Taxonomy on Multi-robot Target Detection and Tracking

Cyril Robin*† and Simon Lacroix*†
*CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
†Univ de Toulouse, LAAS, F-31400 Toulouse, France

***Abstract*—Target detection and tracking encompasses a variety of problems such as coverage, search, patrolling, observing, pursuit-evasion, *etc.* These problems are studied by several communities, that tackle them using diverse formulations, hypotheses and approaches. This variety and the fact that target related robotics problems are pertinent for a large spectrum of applications has motivated a large amount of contributions, which have mostly been surveyed according to one or another viewpoint. In this article, our objective is to go beyond the frontiers of specific communities and specific problems, and to enlarge the scope of prior surveys. We define classes of missions and problems, and relate the results from various communities according to a unifying taxonomy, and propose a transverse analysis of the approaches, models and lacks that are common through all the tackled problems, and isolate the current main research directions.**

I. Motivations

Detecting, localizing or following targets is at the core of numerous robotic applications, in both adversarial and cooperative contexts. Much work has been devoted in various research communities to such problems, which are often referred to as "pursuit-evasion" problems. This very evocative term actually encompasses a variety of scenarios that pertain either to mono- or multi-robot contexts, considering either a single or multiple targets, and whose objective is either to detect, to capture or to track them. On the other hand, other similar problems are named differently and make use of specific vocabulary, *e.g.*, surveillance, search or tracking. This is partly explained by the different application contexts considered (industrial, civilian or military), and by the fact that different communities tackle them with different standpoints (*e.g.*, sensor data processing, symbolic or geometric task planning, task allocation, game theory, *etc.*)

The variety of target related robotics problems and proposed approaches has motivated a vast amount of contributions, and several surveys focused on specific problems are available [1], [2], [3], [4], [5], [6]. In this article, our objective is to go beyond the frontiers of specific communities and specific problems, and to enlarge the scope of prior surveys. Target detection and target tracking, the two broad classes of scenarios related to targets, have *a priori* little to do one with another (and similarly the approaches to solve them), and are often executed in sequence. But in actual applications, these scenarios must be achieved by the same robots, and are sometimes tackled simultaneously: hence we believe it is relevant to analyse them together.

This article introduces a coarse taxonomy and the associated vocabulary of the various robotics missions related to targets (Section II), so as to explicit how the researches carried on in different communities relate. It transversely analyses the approaches and models that are common through the various work related to the tackled problems (Section III), and it finally highlights open areas of research (Section IV). Note that due to a lack of space, this paper does not review the main work in each defined area – the interested reader may refer to [7].

II. Taxonomy

Typical robotics target related scenarios are automated surveillance of secured areas, frontier patrolling, secured area clearing, target tracking or chasing, *inter alia*. In all these scenarios, the environment is mostly known, and exploring the environment is not considered. However, the availability of a prior map is not mandatory (although it could be a side effect of another condition, for instance in *coverage*), and the map criterion does not appear in our taxonomy. Besides, the targets may either be mobile or fixed, but we focus in this article on mobile targets, which are more challenging.

The taxonomy is summarized in Figure 1. It is organized as a tree, in which each branching is defined by a specific criterion. Each leaf refers to a class of problems, including possible variations in the formulation or assumptions. It defines coherent notations and definitions of the problems used throughout the article. Note that even though we try to comply with widely accepted vocabulary, there may be conflicts with definitions used by some authors. The names and definitions of the various problems are indeed not standardised, especially when considered in different communities: the same word may refer to different problems in the literature, and so we stick to the taxonomy vocabulary throughout the paper.

The first branching criterion of the taxonomy relates to prior knowledge on the target position, and yields the two main classes of problems, that often occur in sequence: detecting targets on the one hand, and tracking detected targets on the other hand.

A. Target Detection

Target detection problems consist in finding (*detecting*) a target in a given environment. They may concern one or several targets and may be tackled with one or multiple sensors, either by actively sweeping the environment with mobile sensors, or by monitoring signals emitted from fixed static sensors.

We refer to this later class of problems as *coverage*: they mainly involve sensor positioning strategies, which often come to partition the environment and accordingly distribute sensors within this environment.

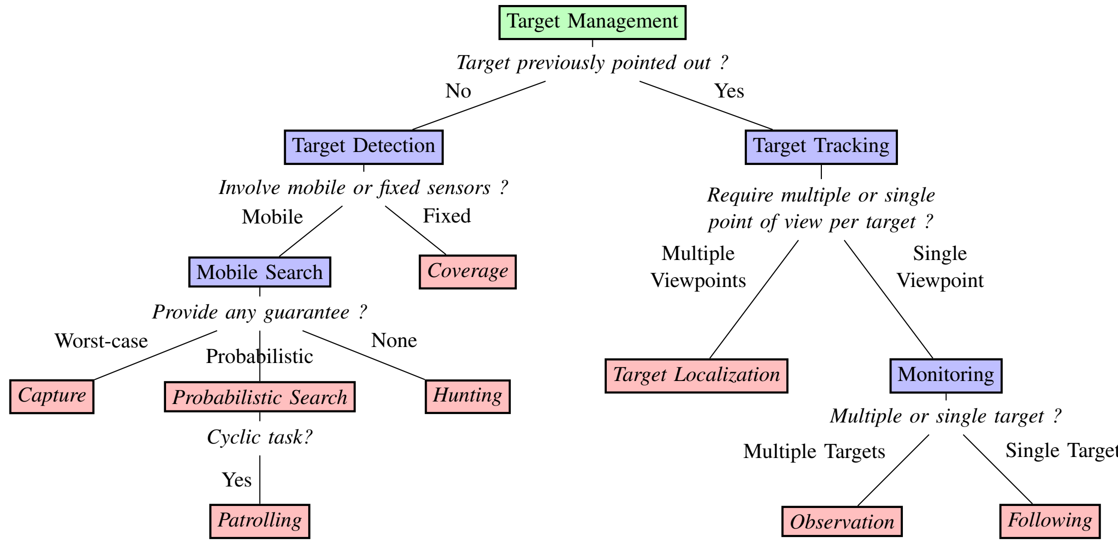

Fig. 1. Proposed taxonomy of the target management problems. Branchings correspond to criteria, here denoted by questions, each red leaf correspond to a specific *problem* analysed in the following sections.

When mobile sensors are exploited, the problem is strongly related to path planning, and we refer to it as *mobile search*. Such problems can be addressed either locally or globally. Depending on the models, the assumptions and the approaches, some authors try to provide worst-case guarantees for the performance (*capture*), whereas some provide probabilistic guarantees (*probabilistic search*) and others do not provide any guarantee at all (*hunting*). The *search* may have a cyclic aspect (*patrolling*), although this presents no interest for *capture* or *hunting*. Note that for each of these problems, a variant that consists in "surrounding" the target in order to prevent its evasion is sometimes considered, instead of merely watching it, or catching it.

a) ***Coverage****:* In *coverage*, the environment is necessarily known and the objective is to optimally position a set of fixed sensors. The traditional form of *coverage* is the famous *Art Gallery* problem which has been well studied and for which numerous results have been obtained [1]. Variants of *coverage* include mobile sensors, but the proposed solutions always focus on the sensor placement aspect (*where* to set the sensors?), and not on path planning (*how* to reach the selected positions?).

b) ***Capture****:* In the *capture* problem, optimality and completeness are essential characteristics. The goal is to clear a given known area while providing a worst-case guarantee, meaning that if a target is inside the considered area, it will be found, no matter what. There is no prior knowledge or assumption on the target location, targets may even have "super abilities" (like infinite speed), and pursuers try to surround them. *Capture* is often referred to as a *pursuit-evasion*, but also as *search and secure*, or as the *cops and robbers* game, mostly when the solution rely on *graph clearing* [3]. Work on this subject usually has strong mathematical foundations, and are developed following two main approaches [2], [3]: stating the problem as a *graph clearing* one, or as a purely geometric one, within 2D polygonal environments. Often, the objective is to assess the minimal number of pursuers required to provide a worst-case guarantee.

c) ***Probabilistic Search****:* The main difference between *capture* and *probabilistic search* is the absence of worst-case guarantee in the latter, in which *probabilities of detection* are assessed [6]. The reason is mainly a lack of resources (robots or time) to tackle the worst-case problem, but it can also be a compromise between efficiency and the probability of occurrence of particularly difficult situations.

Probabilistic Search exploits probability distributions over the model of the environment (of the target presence, of the target visibility, *etc.*) Most authors try to provide bounds on the probability of detecting/catching the target. The target model may either be adversarial or not, the latter being usually easier to deal with because of its lower algorithmic complexity. The non-adversarial target model is widely used in *search and rescue* scenarios, for which emergency and time constraints usually prevent performing an exhaustive search and impose priorities – which is well handled by probabilistic models.

d) ***Patrolling****:* When the mobile search is cyclic, it is denoted by *patrolling*. *Patrolling* may be seen as a cyclic version of *probabilistic search*, as it involves analyses of statistical performance over time, and especially the time elapsed between two visits to the same point. It is a rather recent area of research, whose interest has risen over the last decade [4]. The continuous formulation is related to the watchman route problem (WRP), which consists in finding the shortest closed path in a given polygon such that all points of this polygon are visible from at least one point in the path.

e) ***Hunting****:* There are finally some cases where no guarantee at all is provided for detection or capture of the target, which we refer to as *hunting*. The absence of guarantee comes from the lack of resources (robots, time) or information – in the absence of which no useful probability models for the target location can be exploited, for instance. *Hunting* is often considered within a multi-robot context.

B. Target Tracking

The second major class of problems, *target tracking*, corresponds to the tasks that arise when one or several targets have been detected or assigned – often following the success of target detection tasks. Here, coping with a target may imply keeping it in sight, to provide information on it (mainly to localise it over time, but identifying it can also be an objective), or to catch it. In all cases tracker robots need to stay "close" to the targets, the required distance being zero when it comes to catch the targets. Coping with a single target may require one or more robots, depending on the context. It is for instance preferable to have multiple vantage points on each target to refine their locations. We refer to this latter class of problems as *target localization* problems. We also distinguish the one vs. one problems (*following*) from the multi-robot multi-target problems (*observation*).

*f) **Target Localization**:* The *target localization* problem consists in tracking a target with several robots in order to improve knowledge about the target, in particular the precision of its estimated position. It is most often a multi-robot problem, in which case solutions involve selecting different points of view to maximize the information gain. Of course, data fusion, cooperation, communication and multi-robot localization are issues to be considered. Several targets may be involved, and several observation points of view on each target are often required. Multi-robot *target localization* has naturally gained interest with the development of research on multi-robot systems.

*g) **Following**:* The class of *following* problems is the traditional form of pursuit-evasion. [5] present a survey and the history of the related work. Early work pertained to naval conflict scenarios, and the traditional pursuit problem is also known as *Lion and Man*. In the original version, a single purser (the Lion) is chasing a single evader (the Man) with the same speed. Many different versions of the problem have been defined, with different speeds, environment models, visibility conditions, *etc. Following* embraces all these variations, which are tracking problems involving only one pursuer and one evader.

*h) **Observation**:* The *observation* problem is stated as follows: given several robots and several (moving) targets, how to control the robots in order to simultaneously observe all the targets, and if not possible, how to minimize the time during which any target is not observed by at least one of the robots. This problem has been rigorously specified by Parker in 1997 [8], who refers to it as the CMOMMT problem (*Cooperative Multi-robot Observation of Multiple Moving Targets*). One of the main challenges is to correctly allocate targets to robots and to decide how and when the observers should trade targets.

III. Common Models and Approaches

Through the whole variety of problems gathered by our taxonomy, there are many comparable aspects, be it similar approaches, models, assumptions, validation processes, or flaws. Here, we try to highlight the ones which we consider important, namely the world and agents models, the main approaches, the current trends, and validation processes. This part relates to both the left and the right parts of our taxonomy, as similar platforms, formalisms and models are common to both the detection and tracking problems. In actual scenarios, the tracking may indeed directly follow the detection phase, or be performed in parallel: in a multi-robot context, once a target is detected, some robots track it while the others continue the detection task. Note that there is a large variety of models and approaches, but at the end results and the validation processes assess their validity *a posteriori*: we also discuss these validation processes.

A. Models

Models that represent the environment and the agents (the robots and the targets) are at the core of the decision process, and are required whichever the chosen approach. The combination of environment and agents models yields the ability to *predict* the outcome of the agents actions, *i.e.*, what the agents are able to do and the expected consequences of the possible actions. Most of the encountered models are rather simple, and in our opinion often too simple. They provide a limited expressiveness, and if they ease the way to solve the problem (somehow by reducing its complexity), they are a too coarse abstraction of the reality, hardening the integration issues, and thus minimizing the realistic validation of the proposed approaches.

1) The environment:

a) Beyond 2D models: Most of the authors use a 2D single-layer representation of the world, be it a grid [9], [10], [11] or more continuous models (using tessellation [12], [13], [14], [15] or not [16], [17]). This implies that obstacles to motions and observations are the same. Hence, realistic situations like areas that are obstacles for AGVs (Autonomous Ground Vehicles) but not for their sensors (*e.g.*, ground holes or water ponds), or where AAVs (Autonomous Aerial Vehicles) can fly over but not observe under (*e.g.*, undergrowth) can not be represented.

Along with the growing computation power, more complex models (2.5D [18], 3D [19], or multi-layered models [10], [11]) have recently been proposed. Such models embody more information and are hence more realistic, allowing finer strategies. For instance, one is able to take advantage of higher vantage points to observe a larger viewshed [18] or to distinguish areas that block motions from areas that block observations [11].

Besides these more realistic models, some authors exploit highly abstract representations, as in [20]. This often eases the finding of solutions, and allows to rigorously assess algorithmic complexity. However, the loss of information induced by the abstraction may impede the validity of the solutions when confronted to the real world.

b) Discrete Worlds: The discrete grid or tessellation models are widely used, as they straightforwardly define graphs, upon which algorithms exploiting graph theory can be built [21], [15]. However, it is difficult to transform a metrical model of the world into a meaningful topological model [22], and only few authors provide means to do so [23]: some directly assume that the graph is available (*e.g.*, handmade [24]) while others use "random" sampling as a compromise between the continuous world and a discrete model.

Discretization is also a workaround for the computational complexity of continuous numerical models, even when the algorithms are theoretically valid with continuous models [13]. As stated by Bhattacharya "While discretization invariably implies a certain level of approximation and deviation from the original metric space, in order to make any continuous problem computationally feasible it is an indispensable trade-off." [25]. As a matter of fact, continuous models are mostly used for local reactive control or greedy decisions [16], [17]. On the contrary, discrete models are often used for mathematical proofs in abstract representations [21], [20]. The assumption of continuity is appealing and eases guarantees and proofs, but it raises an issue: indeed there are discontinuities in the real world that impact both motions and visibility and that continuous abstract models may not handle [26].

c) Space-time manifold: Apart from the space representation, time representation is crucial, and can actually hardly be decorrelated from the space representation, especially in discretized models. Note that discrete time can be used with continuous space models [17], but continuous time representations are also used [20]. Time discretization is typically used to define a countable number of states, and as highlighted by LaValle [27], "the next state x_{k+1} will usually not lie exactly at a discretized value" (be it temporal or spatial). This means that discretization will probably reduce the coherence between the real state and the modeled state, and one has to ensure that the algorithms are robust to such incoherences.

However, to ease the mathematical proofs, the "one space unit travelled per one time unit" hypothesis is convenient, and often does not lead to any loss of generality [13]. But it may impact the quality of the resulting solutions. As a matter of fact, the relation between space and time is really sensitive when accurate coordination is required or assumed, or when a robot cannot "wait" for the others, for instance when pursuing a target [11], [28]. In both cases, experiments will validate or not the assumptions about time, and therefore, one has to take a great care about how this validation step is led.

d) Environmental constraints: The environment of the missions can challenge the algorithms by compromising some assumptions or by constraining or even preventing some actions. For instance, some algorithms take for granted that one can mark the environment in a way or an other [29], [30]. In this case, one has to pay attention to the feasibility of such markings – and to their cost. The recent advent of RFIDs (Radio-Frequency Identification) offers an interesting technical solution to mark the environment [31], but they can hardly be used in every context – a difficulty overlooked by most simulators where marking can be easily emulated. As asserted by Glad *et al.*, "Robotic systems come with their own hypotheses that are more restrictive than in simulation." Here, again, only realistic experiments should validate the approach. For instance, the RFID detection distance range can largely impact the performances [32].

While most reviewed work tackle simple environments, *i.e.*, 2D grounds or volumes with only no fly zones for AAVs, some authors tackle more challenging environments like water, which strongly impact the sensing, communication and motion abilities [33].

2) The agents: The term "agent" stands here for the robots and the targets. For the scenarios considered in this article, the considered actions are the motions, the observations and the communications. A model of each of these capacities is required, and its choice is obviously strongly related to the choice of the environment model, as both are combined to assess the outcome of the actions.

a) Motion model: The model accuracy of the motion capacities, and especially the kinematics and dynamic constraints, varies a lot. The more the world model is abstract, the more the motion model is. With graphs, the motion model only describes if a given node (area) can be reached by the agent or not, possibly with a cost (*e.g.*, distance, time, energy...). It is largely assumed that this cost estimate is good, and how the agent should or will effectively move is not of the concern of the motion model in use. This assumption may be presumptuous, and only validation processes can assess its realism. Nowadays, accurate pattern-based motion models are used for (winged-)AAVs, and more rarely for AGVs [9]. Information on the environment geometry may also help to refine or constrain the agents motions [10]. However, a precise or realistic motion model is not always required, especially for the target. For instance, assuming an infinite speed for the target yields the definition of conservative worst-case-guaranteed strategies [3].

b) Sensor model: The sensor models are mainly used for the robots, except in the case of stealth tracking [17] or when the authors also consider the evader point of view [34]. Sensor models may be very basic: most often only distance matters. However, some authors explicitly consider the field of view and visibility constraints [16], [14], [35], with an orientation, a maximum angle, and a distance. Finer models may also represent 2.5D or 3D information [18], [19], or a multi-layered world [11]. Most considered sensors are light-based sensors, be it cameras or LIDARs and so most of visibility constraints are line-of-sight.

c) Communication model: Although communication may be intuitively considered as very similar to sensing (when there is a visibility link, one may reasonably assume that there is a communication link), most of the authors use the full connectivity assumption. However convenient this assumption is, its realism is questionable and may strongly impact the efficiency of the approach. Indeed, accurate communication models are complex and expensive [36]. This explains why some authors build their approaches upon the communication requirements and issues [37], or specifically study its impact [32].

d) Expected behaviour: Apart from the models of actions (motion, sensing, communicating), one may want to model the expected behaviour of the target or the other teammates. The target model is often a probabilistic motion model, be it random walk [28], [38], [37] or more elaborate Bayesian or Markovian models [9], [24], [39]. Predicting the target's behaviour and motion allows elaborating more sophisticated or less conservative strategies. Indeed, in the adversarial case, the target behaviour can be more accurately modeled with the game theory [15], [40], [41]. Various target models are also used as a metric to compare algorithms [42]. Indeed, the model of the target's behaviour does matter, as finer strategies are possible when the model is correct, but they can turn to be counter-productive when the model is not adequate [9].

Modeling the teammates behaviour is also relevant: the main motivation is to reduce the need for communication, as implicit coordination built on the prediction of others motions can be achieved [43]. The models may be hard coded or learned [44].

B. Approaches

Robotics being at the crossroads of numerous disciplines, the state of the art contains a large variety of approaches tackling similar – if not identical – problems. Among the numerous criteria that define an approach, we distinguish (a) the theoretical and analytical results from the experimental results, (b) the centralized systems from the decentralized ones, (c) the cooperative patterns from the "selfish" ones, considering both implicit and explicit cooperation, (d) the consideration of uncertainties, especially through probabilistic models, and (e) the planning processes from the optimization processes. Note that the selection of an approach obviously comes with the definition of the chosen models.

a) Theory vs. Practice: Among every problem defined by our taxonomy, one can distinguish two trends: a large part of the contributions are focused on experimental results [45], [17], [28], while others provide theoretical results [46], [47]. This is a coarse partition, and some papers propose both kinds of results [37], however we believe that this distinction does matter.

Theoretical results are essential as they light the way to efficient and practical solutions. Numerous theoretical results usefully state the complexity of a given class of problems [37], [40], [48], [24], [49]: in case of NP-difficult problems, one should probably focus on off-line computation, or suboptimal online solutions. Some papers also give hints on the solutions: for instance, LaPaugh proved that "recontamination" does not help for solving the *capture* problem [46].

Besides theoretical milestones, most analytical approaches are not applicable in real conditions considering the computational constraints. As stated by Parker about *observation*: "analytical techniques have been developed for solving this problem in complex geometrical environments. However, these previous approaches are very computationally expensive – at least exponential in the number of robots – and cannot be implemented on robots operating in real-time" [38]. This statement endorses experimental results, *i.e.*, "solutions that works for real".

Although the approaches focusing on integration and experiments can be criticized, we stand for the importance of such work, that addresses what matters in the end: solutions that effectively solve real world problems. These approaches often rely on local considerations, and result in efficient and sometimes elegant systems [17], [16], [43]. The associated algorithms are most of the time supported by validation processes, on which we comment further below.

b) Centralized and decentralized systems: The target related problems considered in the literature are mainly multi-robot problems, hence the question of centralized and decentralized systems arises. Many centralized algorithms have been proposed over the years: they are more convenient to provide global optimality. However, they face strong constraints in the real world: they often require full connectivity, are sensitive to dynamic environments, especially when solutions are computed off-line, and most of them hardly scale up, in particular with the number of robots [22].

Most target related problems are at least NP-hard, and one can face computational issues with centralized algorithms. Centralized approaches usually provide optimal or near-optimal solutions, where the computational requirements are not the issues. Otherwise, one should instead focus on the benefits brought by decentralized systems [37]. The latter are more robust, generally scale well with the number of robots, and are more adapted to real world constraints (dynamic environment, communication constraints, *etc.*), easing the integration process [50], [45]. Decentralized approaches generally provide suboptimal solutions with only local optimality [29]. Still, they provide interesting performances, especially under realistic constraints, and constitute the current main research trend [8], [29].

c) The need for cooperation: Most of the surveyed problems require several robots to be solved adequately, and the quality of the solutions is generally improved as a team of robots allows more flexibility in the strategies. But there are many ways to organize robot teams. The robots may cooperate or simply perform their tasks independently, following a prior task allocation. The latter case is well illustrated by both the cycle- and partition-based patrolling schemes [4] whose main advantage is that they do not require any communication, avoiding the associated issues. Their drawback is their weaker robustness to robot failures or changes in the environment of the mission definition, as they cannot modify the team strategy globally.

Online cooperation helps to solve problems efficiently, and is even required for some problems which required tight coordination (*capture*), or dynamic adaptation to the incoming data such as target observation (*observation*, *target localization*, *probabilistic search*). The benefits of coordination are tightly related to the structure of the environment: highly structured worlds like office environment benefit less from a tight coordination than open environments [51].

The cooperation can either be implicit or explicit. The latter allows to finely control the resulting system, because every decision and action are discussed and broadcasted through the whole team (or at least the surroundings agents). However, there are issues with the communication load and the combinatorial complexity of the decision processes. Explicit cooperation is often made through *task allocation*, where the goal is to allocate a set of tasks between several agents to optimize various criteria. The large literature about task allocation is beyond the scope of this article, but it is worth noticing that the target detection problem has often been used to compare or benchmark allocation algorithms [52], [45].

d) Uncertain and dynamic environments: Robots face many uncertainties: sensors are prone to a variety of errors and noise, the agents behaviour can hardly be predicted accurately as results of actions are affected by sources of uncertainty, communications fail, *inter alia.* Ignoring these uncertainties will likely result in a defective system when facing the real world. Two main strategies, which can be coupled, help to prevent this: taking into account uncertainties at the modeling level, and re-planning on-line when the gap between the modeled state and the real state is too large.

Probabilistic models have gained interested over the past 20 years in all robotic problems, including the ones considered here. This is explained by the need to improve the models on which planning rely and by the advent of computing performances which allows to handle these models. The trendy probabilistic models are, in a rough chronological order, classical probabilities [22], [28], Bayesian models [9], [24], [53], Markovian models (MDP) [9], and partially observed Markovian models (POMDP) [33], [24], [39]. The latter are the more appealing but their extensive computational cost has prevented their application to large instances of problems [39]. Yet recent improvements on decentralized POMDP using separability conditions increase considerably the scalability [54]. Probabilistic solutions tend to yield finer strategies but at a significant computational cost, and most importantly they remain sensitive to the modeling phase. For instance, determining the best order of Markovian models has a strong influence on performance, but is not trivial to achieve [9].

Numerous problems tackled here are NP-hard at least, and the computational cost is heavy with probabilistic models, which explains why many solutions are computed off-line for problem instances of reasonable size [25], [48], [22]. However, the real world is rarely static, and one often has to react to external events (like target motion) without being able to precompute strategies for each and every possible state (the state space being globally intractable). In such cases, on-line computing is required. It brings robustness as one is able to recompute a valid solution on demand, when needed. However, on-line computing often comes with local considerations and thus suboptimality. This dilemma is well illustrated by the *following* problem, which has been shown to be entirely decidable, but is NP-complete [40]. This is why the efficient state-of-the-art solutions consider only local information and are computed on-line [16], [39]. They are not optimal, but perform well, even in difficult environments.

e) Planning vs. Optimization: The detection and tracking problems tackled here roughly come to determine *who does what, when, and where*. Most of them are formulated either as *planning* problems or as *optimization* problems. While the underlying problem remains the same, the formulation differs a lot, and so do the solutions and the results.

Planning comes to simulate actions and their effects in order to decide the actions sequence to perform to reach a given state. Several formalisms exist to implement planners, and the critical stage is the modeling of the action space, the action effects and the world states. One of the many available planners will provide a solution on the basis of these models.

In the formalism of *optimization*, an objective function to optimize with respects to some constraints is defined. More than the choice of a solver, the critical stages are the quantification of the world states and actions, and the definition of the objective function.

One key difference is that optimization requires numerical values, while traditional planning work with qualitative statements. Correctly evaluating these numerical values may be straightforward (for instance when trying to minimise the *idleness* in *patrolling*), but also more complex for others (like *capture*). Optimization is close to raw data and commands while traditional planning is semantically expressive and allows a higher level of abstraction in the models, easing hierarchical planning (*e.g.*, the *Hierarchical Task Networks (HTN)* [55]). Besides, a hierarchical decomposition allows to mix both traditional planning (*e.g.*, to roughly define tasks) and optimization (to refine the local solution). Recently, thanks to the increase of computing power, stochastic optimization has been proven to be efficient in solving complex problems.

C. Results and validation process

Apart from the theoretical results, most papers propose one or several algorithms and associated models to solve a given problem. The algorithms may come with theoretical guarantees (complexity, optimality), but experiments are required to assess their applicability. For instance, Eaton and Zadeh "solved" the *following* problem in 1962 [47], but the proposed solution is actually not applicable, and to the best of our knowledge not been implemented on-board any robot: it operates in an abstract space which is not correlated to the real world – or at least no one provided a valid transformation between the real world and this abstract space. There have been much work on *following* since then: several approaches have good performances, but are far from Zadeh's solution.

This illustrates that algorithms can only be validated with integration: only testing them on-board robots in an actual context will assess that they can cope with the challenging issues that are uncertainties, non-determinism, asynchronous systems, *etc.* Not considering these common issues yields solutions that are not robust.

Yet, most papers lack actual validation. Of course authors usually try to provide a fair description of their experimental protocol (parameters, number of runs, description of the simulator, *etc.*) but very few present realistic testing environments. The proposed approaches are mostly validated with *ad hoc* simulations [18], [49], [29], [35], *i.e.*, simulations that neglect to represent most of real world characteristics. These simulated worlds are often discretized in time and space by construction, and there are no or few uncertainties. In this context, the initial assumptions and simplifications that lead to the algorithms are introduced into the simulators, which can hardly exhibit the solution flaws. Experiments on-board robots may present the same limits when they twist the reality to fit the models (*e.g.*, restraining the agents movements to an artificial grid-based space [9], [53] or artificially enforcing the time discretization [28]). This gap between models and reality have been spotted and studied by several authors [50], [30], [53].

The validation process often present other flaws as the lack of state-of-the-art comparisons – algorithms are often "benchmarked" against trivial solutions like random walkers or brute-force solutions [49], [26] –, or the lack of "culture of statistics" – results are rarely statistically significant. Some of these flaws are understandable: thorough validations call for a lot of engineering, numerous tests and a logistic out of reach of most researchers. Also, fair comparisons are restrained by the variety of robots and testing environments between teams and labs.

Nevertheless, some authors provide nice state-of-the-art comparisons [29], [34], [35] and others have recently introduced better statistic analysis [42], [51] with standard deviations and p-values (Wilcoxon test, Shapiro-Wilk test). Besides, in real practical experiments, it is hard to obtain enough significant results to make a statistical evaluation. Efforts have also been made to homogenize the testing platforms or to reuse past testing environments or simulators to fairly compare algorithms [28]. Challenges like RoboCup (including the Rescue Virtual Robot Competition [56], [57]) play an important role to validate and compare the systems in similar environments and setups. Some papers also present benchmarks of various algorithms and compare them through a set of metrics [57], [42], which offer references in the considered domains. Making the code publicly available to foster experimental reproducibility and comparison is also a good practice for the validation process.

We believe that realistic simulations is a key to tackle the validation issues, as they provide common testing environments with realistic conditions, while easing statistically significant experiments. They also ease the transfer of the results across different approaches without performing all the experiments again. Realistic simulators differ from *ad hoc* ones in the way they model reality: the simulated space and time are continuous, and they generally embed a physical engine which properly models the robots' dynamics. Sensors are at least geometrically modeled (*e.g.*, ray-tracing), and noise can be introduced. The development of realistic simulators requires engineering efforts, but fortunately one can now easily find off-the-shelf open source simulators (*e.g.*, Morse [58], Gazebo [59], or USARSim [56]). These simulators come with various robot models and environments, and using a common framework and a modular architecture (like ROS [60]) would even allow to compare running strategies against one another (*e.g.*, a pursuer strategy against an evader strategy elaborated by another team). Validations that involve open source codes and scenarios defined in realistic simulators can provide both reproducible experiments and statistically significant results and comparisons, and would also pool efforts in the communities.

IV. Concluding remarks

This paper presents a unifying taxonomy of the various target related robotics problems. It goes beyond the frontiers of specific communities and specific problems, and enlarges the scope of prior surveys. We have identified transverse models and approaches that are common through all the tackled problems. From this overall analysis, we believe the three most important points on which one should put efforts are the modelling, the development of decentralized approaches, and the validation.

The finest models that have been recently been proposed embrace probabilistic considerations, multi-layered world models and hierarchical representations. These improved models are now tractable thanks to advances in sensing and increasing computational power. They meet the need in robotics to fill, or at least to reduce the gap between models and reality. The first results given by these finer models are promising, and one may wish and expect that they become the new state-of-the-art standard. Communications remain however to be modeled (and considered) in a more realistic manner to allow valid approaches.

Besides, considering that many analysed problems are NP-hard at least, we think suboptimal yet efficient real-time algorithms are the key to solve the real cases scenarios while handling online the dynamics changes. One should focus on decentralized algorithms that scales well. Anytime algorithms may be a bridge between efficiency and optimality [23], especially when time matters as in *search and rescue* scenarios. Yet, most of the time only experimental bounds on the performances are given, and we yearn to see more theoretical bounds on the suboptimality of the resulting solutions.

Finally, there should be drastic changes in the validation processes of the proposed approaches. Following Portugal and Rocha's statement, "it is the authors belief that research in this field should be more oriented towards effective solutions with applicability in the real world" [53]. Yet too many papers provide unrealistic test conditions which prevent the evaluation of such applicability. True validation should involve reproducible results, with a statistical analysis and a fair comparison to other state-of-the-art solutions. Modular architectures combined to realistic simulators as a validation platform, and used with common datasets, can provide all the tools required by a solid validation process. This would also ease the development of new algorithms and the adoption of breakthroughs from related communities.

Acknowledgments

This research is partially funded by the DGA.

References

[1] J. Urrutia, "Art gallery and illumination problems," *Handbook of computational geometry*, 2000.

[2] B. Alspach, "Searching and sweeping graphs: a brief survey," *Le matematiche*, vol. LIX, pp. 5–37, 2006.

[3] T. H. Chung, G. a. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Autonomous Robots*, vol. 31, no. 4, pp. 299–316, Jul. 2011.

[4] D. Portugal and R. Rocha, "A survey on multi-robot patrolling algorithms," *Technological Innovation for Sustainability*, pp. 139–146, 2011.

[5] P. J. Nahin, *Chases and Escapes: The Mathematics of Pursuit and Evasion (New in Paper)*, Princeton University Press, Ed. Princeton University Press, 2012.

[6] L. D. Stone, *Theory of Optimal Search (2nd Edition)*, Dec. 2007, vol. 71, no. 356.

[7] C. Robin and S. Lacroix, "Multi-robot Target Detection and Tracking: Taxonomy and Survey," LAAS/CNRS, Tech. Rep., 2014.

[8] L. E. Parker and B. A. Emmons, "Cooperative Multi-Robot Observation," in *ICRA'97*, no. April 1997, 1997, pp. 2082–2089.

[9] H. Yu and R. Beard, "Probabilistic path planning for cooperative target tracking using aerial and ground vehicles," in *American Control Conference (ACC)*, 2011, pp. 4673–4678.

[10] E. F. Flushing, L. Gambardella, and G. A. D. Caro, "GIS-based Mission Support System for Wilderness Search and Rescue with Heterogeneous Agents," *IROS'12 Workshop on Robots and Sensors integration in future rescue INformation system (ROSIN'12)*, 2012.

[11] C. Robin and S. Lacroix, "Failure anticipation in pursuit-evasion," in *Proceedings of Robotics: Science and Systems VIII*, 2012.

[12] P. Fazli, A. Davoodi, P. Pasquier, and A. K. Mackworth, "Fault-Tolerant Multi-Robot Area Coverage with Limited Visibility," in *ICRA'10 Workshop: Search and Pursuit/Evasion in the Physical World*, 2010.

[13] D. E. Soltero, M. Schwager, and D. Rus, "Decentralized path planning for coverage tasks using gradient descent adaptive control," *The International Journal of Robotics Research*, Oct. 2013.

[14] S. Bhattacharya, S. Candido, and S. Hutchinson, "Motion Strategies for Surveillance," in *RSS'07*, 2007.

[15] R. Murrieta-Cid, T. Muppirala, A. Sarmiento, S. Bhattacharya, and S. Hutchinson, "Surveillance Strategies for a Pursuer with Finite Sensor Range," *IJRR*, vol. 26, no. 3, pp. 233–253, Mar. 2007.

[16] R. Murrieta-cid, H. H. Gonzales-Banos, and B. Tovar, "A reactive motion planner to maintain visibility of unpredictable targets," in *ICRA'02*, 2002.

[17] T. Bandyopadhyay, Y. Li, and D. Hsu, "Stealth Tracking of an Unpredictable Target among Obstacles," *Algorithmic Foundations of Robotics VI*, pp. 43–58, 2005.

[18] A. Kolling, A. Kleiner, M. Lewis, and K. Sycara, "Pursuit-Evasion in 2.5D based on Team-Visibility," in *IROS'10*, 2010.

[19] T. Bandyopadhyay, M. A. Jr, and D. Hsu, "Motion planning for 3-d target tracking among obstacles," *Robotics Research*, 2011.

[20] S. Alexander, R. Bishop, and R. Ghrist, "Pursuit and evasion in non-convex domains of arbitrary dimensions," in *RSS'06*, no. 0, 2006.

[21] T. Parsons, "Pursuit-evasion in a graph," *Theory and applications of graphs*, 1978.

[22] M. Moors, T. Rohling, and D. Schulz, "A probabilistic approach to coordinated multi-robot indoor surveillance," in *IROS'05*, 2005.

[23] G. Hollinger, A. Kehagias, and S. Singh, "GSST: anytime guaranteed search," *Autonomous Robots*, vol. 29, no. 1, pp. 99–118, Apr. 2010.

[24] G. Hollinger, S. Singh, J. Djugash, and a. Kehagias, "Efficient Multi-robot Search for a Moving Target," *The International Journal of Robotics Research*, vol. 28, no. 2, pp. 201–219, Feb. 2009.

[25] S. Bhattacharya, R. Ghrist, and V. Kumar, "Multi-robot coverage and exploration on Riemannian manifolds with boundaries," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 113–137, Oct. 2013.

[26] M. Israel, E. Khmelnitsky, and E. Kagan, "Search for a mobile target by ground vehicle on a topographic terrain," in *2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel*. IEEE, Nov. 2012, pp. 1–5.

[27] S. M. LaValle, H. H. Gonzales-Banos, C. Becker, and J.-c. Latombe, "Motion Strategies for Maintaining Visibility of a Moving Target," in *ICRA'97*, 1997, pp. 731–736.

[28] J. Strom, R. Morton, K. Reilly, and E. Olson, "Online probabilistic pursuit of adversarial evaders," in *ICRA'10 Workshop: Search and Pursuit/Evasion in the Physical World*, 2010.

[29] H.-N. Chu, A. Glad, O. Simonin, F. Sempe, A. Drogoul, and F. Charpillet, "Swarm approaches for the patrolling problem, information propagation vs. pheromone evaporation," in *ICTAI'07*, 2007, pp. 442–449.

[30] A. Glad, O. Simonin, O. Buffet, and F. Charpillet, "Influence of Different Execution Models on Patrolling Ant Behaviors: from Agents to Robots," in *AAMAS'10*, 2010.

[31] M. Mamei and F. Zambonelli, "Spreading pheromones in everyday environments through RFID technology," *In Proc. of the 2nd IEEE Symposium on Swarm Intelligence*, pp. 281–288, 2005.

[32] J. Pugh and A. Martinoli, "Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization," in *SIS'07 - Swarm Intelligence Symposium*, Apr. 2007, pp. 332–339.

[33] F. Ehlers, "Multi-Robot Teamwork in Multistatic Sonar," in *ICRA'10 Workshop: Search and Pursuit/Evasion in the Physical World*, 2010.

[34] S. Markov and S. Carpin, "A cooperative distributed approach to target motion control in multirobot observation of multiple targets," *IROS'07*, 2007.

[35] C. Vo and J.-m. Lien, "Visibility-Based Strategies for Tracking and Searching Unpredictable Coherent Targets Among Known Obstacles," in *ICRA'10 Workshop: Search and Pursuit/Evasion in the Physical World*, 2010.

[36] A. R. Mosteo, "Multi-Robot Task Allocation for Service Robotics : from Unlimited to Limited Communication Range," Ph.D. dissertation, Universidad de Zaragoza, 2010.

[37] G. A. Hollinger and S. Singh, "Multirobot Coordination With Periodic Connectivity: Theory and Experiments," *Transactions on Robotics*, vol. 28, no. 4, pp. 967–973, Aug. 2012.

[38] L. E. Parker, "Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets," *Autonomous Robots*, pp. 231–255, 2002.

[39] T. Bandyopadhyay, N. Rong, M. Ang, D. Hsu, and W. S. Lee, "Motion Planning for People Tracking in Uncertain and Dynamic Environments," in *ICRA'09*, May 2009.

[40] R. Murrieta-Cid, R. Monroy, S. Hutchinson, and Jea, "A complexity result for the pursuit-evasion game of maintaining visibility of a moving evader," in *ICRA'08*, 2008.

[41] M. Vieira, R. Govindan, and G. Sukhatme, "Scalable and practical pursuit-evasion with networked robots," *Intelligent Service Robotics*, 2009.

[42] T. Sak, J. Wainer, and S. K. Goldenstein, "Probabilistic multiagent patrolling," *Advances in Artificial Intelligence*, pp. 124–133, 2008.

[43] B. Grocholsky, R. Swaminathan, J. Keller, V. Kumar, and G. Pappas, "Information Driven Coordinated Air-Ground Proactive Sensing," in *ICRA'05*, April 2005, pp. 2223–2228.

[44] Z. Xu, R. Fitch, and S. Sukkarieh, "Decentralised Coordination of Mobile Robots for Target Tracking with Learnt Utility Models," in *ICRA'13*, 2013, pp. 2006–2012.

[45] N. Kalra, D. Ferguson, and A. Stentz, "Hoplites: A market-based framework for planned tight coordination in multirobot teams," in *ICRA'05*, 2005.

[46] A. LaPaugh, "Recontamination does not help," *Princeton University, Computer Science Dept.*, 1982.

[47] J. H. Eaton and L. Zadeh, "Optimal pursuit strategies in discrete-state probabilistic systems," *Trans. ASME Ser. D, J. Basic Eng*, 1962.

[48] A. Kolling and S. Carpin, "Pursuit-Evasion on Trees by Robot Teams," *Transactions on Robotics*, vol. 26, no. 1, pp. 32–47, Feb. 2010.

[49] V. Isler, S. Khanna, J. Spletzer, and C. J. Taylor, "Target Tracking with Distributed Sensors: The Focus of Attention Problem," *Computer Vision and Image Understanding*, no. March 2005, pp. 1–37, 2005.

[50] B. Jung and G. S. Sukhatme, "Tracking Targets Using Multiple Robots: The Effect of Environment Occlusion," *Autonomous Robots*, pp. 191–205, 2002.

[51] F. Amigoni, N. Basilico, and A. Li, "How much worth is coordination of mobile robots for exploration in search and rescue?" *RoboCup 2012: Robot Soccer World Cup XVI*, 2013.

[52] Z. Cai, L. Sun, H. Gao, and P. Zhou, "Multi-robot Cooperative Pursuit Based on Task Bundle Auctions," *Intelligent Robotics and Applications*, pp. 235–244, 2008.

[53] D. Portugal and R. P. Rocha, "Distributed multi-robot patrol: A scalable and fault-tolerant framework," *Robotics and Autonomous Systems*, Jul. 2013.

[54] J. S. Dibangoye, C. Amato, O. Buffet, and F. Charpillet, "Exploiting separability in multiagent planning with continuous-state MDPs," in *AAMAS'14*, May 2014, pp. 1281–1288.

[55] K. Erol, "Hierarchical Task Network Planning: Formalization, Analysis, and Implementation," Ph.D. dissertation, 1996.

[56] S. Balakirsky, "Usarsim: a robocup virtual urban search and rescue competition," *Defense and Security Symposium*, 2007.

[57] A. Jacoff, E. Messina, B. A. Weiss, S. Tadokoro, and Yuki Nakagawa, "Test arenas and performance metrics for urban search and rescue robots," in *IROS'03*, 2003, pp. 3396–3403.

[58] G. Echeverria, S. Lemaignan, A. Degroote, S. Lacroix, M. Karg, P. Koch, C. Lesire, and S. Stinckwich, "Simulating Complex Robotic Scenarios with MORSE," in *SIMPAR*, 2012, pp. 197–208.

[59] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IROS'04*, 2004, pp. 2149–2154.

[60] M. Quigley, K. Conle, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," *ICRA workshop on open source software*, vol. 3, no. 3.2, p. 5, 2009.