# Filming Sport Events with Mobile Camera Drones: Mathematical Modeling and Algorithms

Enrico Natalizio, Rosario Surace, Valeria Loscri, Francesca Guerriero, Tommaso Melodia

# Filming Sport Events with Mobile Camera Drones: Mathematical Modeling and Algorithms

Enrico Natalizio*, Rosario Surace†, Valeria Loscrí†, Francesca Guerriero†, Tommaso Melodia‡

* Lab. Heudiasyc, UMR CNRS 7253, France. e-mail: enrico.natalizio@hds.utc.fr
† University of Calabria, Italy. e-mail: (rsurace, vloscri, guerriero)@deis.unical.it
‡ State University of New York at Buffalo, USA. e-mail: tmelodia@eng.buffalo.edu

***Abstract*—We introduce and formulate the *Sport Event Filming (SEF)* problem. We are interested in controlling the movement of a set of camera drones that are filming the event by moving over the field where the event takes place. The objective of the problem is to maximize the satisfaction of event viewers while minimizing the distance traveled by the camera-drones. We model the static version of the SEF problem as a Vehicle Routing Problem with Soft Time Window (VRP-STW), where the whole sequence of *actions* in the event are known *a priori*. Since this assumption is unrealistic for real sport events, we propose two families of heuristics to solve the dynamic version of the problem, where the camera-drones do not have any knowledge of the input sequence and move in reaction to the movements of the protagonists of the event. The first family (Nearest Neighbor) is based on a technique used in robotic systems, whereas the second family (Ball Movement Interception) is designed based on specific characteristics of the SEF problem. We present extensive simulation results for both families in terms of average viewer satisfaction and traveled distance for the camera-drones, when several parameters vary.**

***Index Terms*—Sport Event Filming (SEF) problem, Mobile Camera Drones, UAV Routing Problem, VRP with Soft Time Windows**

## I. Introduction

Most popular sport events are played by two teams that confront each other over a field of limited size. The game typically consists of a sequence of actions where usually one team attacks and the other defends. Typically, the objective of each team is to achieve a certain goal while preventing the opposite team from achieving the same goal. Usually the objective is accomplished by carrying, kicking, shooting or dropping a single object (the *ball* in the rest of the text), such as a ball, a football, a disk etc., in a specific area that is defended by the opposite team. The main actors of the game are the two teams, the ball and the referees, i.e., one or more people that look after player safety and make sure they obey the rules of the game.

In many sport events, a small fraction of the spectators enjoy the emotion of a close contact with the main actors and the game by surrounding the game field. All other spectators (*viewers* in the following) watch the game on TV (or on the Internet) where the event is broadcasted (or streamed) along with statistics, replays and comments. In recent years, several techniques and devices have been developed by event broadcasting companies to make the spectator feel inside the game field. Hence, new types of cameras such as automatic cameras and spider cameras have been deployed to cooperate with or replace traditional human-centric filming techniques. This work copes with the organization of a fleet of drones able to fly over a limited field to film a sport event with the objective of maximizing the satisfaction experienced by viewers who watch the game on TV. We refer to this problem as the *Sport Event Filming* (SEF) problem.

The problem is to develop strategies to coordinate the movement of a group of mobile robots in the presence of highly varying time-space constraints to film/monitor a sequence of actions while optimizing some specific objective. A solution to this problem is of interest for several application domains. Besides TV filming, it would be beneficial for environmental monitoring, disaster recovery, site inspection and exploration, etc. and would pave the way for the design of mission-oriented devices and the definition of their coordination/cooperation schemes.

The family of problems we deal with are usually referred to as Dynamic Vehicle Routing (DVR) problem, and the static variants taken into consideration in this work are all NP-hard problems. Specifically for the event filming problem some solutions have been proposed [2], [12], [16]. The main disadvantage of these solutions is that cameras are fixed. Thus, they cannot provide the same level of accuracy or entertainment given by mobile devices. To the best of our knowledge, no schemes using mobile devices have been proposed to solve this problem.

Our solution is based on controlling the movement of mobile cameras mounted on a fleet of *Unmanned Aerial Vehicles* (UAV). UAVs are usually equipped with a positioning system, storage memory, and a wireless transceiver. They can fly at considerable speed, $60\,km/h$ for commercial drones and $220\,km/h$ for military aircrafts. These devices can be controlled in a distributed fashion through the proposed algorithm to film a sport event while maximizing viewers' satisfaction and minimizing the travelled path.

The core contributions of our work can be outlined as follows:

- we describe and formulate an interesting and unexplored problem in the framework of self-organization of mobile video sensing devices;
- we model the problem as a centralized mathematical

program by assuming that the whole sequence of actions in the event is known *a priori*;
- we propose two families of algorithms to solve the *dynamic* version of the problem in a distributed way and without any knowledge of the sequence of actions.

The rest of this paper is organized as follows. Section II presents the *Vehicle Routing Problem*. The mathematical model is discussed in Section III, while in Section IV we propose two families of distributed techniques for the optimal placement of UAV camera-drones. These schemes are validated in comparison with the mathematical model, tested and analyzed through several simulation campaigns in Section V. Finally, Section VI concludes this paper.

## II. Vehicle Routing Problem and its extensions

The Vehicle Routing Problem (VRP) can be described as the problem of defining an optimal delivery from one or more depots to a number of geographically scattered cities or customers. This planning aims at determining the best route that vehicles starting from their depots must travel to reach all customers in a certain period of time. Each customer must be visited exactly by one vehicle.

In the classic version of the VRP the goal is to minimize the total transportation cost. It is well known that VRP is an *NP-hard* (*Nondetermistic Polynomial-Time Hard*) problem [8]. A special case of a Vehicle Routing Problem is the famous *Traveling Salesman Problem* (TSP), in which one is restricted to only one vehicle. Similar to VRP, TSP is an NP-hard problem [6].

The basic version of the VRP is the *Capacitated Vehicle Routing Problem* (CVRP). In this version, all vehicles are initially located at the starting point, have a limited capacity equal for all and the only constraint is not to exceed vehicle capacity. CVRP is also known to be NP-hard [15].

The VRP, TSP and CVRP cannot describe completely the Sport Event Filming (SEF) problem because they do not include time constraints. A possible extension of the CVRP is the *Vehicle Routing Problem with Time Windows* (VRPTW). The VRPTW introduces two new concepts with respect to the classical CVRP: a *time window* is assigned to each customer and a vehicle must remain in proximity of the customer for the entire duration of the time window. These time windows can be essentially of two types [4]: Hard and Soft Time Windows. In the first case none of the time windows may be violated, *i.e.,* vehicle(s) must arrive at the customer location in time to serve the customer. In the case of Soft Time Windows, instead, these time windows can be violated at the price of paying a penalty. VRPTW is also an NP-hard problem [8], and NP-complete when we fix the number of vehicles [14].

The problem of determining the movement pattern for a certain number of UAV camera-drones when they have to film an event while maximizing viewer satisfaction and minimizing the total transportation costs can be considered as a special case of the VRPTW. Specifically, it can be classified as a *Vehicle Routing Problem with Soft Time Windows* (VRP-STW), where the sequence of points in the field to be filmed represent the customer to be served. If a specific point is not timely filmed, this affects only the satisfaction of the viewers without invalidating the overall solution.

Both VRP and its extensions assume that the cities to be visited (actions in the SEF problem) are known *a priori* and will not change during the execution of the solution. However, in real applications this assumption may be too strict. In reality, we have locations to be served that can be highly variable [7]: they can be born and die at any moment, their demands can change over time even when the solution has already been calculated. Also in the SEF problem, the position to film and the time to film that location change action by action.

The problem of planning routes through service demands that arrive during a mission execution is known as the *Dynamic Vehicle Routing Problem* (DVRP) [10], because part or all the locations to reach are not known *a priori*.

In [10], the authors identify three main approaches to address DVR problems. The first approach is to simply re-optimize every time a new event takes place; in the second approach, routing policies are designed to minimize the worst-case ratio between their performance and the performance of an optimal offline algorithm that has *a priori* knowledge of the entire input sequence; in the third approach, the routing problem is embedded within the framework of queueing theory and routing policies are designed to minimize typical queueing-theoretical cost functions such as the expected waiting time in the system for the demands.

Both families of distributed algorithms we present in this work follow the first approach. In the second family, we additionally consider specific characteristics of the problem to forecast the next locations to be covered.

## III. Mathematical model for static $SEF$ problem

In this Section we introduce the mathematical formulation of the sport event filming problem, aimed at determining the routes followed by a set of camera-drones which simultaneously maximize the satisfaction of viewers and minimize the total traveled distance of the camera-drones. The problem is modeled as a *Vehicle Routing Problem with Soft Time Windows* (VRP-STW). In the formulation, we assume knowledge of the entire sequence of actions. Since this assumption is not realistic, we will use the results obtained by solving to optimality the mathematical model only as a comparative benchmark against the distributed solutions proposed in this work. Nevertheless, the effort of modeling the whole problem is very useful when the approach used to tackle the dynamic version of the problem is to re-optimize every time a new event takes place with no knowledge of the sequence of actions.

### A. Definitions and initial assumptions

*1) Definitions:* **Action**

An action consists of a sequence of simultaneous movements performed by the main actors of the event: the ball (or football, disk, etc.), the players and the referees. The complete characterization of an action is composed of a quadruple $(x, y, z, t)$ for each actor, where $(x, y, z)$ are the coordinates of the actor's position in a tri-dimensional spatial reference system and $t$ is the time instant when the actor is in that position. We assume that the camera-drones move on a plane that is fixed, parallel to the plane where the actions take place and high enough so as not to interfere with the actions. To correctly film the game, cameras will have to match the players $(x, y) - coordinates$ exactly by moving over their heads and filming them from a *perpendicular* perspective. It is worth noting that in reality a perfect matching of the $(x, y) - coordinates$ is not necessary, because the *camera aperture diameter* can compensate for moderate errors in camera positioning. Nevertheless, this assumption remarkably simplifies the mathematical modeling and, to obtain a fair comparison, it will be used also for evaluating the proposed distributed algorithms. From the camera perspective the images will be flattened on the game field plane. Hence, the $z - coordinate$ for both the players and the drones can be neglected in the definition of the problem and its solution. Our future work will also consider movement of the camera over the $z - axis$ or, equivalently, its zooming capabilities. A *transversal* (rather than *perpendicular*) perspective of filming, which is possible by considering the rotational capabilities of the cameras, would lead to a totally different problem and mathematical formulation that we do not consider in this work.

An action is defined within a time window, which consists of the following time instants and intervals:

- $t_{birth}$: is the moment when the possession of the ball is gained by another player;
- $t_{start}$: is the moment when the player who has possession of the ball starts moving with it;
- $t_{stop}$: is the moment when the player in possession of the ball loses it;
- $T_{fly}$: is the time interval between the loss of the ball by one player ($t_{stop}$) and the gain of it by another ($t_{birth}$).

We are not interested in $T_{fly}$ because we assume that a typical camera can follow the movement of the ball from one player to another for the whole event. Therefore, we will not use $T_{fly}$ in optimizing the viewer satisfaction. Instead, we want to offer the viewers the possibility of enjoying the performance of the individual players involved in the event by "personalized" shots when they get possession of the ball. The duration of an action is the time interval between $t_{birth}$ and $t_{stop}$. As discussed in Section III-A2, drones can recognize when a player is in possession of the ball. Hence, the time interval between $t_{start}$ and $t_{stop}$ involves all the movements of the player in possession of the ball as well as the consequent movements of the drone that is filming the current action, even when the player moves away from the ball reception position.

**Event**

An event is a sequence of actions that takes place in a size-limited field and in a predefined time span. In the following, we will use a subscript for $t_{birth}$, $t_{start}$ and $t_{stop}$ that indicates the action within the whole event.

**Viewer satisfaction**

The satisfaction $S_i^k$ of the viewer by watching the $i^{th}$ action of the event, filmed by the $k^{th}$ drone is modeled as

$$\begin{cases} S_{max} & t^k_{arr,i} < t_{start,i} \\ -S_{max} \cdot \frac{t^k_{arr,i} - t_{start,i}}{t_{stop,i} - t_{start,i}} + S_{max} & t_{start,i} \leq t^k_{arr,i} \leq t_{stop,i} \\ 0 & t^k_{arr,i} > t_{stop,i} \end{cases} \tag{1}$$

where $t^k_{arr,i}$ is the arrival time of drone $k$ at the position of action $i$. It is worth observing that the no-linear piecewise function (1) can be linearized by using binary variables. The related linearized function has been used when testing the proposed model. The total satisfaction of the viewer is obtained by averaging the sum of the satisfaction experienced from each action on the whole event. The satisfaction of the viewer is represented graphically in Fig. 1. It is maximized if the drone arrives at the action location before $t_{start}$ and it decreases linearly until becoming 0 when $t = t_{stop}$. Hence, the definition of viewer satisfaction matches with the definition of Soft Time Window, because the only effect of a drone not arriving before $t_{start}$ at the action location is to reduce the overall satisfaction of the viewer without invalidating the solution of the problem.

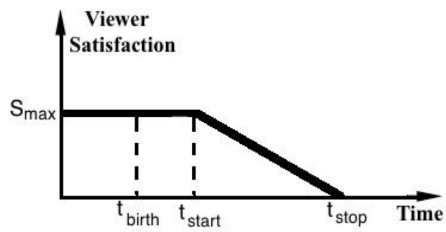


**Fig. 1:** Game action's soft time window.

*2) Relevant assumptions related to the drones:* With regards to the drones movement, we refer to UAV camera-drones as any aircraft that has a limited energy $E$ provided by its battery and capable of moving autonomously at constant speed $v$. We assume that our UAV camera-drones are equipped with a positioning system (GPS or indoor positioning system), a camera, storage memory and a wireless transceiver to send the filmed images to a base station and to allow communication and cooperation with other UAV camera-drones. These drones are capable of identifying and localizing a target by some radio frequency identification tag applied to it or by using a sensor network [9], [11], [13] deployed on the side of the field and capable of locating the target and communicating with the drones. Therefore, drones are able to recognize when a player is in possession of the ball because player and ball tags will be overlapping. When this occurs, the drone that is filming the action "sticks" to the player and follows him until the ball is released, and the current action terminates. Further extension of this work will consider in detail the communication aspects of filming and transmitting the images to a base station.

In the mathematical model we refer to the total traveled distance as the objective to be minimized (rather than the consumed energy) since this is the limiting factor for the drone

[3], [5].

### B. Variables and model assumptions

In this section we introduce all the variables, constraints and assumptions used to formulate the proposed mathematical model. At the end of this subsection, we will illustrate the constraints of the model.

*1) Variables:*

- $F = L \times W$ size-limited field;
- $0 \rightarrow T$ event time duration;
- $M = \{1...m\}$ drones able to move in 2 dimensions with constant speed $v$. Initially they have the same energy $E$;
- $N = \{1...n\}$ with $0 < m < n$ actions spatially distributed in $F$ and time-distributed in $0 \rightarrow T$;
- $N_1 \neq N_2 \neq ... \neq N_{m-1} \neq N_m$ initial positions of the drones, before they start moving to film the actions;
- $N_n$ final position of the drones, after they filmed the whole event.
- $t_{birth,i}$ birth time of action $i$, $\forall i \in N \setminus \{N_1...N_m\}$;
- $t_{start,i}$ start time of action $i$, $\forall i \in N \setminus \{N_1...N_m\}$;
- $t_{stop,i}$ stop time of action $i$, $\forall i \in N \setminus \{N_1...N_m\}$;
- $t_{birth,i} < t_{start,i} < t_{stop,i}$ $\forall i \in N \setminus \{N_1...N_m\}$;
- $t_{stop,i} < t_{birth,j}$ $\forall i, j \in N \setminus \{N_1...N_m\}$ with $i < j$ ensures that actions are sequential and non-simultaneous;
- $t_{stop,n} = T$ ensures that all actions terminate within the given time frame.
- $d^k_{max}$ maximum distance that drone $k$ is able to travel;
- $d_{ij}$ Euclidean distance between the location of action $i$ and action $j$, $\forall i, j \in N$;
- $d_{ik} \leq d_{ij} + d_{jk}$ triangle inequality for the distances among actions locations $\forall i, j, k \in N$;
- $t^k_{arr,i}$ arrival time of drone $k$ to the location of action $i$ $\forall i \in N \setminus \{N_1...N_m\}$ and $\forall k \in M$;
- $t^k_{dep,i}$ departure time of drone $k$ from the location of action $i$ $\forall i \in N \setminus \{N_n\}$ and $\forall k \in M$;
- $t^k_{i \rightarrow j} = \frac{d_{ij}}{v}$ time required by drone $k$ to move from action $i$ to action $j$ $\forall i, j \in N$ and $\forall k \in M$;
- $x^k_{ij}$ on/off variable with the following meaning:
$$x^k_{ij} = \begin{cases} 1 & \text{if arc } i - j \text{ is crossed by drone } k \\ 0 & \text{else} \end{cases}$$
- $y^k_i$ on/off variable with the following meaning:
$$y^k_i = \begin{cases} 1 & \text{if drone } k \text{ was on game action } i \\ 0 & \text{else} \end{cases}$$
- $S^k_i$ viewer's satisfaction due to the drone $k$ that filmed action $i$ $\forall i \in N \setminus \{N_1...N_m$ and $N_n\}$ and $\forall k \in M$. We already defined this in (1);
- $S_{max} = 1$ maximum satisfaction obtainable by the viewer in a single action;
- $\bar{\Psi} = \frac{\sum_{k=1}^{m} \sum_{i=m+1}^{n-1} S^k_i}{n-m-1}$ average viewer satisfaction;
- $\Psi_{min}$ minimum guaranteed satisfaction for the viewer;

*2) Assumptions:*

- The time and spatial sequences of actions are known;
- All actions (except the last) must be filmed by 1 drone;
- Since the last game action is not an action to film but only a dummy location where the drones come together so that maintenance operations can be performed, the distance traveled to reach this final position is not considered in the cost for the drones.
- All actions $\in N \setminus \{N_1...N_m\}$ have their own Soft Time Window already presented in Fig. 1;

### C. Problem Formulation

The objective function is given by the total distance traveled by the drones involved in the sport event filming (to be minimized). This can be expressed as

$$minimize \qquad \sum_{k=1}^{m} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} d_{ij} \cdot x^k_{ij} \tag{2}$$

This objective has to be accomplished by satisfying the following constraints:

$$\sum_{k=1}^{m} y^k_i = 1 \qquad \forall i \in N \setminus \{N_n\} \tag{3}$$

$$\sum_{k=1}^{m} y^k_n = m \tag{4}$$

$$y^k_i \geq x^k_{ji} \qquad \forall i, j \in N \text{ and } \forall k \in M \tag{5}$$

$$y^k_i \geq x^k_{ij} \qquad \forall i, j \in N \text{ and } \forall k \in M \tag{6}$$

$$\sum_{i=1}^{m} \sum_{j=m+1}^{n} x^k_{ij} = 1 \qquad \forall k \in M \tag{7}$$

$$\sum_{i=1}^{n-1} x^k_{in} = 1 \qquad \forall k \in M \tag{8}$$

$$\sum_{j=1}^{n-1} \sum_{k=1}^{m} x^k_{ji} = 1 \qquad \forall i \in N \setminus \{N_1...N_m \; and \; N_n\} \tag{9}$$

$$\sum_{j=1}^{n} \sum_{k=1}^{m} x^k_{ij} = 1 \qquad \forall i \in N \setminus \{N_n\} \tag{10}$$

$$\sum_{i=1}^{n} x^k_{iz} - \sum_{j=1}^{n} x^k_{zj} = 0 \tag{11}$$

$$\forall z \in N \setminus \{N_1...N_m \; and \; N_n\} \text{ and } \forall k \in M$$

$$\sum_{j=1}^{i} x^k_{ij} = 0 \qquad \forall i \in N \text{ and } \forall k \in M \tag{12}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{m} x^k_{ij} = 0 \qquad \forall k \in M \tag{13}$$

$$\sum_{i=1}^{n-1} \sum_{j=m+1}^{n-1} d_{ij} \cdot x^k_{ij} \leq d^k_{max} \qquad \forall k \in M \tag{14}$$

$$t^k_{arr,i} = 0 \qquad \forall i \in \{N_1...N_m\} \text{ and } \forall k \in M \tag{15}$$

$$t^k_{arr,j} = \sum_{i=1}^{n} (t^k_{dep,i} + t^k_{i \rightarrow j}) \cdot x^k_{ij} \tag{16}$$

$$\forall j \in N \setminus \{N_1...N_m\} \text{ and } \forall k \in M$$

$$t^k_{dep,i} \geq 0 \qquad \forall i \in N \text{and} \, \forall k \in M \tag{17}$$

$$t^k_{dep,i} \geq t_{stop,i} \cdot y^k_i \tag{18}$$

$$\forall i \in N \setminus \{N_1 ... N_m \; and \; N_n\} \text{ and } \forall k \in M$$

$$t^k_{dep,i} \leq t_{stop,n} \cdot y^k_i \qquad \forall i \in N \text{ and } \forall k \in M \tag{19}$$

$$t^k_{dep,n} = t_{stop,n} \qquad \forall k \in M \tag{20}$$

$$t^k_{arr,n} \leq t_{stop,n} \qquad \forall k \in M \tag{21}$$

$$\bar{\Psi} = \frac{\sum_{k=1}^{m} \sum_{i=m+1}^{n-1} S^k_i}{n-m-1} \geq \Psi_{min} \tag{22}$$

where $S^k_i$ is defined as in (1).

Each action (except the final one) must be filmed exactly by one drone (3), whereas the last dummy action will make all the drones move together (4). The action $j$ must be filmed by the drone that moved from a previous action $i$ for that purpose (5-6). Drones must start from different initial positions and terminate in the same final position, respectively (7-8). Exactly one entering and one outgoing arc for each action can be crossed and must be present in the final solution (9-11). Actions are sequential, therefore drones cannot reach an action occurred earlier in time than the current one and can not produce loops, $i.e.$ they cannot move away from the action and then return onto the same action (12). The drones cannot travel towards the initial positions of other drones (13), and the whole route of each drone should not exceed the maximum allowed traveled distance (14). The arrival time of the drones at the locations of the initial action is the starting time of the whole event (15), instead, the arrival time at the location of action $j$ (excluded the initial actions) by the drone $k$ must be equal to the sum of the departure time from the previous location $i$ and the time needed to move from $i$ to $j$, (16). A drone cannot move before the event started (17), it can leave the location of the filmed action only after the action terminated (18), and cannot move after the event is over (19). A drone must never leave the location of the last action, which is used to collect all the drones in the same location for maintenance operations (20), and this location must be reached before the end of the event (21). The average satisfaction experienced by the viewer must never be less than a certain minimum value (22).

## IV. Distributed algorithms for dynamic VRP-STW

When no knowledge of the event sequence of actions is available *a priori*, the static vehicle routing problem with soft time windows of Section III becomes a dynamic vehicle routing problem that requires different optimization techniques.

In Section II we mentioned that three different approaches have been identified to tackle the dynamic version of the problem. The first of these approaches simply proposes to re-optimize every time a new event takes place. This approach is the most suited for the specific communication and movement capabilities of the drones to offer a feasible and practical solution to the event filming problem. In fact, the sub-optimal solution will be computed action-by-action by the drones that cooperate by exploiting their communication capabilities in a distributed and self-organized fashion. For this purpose, we introduce in the distributed strategies the *coordination time*, $T_{coord}$, which is the time needed by the drones to communicate with each other and determine which of them will move to follow the newly generated action.

In the following we present two families: Nearest Neighbor (NN) and Ball Movement Interception (BMI), each of them consisting of four different distributed techniques to solve the event filming problem.

### A. Nearest Neighbor

The *Nearest Neighbor* technique for DVR problems in robotic system is presented in [1]. The core idea is that viewer satisfaction increases when a drone is able to reach the location of the current action as quickly as possible, and that the minimum traveled distance is achieved by the closest drone. Thus, the drone that is the closest to the location of the action is the one chosen to move and film the action. The following three techniques are extensions of the basic NN technique.

### B. Nearest Neighbor-Division Field

A disadvantage of the NN technique is that when a sequence of actions occurs in a limited area, the same drones will be chosen to film it. If the duration of this sequence extends over time, it would cause one drone to reach its maximum feasible traveled distance much earlier than the others.

Based on these considerations, we introduce the *Nearest Neighbor-Division Field* (NN-DF). In the NN-DF technique, each drone is assigned to a portion of the field, and it will film the actions that are located inside that portion.

This technique has the disadvantage of not choosing the drone that is the nearest to the current action, which can result in a reduced satisfaction for the viewer. We will see in Section V the effects of this with respect to the reduced area for each drone to monitor.

### C. Nearest Neighbor with Specular Repositioning

In the previous two techniques only one drone moves when a new action is born. The *Nearest Neighbor with Specular Repositioning* (NN-SR) technique considers drones as belonging to a pair. When one of them, $k$, is chosen to move to film an action for which it is the nearest neighbor, the drone that is closest to the position specular to the action position, $\bar{k}$, moves as well to mirror the movement of the first. More precisely, let $L$ and $W$ be the length and the width of the field. When drone $k$ moves to the position of the new action $(x_a, y_a)$, $\bar{k}$ will move to $(L - x_a, W - y_a)$. It is worth noting that drones are not coupled at the beginning of the event, instead $\bar{k}$ is chosen action-by-action depending on the proximity to the action specular position. We expect that this technique, which results in drones traveling more than the previous techniques, will be more reactive and timely in filming the actions so as to offer a higher satisfaction to the viewer.

### D. Nearest Neighbor with Quasi-Specular Repositioning

A generalization of the NN-SR technique is the *Nearest Neighbor with Quasi-Specular Repositioning* (NN-QSR). The NN-SR technique makes pairs of drones move specularly. As we have already highlighted, this behavior can lead to a quick depletion of the maximum allowed traveled distance, due to the specular movements of the drone ($\bar{k}$) that is not filming any action. Thus, the idea behind the NN-QSR is to make the center of the field be an attractor for $\bar{k}$ while it is repositioning in the direction of $k$'s specular position.

The attraction strength on the movement can be modulated through an appropriate *detour factor*, $0 \leq \beta \leq 1$. When $\beta = 0$, no detour is applied on the movement of $\bar{k}$, which moves to the specular position in respect of the current action position, and NN-QSR coincides with NN-SR. When $\beta = 1$, $\bar{k}$ is completely detoured towards the center of the field. For intermediate values between 0 and 1, $\bar{k}$ move on a point on the straight line between these two extreme points. More precisely, if $(x_a, y_a)$, $L$, $W$ are the positions of the new action, the length and the width of the field, respectively, then $\bar{k}$ will move to $(L \cdot (1 - \frac{\beta}{2}) - x_a \cdot (1 - \beta), W \cdot (1 - \frac{\beta}{2}) - y_a \cdot (1 - \beta))$.

By detouring the movement of $\bar{k}$, we expect a higher satisfaction of the viewer as compared to the NN and NN-DF techniques, without introducing a high traveled distance expenditure as in the NN-SR technique.

### E. Ball Movement Interception

All the previous techniques work well if $t_{birth}$ and $t_{start}$ are sufficiently far in time to allow a drone to reach the action location before $t_{start}$. In fact, these techniques try to solve the dynamic version of the proposed problem simply by adapting as quick as possible the position of one (or one pair of) drones. None of them try to forecast the location to film for next action before its $t_{birth}$. As we described in Section III-A, our static model introduces the time of "flight" of the ball when the ball is not possessed by any player, $T_{fly}$. This interval of time between $t_{stop_i}$ and $t_{birth_{i+1}}$ could be used to forecast the location of next action.

We can realistically assume that drones, which are able to constantly detect the ball and its location, are also able to easily compute their trajectory. For the sake of simplicity, in this work, we consider only that the ball moves along straight lines. We consider the parabolic trajectory of the ball as flatted on the straight line lying on the game field plane, and we do not take into consideration special effects that can be given to the ball.

By assuming that drones know the trajectory of the ball, they can estimate the next player who will hold the ball. Through this estimation, before the ball reaches the next player they can start moving towards the straight line between the position of the previous action and that of the player expected to receive the ball. Thus, we introduce a new family of techniques, called *Ball Movement Interception* (BMI), which includes all the previous techniques augmented by this knowledge: *Ball Movement Interception* (BMI), *Ball Movement Interception with Division Field* (BMI-DF), *Ball Movement Interception with Specular Repositioning* (BMI-SR) and *Ball Movement Interception with Quasi-Specular Repositioning* (BMI-QSR). It is important to note that we do not assume that unexepected interceptions of the ball destined to a specific player are neglected. In fact, such events would simply cause a degradation in the performance of this family of techniques.

## V. Performance Results

In this Section we will show three simulation campaigns illustrating selected results obtained for both the model and the algorithms when several parameters vary. We consider the *average viewer satisfaction* as the output parameter for assessing the quality of the route chosen for the drones, and the *total traveled distance* as the output parameter representing the cost of the route. The first simulation campaign is used to validate the distributed techniques and tune the main parameters. In the second campaign we study the impact of the detour factor, $\beta$, on the performance of the Specular Repositioning techniques. The third simulation campaign is a more general comparison among the different distributed techniques. The results have been achieved by using LINGO 9.0 for the mathematical model and MATLAB 7.9.0.529 (R2009b) for the algorithms, and they have been averaged over 1000 runs with a confidence interval of 95%. The parameters presented in Table I are used in all the simulation campaigns, specific differences will be highlighted in each campaign subsection. In Table II, we list the distributed algorithms names and the corresponding acronyms.

| Parameter | Value |
|---|---|
| Size of the game field ($L \times W$) | $110 \times 80$ $[m^2]$ |
| Max Distance Feasible by Drones | 65 $[km]$ |
| Speed of Drones | 15 $[m/s]$ |
| Action Min Duration ($t_{birth} \rightarrow t_{stop}$) | 0.2 $[s]$ |
| Ball Min and Max Speed | $\{1 \div 40\}$ $[m/s]$ |
| Coordination Time ($T_{coord}$) | 0.2 $[s]$ |
| Threshold Satisfaction ($\Psi_{min}$) | 95% |
| Max Satisfaction ($S_{max}$) | 1 |
| Actions Spatial and Temporal Distribution | random |
| Number of run for each scenario | 1000 |

**TABLE I:** Fixed parameters used for all simulations

| Name | Acronym |
|---|---|
| Mathematical Model | *MM* |
| Nearest Neighbor | *NN* |
| Nearest Neighbor with Division Field | *NN-DF* |
| Nearest Neighbor with Specular Repositioning | *NN-SR* |
| Nearest Neighbor with Quasi-Specular Repositioning | *NN-QSR*$_\beta$ |
| Ball Movement Interception | *BMI* |
| Ball Movement Interception with Division Field | *BMI-DF* |
| Ball Movement Interception with Specular Repositioning | *BMI-SR* |
| Ball Movement Interception with Quasi-Specular Repositioning | *BMI-QSR*$_\beta$ |

**TABLE II:** Simulated algorithms and their acronyms

We simulate the behavior of the algorithms when the number of actions in the event and the duration of each action vary, respectively. The number of actions is useful to characterize the time-space variability of the actions in the event, whereas the duration of an action represents the dynamicity of the event. Both these input parameters depend on the kind of sport that has to be filmed and their characterization is left as a future work. In our simulations we also used a variable number of drones ($2 \div 6$), but for matter of space we will be able to show few results of scenarios with 2 and 4 drones.

### *A. Validation of the distributed algorithms*

A *a priori* knowledge of the whole event actions gives a significant advantage to the mathematical model with respect to our algorithms. Therefore, in this section we verify that the mathematical model achieves results that are an upper bound for the viewer satisfaction and a lower bound for the total traveled distance with respect to the basic heuristics of the NN and the BMI families.

In this campaign, we performed simulations when both the number of actions and the action maximum duration vary. In the interest of space we report only the diagrams of the maximum duration of an action varying between 6 and 12 s, with a step of 2 s. All parameters for this simulation campaign are reported in Table III. As previously stated, the actual duration of each action is randomly chosen in the interval between the fixed minimum and the maximum. The number of actions in the event is set to 20. From Fig. 2, we can see that the action maximum duration does not affect the total traveled distance and that the NN technique is very close to the centralized optimum obtained through the mathematical model, whereas the BMI needs $40\%$ more movements than NN. This extra movement is compensated by an increased performance in terms of viewer satisfaction. The BMI performance is between $10\%$ and $25\%$ better than NN. The highest difference among NN and BMI is for short actions, that is the case when the ball moves frequently among the players, while both the techniques tend to reach the mathematical model when the maximum duration of the action increases.

### *B. Performance evaluation varying detour factor*

In this simulation campaign we want to investigate the impact of the detour factor on the performance of the QSR techniques. Hence, we compare the results of NN-QSR and BMI-QSR, when the detour factor, $\beta$, varies in the range $\{0 \div 1\}$. As we did for the first simulation campaign, we let the number of actions in the event and the duration of an action vary, as shown in Table IV. The range considered for the former parameter has been increased to match the time duration of a real event. The performance of the two techniques in terms of average viewer satisfaction for different number of actions is reported in Fig. 4, and traveled distance for different maximum durations of the actions in Fig. 5.

From Fig. 4 we observe that use of the ball movement interception does limit the need to reposition the drone specular to the drone that is filming the action. The BMI technique leads to a high viewer's satisfaction when the attraction strength towards the center of the field increases, whereas the NN-QSR technique presents a maximum when the detour factor is between $0.5$ and $0.6$. This also means that different detour strengths should be applied depending on the used technique. As expected, the viewer satisfaction experienced with the BMI technique is higher on average.

The same behavior for the NN-QSR technique is presented

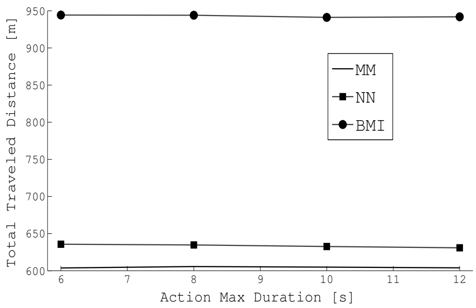


**Fig. 2:** Distributed algorithms validation: total traveled distance when the maximum duration of the actions varies

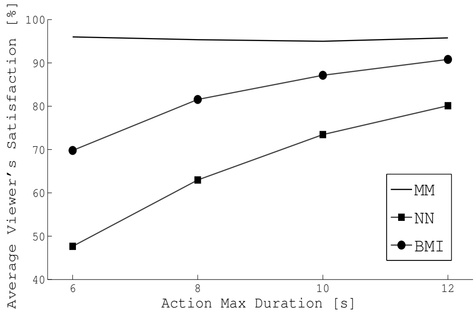


**Fig. 3:** Distributed algorithms validation: average viewer's satisfaction when the maximum duration of the actions varies

| | |
|---|---|
| Number of Drones | 2 |
| Drone $k$ Position | $\{(-1)^k \frac{L}{4} + \frac{L}{2}, \frac{W}{2}\}$ |
| Action Max Duration ($t_{birth} \rightarrow t_{stop}$) | $\{6 \div 12\}[s]$ |
| Number of Actions | 20 |

**TABLE III:** Simulation parameters used for distributed algorithms validation

| | |
|---|---|
| Number of Drones | 2 |
| Drone $k$ Position | $\{(-1)^k \frac{L}{4} + \frac{L}{2}, \frac{W}{2}\}$ |
| Action Max Duration ($t_{birth} \rightarrow t_{stop}$) | $\{2, 6, 10\}[s]$ |
| Detour Factor ($\beta$) | $\{0 \div 1\}$ |
| Number of Actions | $\{100, 500, 1000\}$ |

**TABLE IV:** Simulation parameters used for simulating NN-QSR and BMI-QSR

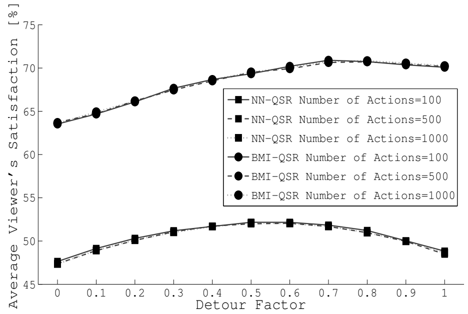


**Fig. 4:** QSR techniques: average viewer's satisfaction when the detour factor and the number of actions vary

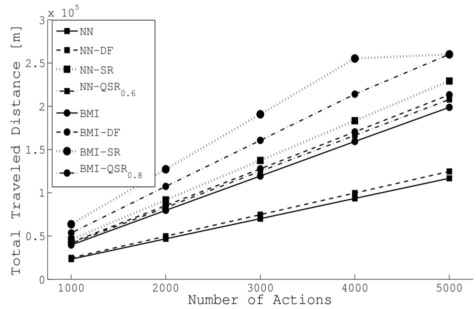


**Fig. 6:** Distributed algorithms comparison: total traveled distance for fixed actions maximum duration (6 $[s]$)

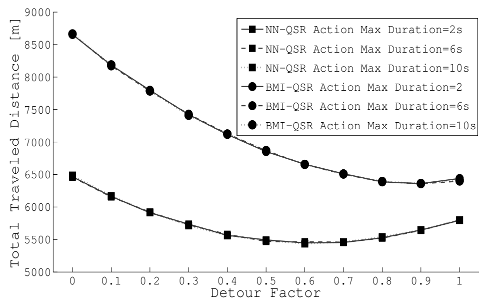


**Fig. 5:** QSR techniques: traveled distance when the detour factor and the action maximum duration vary

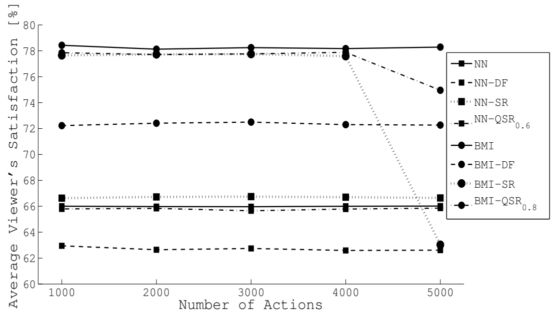


**Fig. 7:** Distributed algorithms comparison: average viewer's satisfaction for fixed actions maximum duration (6 $[s]$)

in Fig. 5, where we can appreciate the existence of a minimum in the distance traveled by the drones when the detour factor is around $0.6$. The BMI-QSR improves its performance when the detour factor grows until values very close to 1. It is interesting to remark that for both the techniques, a decrease in the distance traveled by the drones corresponds to an increase in the satisfaction experienced by the viewer. Comparatively, the NN-QSR technique leads the drones to travel about $1.3km$ on average less than the BMI-QSR technique, with a corresponding about $18\%$ on average of decrease in the viewer's satisfaction. For both the output parameters, the number of actions and the action maximum duration do not significantly impact the performance of the different techniques, therefore the three simulated algorithms produce overlapping curves.

### *C. Comparison of Positioning Techniques*

This last simulation campaign, whose main parameters are in Table V, shows the results when all the distributed techniques are applied to a scenario with a variable number of actions and a fixed action maximum duration (Fig. 6, 7) and a fixed number of actions and a variable action maximum duration (Fig. 8, 9).

In Fig. 6 we show that the distance traveled by the drones grows linearly with respect to the number of actions for all the algorithms. Thus, it is easy to predict the distance that each technique will make drones travel through an estimate of the number of actions a real event will consist of. As expected, the NN technique is the best in terms of traveled distance, both when the Division Field is used and when it is not. The basic technique of the BMI family performs as the third best for this metric, which is a very encouraging result because of the consideration we will make about the average viewer satisfaction. On average the BMI technique makes drones travel about $73\,km$ more than NN in the considered interval. The techniques with Specular Repositioning are the worst for this metric because of the distance traveled by the drone that does not film the action.

The situation is reversed in Fig. 7, which shows the average

| | |
|---|---|
| Number of Drones | 4 |
| Drone $k$ Position | $\{(-1)^k \frac{L}{4} + \frac{L}{2}, (-1)^{\lceil k/2 \rceil} \frac{W}{4} + \frac{W}{2}\}$ |
| Action Max Duration ($t_{birth} \rightarrow t_{stop}$) | $2 \div 10\ [s]$ |
| NN-QSR Detour Factor ($\beta$) | 0.6 |
| BMI-QSR Detour Factor ($\beta$) | 0.8 |
| Number of Actions | $\{1000 \div 5000\}$ |

**TABLE V:** Simulation parameters used for distributed algorithms comparison

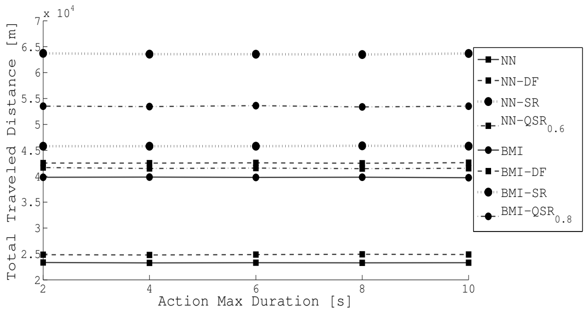


**Fig. 8:** Distributed algorithms comparison: total traveled distance for a fixed number of actions (1000)

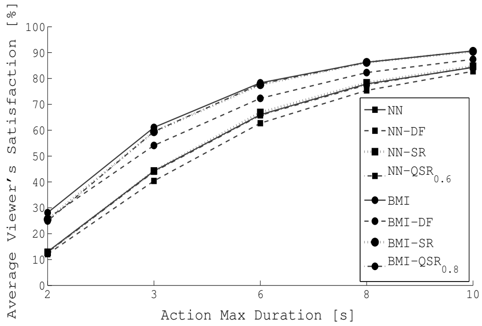


**Fig. 9:** Distributed algorithms: average viewer's satisfaction for a fixed number of actions (1000)

viewer satisfaction. All techniques in the BMI family achieve a higher satisfaction than the corresponding techniques in the NN family. The distance between the best techniques of the two families for this parameter, which are the basic technique and the SR technique, is $14\%$ on average. When the upper limit on the feasible traveled distance is reached, both the satisfaction achieved by BMI-SR and BMI-QSR start decreasing, since drones are not allowed to move anymore. Thus, the instantaneous satisfaction goes to zero and the average satisfaction decreases. Until the feasible distance limit is reached, the two techniques of the BMI family perform very similarly, the only main difference is that the QSR let drones travel more efficiently. Instead, we can appreciate some difference in the performance of the same techniques for the NN family, the SR technique performs $2\%$ better on average than the QSR technique.

In Fig. 8 we can appreciate the traveled distance when the maximum duration of the actions varies. We can see that all the proposed algorithms are scalable with respect to this input parameter, and the heuristics ranking is the same of that in Fig. 6. Fig. 9 shows a logarithmic growth of the average viewer's satisfaction when the actions maximum duration increases. Very quick and short actions create troubles to all the algorithms, which do not achieve more than $30\%$ of viewer's satisfaction, whereas they perform much better and reach $90\%$ of satisfaction when the maximum duration is the upper value. On average, the BMI techniques have a gain of $15\%$ over the corresponding NN techniques.

## VI. Conclusion

We have introduced the Sport Event Filming problem, whose objective is to maximize the average satisfaction of a sport event viewer while minimizing the distance traveled by the camera-drones that film the event. First, we modeled the problem as a Vehicle Routing Problem with Soft Time Window. We then considered the dynamic version of the problem where knowledge of the entire sequence of actions is not assumed to be known *a priori*. The dynamic version of the Sport Event Filming can be treated as a Dynamic Vehicle Routing problem. We solved it by re-optimizing the position of the drones every time that a new action occurs. As a future work, we will use a different approach based on queuing theoretical cost functions.

## References


[1] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith. Dynamic vehicle routing for robotic systems. *IEEE Proceeding*, 99(9):1482–1504, 2011.

[2] F. Daniyal and A. Cavallaro. Multi-camera scheduling for video production. In *Euorpean Conference on Visual Media Production (CVMP)*, London, UK, Nov. 2011.

[3] L. De Filippis, G. Guglieri, and F. Quagliotti. A minimum risk approach for path planning of uavs. *J. Intell. Robotics Syst.*, 61(1-4):203–219, January 2011.

[4] B.L. Golden, S. Raghavan, and E.A. Wasil. *The vehicle routing problem: latest advances and new challenges*. Operations research/computer science interfaces series. Springer, 2008.

[5] M.O. Hammouri and M.M. Matalgah. Voronoi path planning technique for recovering communication in uavs. In *Proceedings of the 2008 IEEE/ACS International Conference on Computer Systems and Applications*, pages 403–406. IEEE Computer Society, 2008.

[6] R.M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 40(4):85–103, 1972.

[7] P. Kilby, P. Prosser, and P. Shaw. Dynamic vrps: A study of scenarios. *University of Strathclyde Technical Report*, (APES-06-1998):1–11, 1998.

[8] J.K. Lenstra and A.H.G. Rinnooy Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981.

[9] T. Melodia and I. F. Akyildiz. Cross-layer QoS-Aware Communication for Ultra Wide Band Wireless Multimedia Sensor Networks. *IEEE Journal of Selected Areas in Communications*, 28(5):653–663, 2010.

[10] M. Pavone. *Dynamic Vehicle Routing for Robotic Networks*. PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2010.

[11] S. Pudlewski and T. Melodia. C-DMRC: Compressive Distortion-Minimizing Rate Control for Wireless Multimedia Sensor Networks. In *Proc. of IEEE Intl. Conf. on Sensor and Ad-hoc Communications and Networks (SECON)*, Boston, MA, USA, June 2010.

[12] F. Z. Qureshi and D. Terzopoulos. Surveillance in virtual reality: system design and multi-camera control. *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVRP 07*, pages 1–8, June 2007.

[13] J. Rullán-Lara, S. Salazar, and R. Lozano. Uav real-time location using a wireless sensor network. In *Positioning Navigation and Communication (WPNC), 2011 8th Workshop on*, pages 18–23, April 2011.

[14] M.W.P. Savelsbergh. Local search in routing problems with time windows. *Annals of Operations Research*, 4:285–305, 1985.

[15] P. Toth and D. Vigo. *The vehicle routing problem*. SIAM monographs on discrete mathematics and applications. Society for Industrial and Applied Mathematics, 2002.

[16] X. Zhou, R.T. Collins, T. Kanade, and P. Metes. A master-slave system to acquire biometric imagery of humans at distance. In *ACM International Workshop on Video Surveillance*, pages 113–120. ACM Press, 2003.