/ASME Trans. on Mechatronics, Focused Section on Aerospace Mechatronics*, vol. 18, no. 4, pp. 1334–1345, 2013.