A Decision-Theoretic Framework to Select Effective Observation Locations in Robotic Search and Rescue Scenarios

Francesco Amigoni, Nicola Basilico

***Abstract*— In some applications, like mapping and search and rescue, robots are autonomous when they are able to decide where to move next, according to the data collected so far. For this purpose, *navigation strategies* are used to drive the robots around environments. Most of the navigation strategies proposed in literature are based on the idea of evaluating a number of candidate locations according to an utility function and selecting the best one. Usually, *ad hoc* utility functions are used to provide a global evaluation of candidates by combining a number of criteria. In this paper, we propose to use a more theoretically-grounded approach, based on Multi Criteria Decision Making (MCDM), to define exploration strategies for robots employed in search and rescue applications. We implemented our MCDM-based exploration strategies within an existing robot controller and we experimentally evaluated their performance in environments used in the RoboCup Rescue Virtual Robots Competition.**

I. Introduction

Several robotic applications, including mapping [1] and search and rescue [2], require the availability of effective *navigation strategies* that drive an autonomous robot around an environment on the basis of the data collected so far. Most navigation strategies are based on the idea of evaluating a number of candidate locations according to an utility function. The candidate location that maximizes the utility function is selected as the next location the robot should reach. Navigation strategies differ in the way they implement the utility function. With few exceptions, utility functions are defined using *ad hoc* formulations that combine some criteria in a global evaluation of candidate locations. In a previous work, we proposed a more general decision-theoretic approach to define utility functions for navigation strategies. The approach, called Multi-Criteria Decision Making (MCDM), has been applied to define effective exploration strategies for mapping an environment [3].

In this paper, we apply the MCDM approach to the definition of effective *exploration strategies* for search and rescue applications. We consider a situation in which a single robot has to search an (initially unknown) environment for victims. No a priori knowledge about the possible locations of the victims is available, so we can safely reduce the problem of maximizing the number of victims found in a given time interval to the problem of maximizing the amount of mapped area in the same time interval. The robot operates according to the following steps: (a) it perceives the surrounding environment, (b) it integrates the perceived data within a map representing the environment known so far, (c) it decides where to go next, and (d) it goes there and starts again from (a). We propose to use MCDM for addressing step (c), namely for defining the exploration strategy the robot uses to search the environment. We implemented the proposed approach as a modification of a publicly available controller used for the RoboCup Rescue Virtual Robots Competition [4]. In this way, from the one hand, we can focus on exploration strategies (step (c)) exploiting an already tested framework for steps (a), (b), and (d) and, from the other hand, we can fairly compare our exploration strategies with those originally used in [4].

This paper is structured as follows. The next section reviews the relevant previous works on exploration strategies. Section III introduces the basics of the MCDM approach, which is applied to search and rescue settings in Section IV. Section V presents some experimental results and Section VI concludes the paper.

The authors are with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy {amigoni,basilico}@elet.polimi.it

II. Related works

Exploration strategies are a fundamental element of an autonomous mobile robot that explores an unknown environment for mapping it or for searching it without continuous human control. Let us first review the works that defined exploration strategies for mapping unknown environments. Besides very simple strategies that make the robots move along predefined trajectories [5], [6], the mainstream approach models exploration as a Next Best View (NBV) process, i.e., a repeated greedy selection of the next best observation location. The most important feature of an exploration strategy is how it evaluates candidate locations in order to select the best one. Usually, in NBV systems, candidate locations are chosen on the frontier between the known free space and the unexplored part of the environment in such a way they are reachable from the current position of the robot [7].

In evaluating a candidate location, different criteria can be used. A simple one is the distance from the current position of the robot [8], according to which the best observation location is the nearest one. Some works combine different criteria in an utility function defined *ad hoc*. For example, in [1] the distance $d(p)$ of a candidate location p, is combined with an estimate of the new information $A(p)$ acquirable from p and obtained by simulating a perception from p. The utility function is the following:

$$u(p) = A(p)e^{-\lambda d(p)}$$

where λ is a parameter balancing the relative weight of the two criteria.

Other examples of combination of different criteria are [9], in which distance is linearly combined with the expected reduction of the uncertainty in the map after the observation, and [10], in which a technique based on relative entropy is used. Also in [11], several criteria are proposed: distance, uncertainty in landmark recognition, number of visible features, length of visible free edges, rotation and quality of the path to the location. They are combined in a multiplicative function to obtain a global utility value.

The above strategies are based on aggregation functions that are defined *ad hoc* and are strongly dependent on the criteria they combine. In [12], the authors dealt with this problem and proposed a more theoretically-grounded approach based on multi-objective optimization in which the best candidate location is selected on the Pareto frontier. Besides distance and expected information gain, also overlap can be taken into account. This criterion is related to the amount of old information that will be acquired again from a candidate location. Maximizing the overlap can improve the performance of self-localization of the robot. In [3], MCDM, a general decision theoretic-method to combine different criteria [13], is proposed for exploration of indoor environments in order to build their maps. The work presented in this paper exploits the same approach in order to search for victims in rescue applications and will be described in Sections III and IV.

Exploration strategies play an important role also in search and rescue applications where a robot has not only to explore the environment (a disaster site), but also to find and communicate the position of victims. Indeed, assuming that no a priori information about locations of the victims is available, if an effective exploration is performed, the probability of finding victims increases. Therefore, developing good exploration strategies has been seen as a key objective also in search and rescue applications. A work that explicitly addressed the problem of defining an exploration strategy by introducing an utility function to combine multiple criteria is [4]. Authors propose to combine, within an *ad hoc* utility function, the distance, the expected information gain, and the probability of a successful communication from a candidate location. This strategy has been employed, with good results, in different RoboCup Rescue Virtual Competitions. In this work we experimentally compare the exploration strategies developed with our approach with that proposed in [4], which is explicitly devoted to the same goal.

III. Multi Criteria Decision Making

Following the mainstream approach, we model exploration as a sequence of greedy decisions autonomously made by the robot. Each decision determines a location on the frontier of the mapped space where the robot should take the next sensing action. The exploration strategy is the algorithm used by the robot to make such decisions. When designing an effective exploration strategy, the main challenge is to achieve a good long-term performance by means of short-term decisions that are made on the basis of a partial knowledge and of a number of criteria. To deal with this last problem, we propose to adopt a general decision-theoretic perspective, where the robot is considered as the decision maker and decisions are made according to a technique called Multi Criteria Decision Making (MCDM) [13], [14]. In this section we formally describe how MCDM works.

The formulation of the problem is straightforward: given a set of alternatives C, choose the "best" one among them. In the scope of this paper, alternatives will be candidate locations on the frontiers between the explored and unexplored space. Despite the simplicity of its formulation, the definition of "best" requires to address some significant issues.

The first issue is related to the features that can be employed in evaluating candidate locations. For example, a candidate location can be evaluated with respect to its distance from the robot's current position or with respect to an estimate of the new environment's area visible from there (expected information gain). In general, we will refer to these features as *criteria* and we will denote by $u_i(p)$ the *utility* of a candidate $p \in C$ with respect to criterion i. Utility is a measure of how good a candidate is with respect to the considered criterion. Without any loss of generality, we will assume that $u_i(p) \in [0, 1]$ and that the larger the utility the better a candidate. In this way, we have a common scale of evaluation for each criterion. If we assume to have n criteria denoted by the set $N = \{1, 2, \dots, n\}$, a candidate p can be associated to a vector of n elements, namely its utilities, $p = (u_1(p), u_2(p), \dots, u_n(p))$. Hence, the Pareto frontier of C can be determined as the largest subset $P \subseteq C$ such that for every $p \in P$ there does not exist a candidate $q \in C$ with $u_i(q) > u_i(p)$ for all $i \in N$.

Informally, providing a method to select a candidate on the Pareto frontier amounts to define the meaning of "best". The proposed MCDM approach solves this problem by providing a general way to define a global utility function, according to which a candidate on the Pareto frontier is selected. Global utility can be simply defined as a (non decreasing) aggregation function which combines all the utilities of a candidate to obtain an overall evaluation of it. We will denote global utility as $u : [0, 1]^n \rightarrow [0, 1]$. Examples of well-known aggregation functions are the arithmetic mean and the weighted average. With the definition of u, we can determine the Pareto optimal candidate that maximizes it, thus selecting the "best" candidate. In this sense, the function u encloses the definition of "best".

When combining multiple criteria, their dependency relation is an issue of paramount importance, which is often neglected when simple utility functions are employed. Indeed, criteria are not always independent, because there can be situations where two or more criteria are very similar or closely related one to each other. For example, think of criteria that are based on different methods to estimate the same feature (e.g., two criteria that estimate the distance of a candidate location from the current position of the robot according to Euclidean distance and according to Manhattan distance). Intuitively, when combining them into an aggregation function, their overall contribution to the global utility should be less than the sum of their individual ones. In this case, a *redundancy* relation holds between criteria. A dual situation occurs when two or more criteria are very different and, in general, can hardly be optimized together. In this case, a *synergy* relation holds between criteria, and their overall contribution should be considered larger than the sum of the individual ones. An example arising in our search and rescue scenario involves the estimated information gain and the overlap. Two criteria are synergic according to the idea that, since good utilities for both are very difficult to achieve in a single candidate, candidates that satisfy the two criteria reasonably well should be preferred to candidates that satisfy them in an unbalanced way. In order to consider these issues we need a way to define a function that accounts for redundancy and synergy between criteria. MCDM provides a general aggregation technique which can deal with this aspect: the *Choquet integral* [14].

We introduce a function[1] $\mu : \mathcal{P}(N) \rightarrow [0, 1]$ for which the following three properties hold:

- $\mu(\{\emptyset\}) = 0$;
- $\mu(N) = 1$;
- if $A \subset B \subset N$, then $\mu(A) \leq \mu(B)$.

That is, μ is a normalized *fuzzy measure* on the set of criteria N that will be used to associate a weight to each group of criteria. The weights specified by the definition of μ describe the dependency relations that hold for each group of criteria. More precisely, criteria belonging to a group $G \subseteq N$ are said to be:

- redundant, if $\mu(G) < \sum_{g \in G} \mu(g)$;
- synergic, if $\mu(G) > \sum_{g \in G} \mu(g)$;
- independent, otherwise.

The global utility u for a candidate p is computed as the discrete Choquet integral with respect to the fuzzy measure μ using p's utilities:

$$u(p) = \sum_{j=1}^{n} (u_{(j)}(p) - u_{(j-1)}(p))\mu(A_{(j)}) \quad (1)$$

where $(j) \in N$ indicates the j-th criterion according to an increasing ordering with respect to utilities, i.e., after that criteria have been permutated to have, for candidate p, $u_{(1)}(p) \leq \ldots \leq u_{(n)}(p) \leq 1$. It is also assumed that $u_{(0)}(p) = 0$. The set $A_{(j)}$ is defined in the following way, $A_{(j)} = \{i \in N | u_{(j)}(p) \leq u_i(p) \leq u_{(n)}(p)\}$. It is easy to prove that the candidate p that maximizes (1) belongs to the Pareto frontier P.

Global utility function expressed by (1) can represent, according to different definitions of μ, different aggregation functions. As an example, the weighted average is a particular case of the Choquet integral, obtained when all the criteria are independent one from another, i.e., when $\mu(G) = \sum_{g \in G} \mu(g)$ for every $G \subseteq N$.

In the following we will use the MCDM framework to define effective exploration strategies for search and rescue settings. We explicitly note that MCDM provides a theoretically-grounded way to *compose* criteria that contrasts with *ad hoc* composition methods proposed in literature, but it does not say anything about the individual criteria to be composed.

IV. MCDM-BASED EXPLORATION STRATEGIES FOR SEARCH AND RESCUE

To evaluate the MCDM approach in the search and rescue context, we implemented a MCDM-based exploration strategy in an already available controller. The reason for this choice is that we wanted to focus our activities mainly on exploration strategies defined with MCDM, without worrying about issues like building and keeping a representation of the environment. To select the controller, we looked at the participants to the RoboCup Rescue Virtual Robots Competition.

The RoboCup Rescue Virtual Robots Competition is a competition held at RoboCup, where different teams compete in developing simulated robotic platforms operating in simulated Urban Search And Rescue (USAR) scenarios. Robotic platforms are developed and executed within USARSim [15], an high fidelity 3D robot simulator. The competitions of the last years fostered the development of effective robot controllers whose code is publicly available. We built our MCDM-based exploration strategies within the controller developed by Amsterdam and Oxford Universities (Amsterdam Oxford Joint Rescue Forces[2]) for the 2009 RoboCup Rescue Virtual Robots Competition [16]. The main reasons for this choice are the good performance obtained by the Amsterdam-Oxford team during RoboCup and the fact that the available source code is easy to understand and extend. For a detailed and complete description of the controller we refer the reader to [4], [17]. Here, we present some features relevant to the scope of this paper.

The robotic platform used is a Pioneer P3AT, whose basic model and sensors are provided with the USARSim simulator. The map is two-dimensional and is represented with two superimposed occupancy grids, each one generated with a different laser range scanner. The first one is obtained with a small-range (typically 3 meters) conservative scanner and constitutes the *safe* area, i.e., the area where the robot can safely move. The second one is obtained from maximum-range scans (typically 20 meter) and constitutes the *free* area, i.e., the area which is believed to be free but not yet safe. Given a map represented in this way, frontiers between safe and free regions are extracted and considered as candidate locations in the exploration process. More precisely, the candidate locations are the middle points of the frontiers. A candidate p is evaluated according to the following criteria:

- $A(p)$ is the amount of free area beyond the frontier of p computed according to the free area occupancy grid;
- $P(p)$ is the probability that the robot, once reached the frontier of p, will be able to transmit data (such as the acquired data or the locations of victims) to the base station (whose position is a priori fixed within the environment), this criterion is computed from the distance between p and the base station;

[1] $\mathcal{P}(N)$ is the power set of N.

[2] `http://www.jointrescueforces.eu/`

- $d(p)$ is the distance between p and current robot's position, this criterion is calculated with different methods as we will describe later.

Given these criteria, the global utility for a candidate p is calculated with the following function:

$$u(p) = \frac{A(p)P(p)}{d(p)} \qquad (2)$$

we will refer to the exploration strategy using this global utility function as the "Visser" exploration strategy.

The original algorithm that uses (2) to select observation locations is developed for multirobot settings. It works according to the following steps:

1) compute the global utility of assigning each candidate p to each robot r using (2) where $d()$ is calculated using the Euclidean distance,
2) find the pair (r^*, p^*) such that the previously computed utility is maximum,
3) update the utility of (r^*, p^*) using again (2) but now computing $d()$ with a path planner,
4) if (r^*, p^*) is still the best candidate, then assign robot r^* to location p^*, otherwise go to step 2,
5) continue from step 2 while robots and candidate locations are available.

The reason behind the utility update of step 3 is that computing the real value of $d()$ with the path planner requires a considerable amount of time. Doing this for all the candidate locations would be not affordable in the rescue competition since a maximum limit on exploration time (20 minutes) is enforced.

We have substituted the exploration strategy (2) with a MCDM-based exploration strategy that works for a single robot setting. We now describe the changes we made to the original controller to include our MCDM-based strategies. We adopted the same criteria described above but we combined them with the MCDM approach, substituting function (2) with function (1) and introducing a weight function μ, as described in Section III. The need for normalized utilities forced us to compute the values of criteria for every candidate location. For example, when computing the utility related to the distance $d()$ we need the maximum and minimum values to normalize the utility, i.e.,

$$u_d(p) = 1 - (d(p) - \min_{q \in C} d(q))/(\max_{q \in C} d(q) - \min_{q \in C} d(q))$$

We chose to use a linear relative normalization because of the independence between robot's choices at different steps. Indeed, due to the greedy nature of the approach, the robot's decision at any step depends only on C and not on previous decisions and previous sets of candidate locations. Unfortunately, as already discussed, when $d()$ is computed using the path planner, a large computational time is required, making the 20 minutes limit too strict to achieve an acceptable performance. Hence, we decided to develop two MCDM-based strategies, called full MCDM (*f-MCDM*) and approximated MCDM (*a-MCDM*).

In the first strategy, f-MCDM, we compute the distance criterion using the path planner. In this way $d(p)$ corresponds to the real distance the robot should travel to reach p. To limit the computational time, we introduce a method to reduce the size of set C. We consider up to 10 different candidates, selecting them in increasing order of their Euclidean distance from the robot. Namely, the robot considers (at most) the 10 nearest candidates, according to the Euclidean distance. Moreover, we imposed a threshold over the computational time needed to compute the path to a candidate. When such time exceeds 15 seconds the computation is aborted and the corresponding candidate is skipped. In the second strategy, a-MCDM, no pre-selection is applied to the available candidates, i.e., the size of C is not reduced. All the utilities are computed considering $d()$ as the Euclidean distance. In both the methods, the best location is selected as the one that maximizes the MCDM global utility function.

The set of criteria we decided to use is the same used in the original controller (i.e., $A()$, $P()$, and $d()$ as described above.) Actually, we included a fourth criterion, namely the robot's battery level, denoted by $b()$. This last criterion was introduced for future experiments and it has not been used in the experiments reported here. We considered $u_b(p) = 1$ for all candidates p. Table I reports the chosen weights for the function μ.

$\mu(A) = 0.4$	$\mu(\{A, b\}) = 0.55$	$\mu(\{d, P\}) = 0.4$
$\mu(b) = 0.25$	$\mu(\{A, d\}) = 0.75$	$\mu(\{A, b, d\}) = 0.8$
$\mu(d) = 0.3$	$\mu(\{A, P\}) = 0.55$	$\mu(\{A, b, P\}) = 0.85$
$\mu(P) = 0.05$	$\mu(\{b, d\}) = 0.32$	$\mu(\{A, d, P\}) = 0.9$
	$\mu(\{b, P\}) = 0.28$	$\mu(\{d, P, b\}) = 0.4$

TABLE I
DEFINITION OF $\mu()$ FOR THE MCDM-BASED STRATEGIES

The weight values could be chosen by solving a mathematical programming problem, as described in [14], where a reduced set of parameters is required as input. However, we assigned such values manually in order to have more flexibility in adjusting them. We followed simple principles driven by the kind of relation we wanted to model between groups of criteria. For example, a synergy relation was introduced between $d()$ and $A()$, while criteria $b()$ and $d()$ were considered as redundant. Obviously, this manual method does not scale very well with the number of criteria. Indeed, the number of weights to be assigned is $2^n - 2$. This is a problem, since one of the advantages of using a theoretically-grounded approach like MCDM instead of *ad hoc* approaches is the possibility of adding new criteria in an easy way [3]. In situations where a large number of criteria is considered, automated methods to assign weights are a more suitable solution.

V. EXPERIMENTAL EVALUATION

In this section, we show some preliminary results obtained from evaluating the f-MCDM and the a-MCDM exploration strategies.

Implementing our MCDM-based exploration strategies in the controller of Amsterdam and Oxford Universities [4] has the important consequence that we can fairly compare them with the Visser exploration strategy (2), originally implemented in the controller. To have a consistent comparison, we used the same simulated environment of [4]. This setting, depicted in Fig. 1, was employed in the RoboCup Rescue Virtual Robots Competition of 2006 and it is called "Hotel Arena". It is an indoor environment composed of corridors and many rooms where victims are randomly spread.

Fig. 1. The map used for tests (red dots indicate starting locations)

We performed three different tests, using three different starting positions for the robot. They are reported as red dots in Fig. 1. In the first test the robot started from the bottom location, in second one from the north-east location, and in the last one from the north-west location. Metrics used in the evaluation are the total amount of free and safe area mapped after a 20 minutes run and the total number of visited locations, i.e., the total number of decisions made by the robot (these two metrics are commonly used to evaluate exploration strategies, see, for example, [18]). In addition we report also the number of found victims. However, the value of this metric is not as significant as the amount of mapped area, since no a priori information about the victims locations is exploited in the selection of observation locations.

	Visser	f-MCDM	a-MCDM
locations	1036	776	885
free area (m^2)	499.29	434.34	495.83
safe area (m^2)	241.66	175.84	187.25
victims	7	2	2

TABLE II
RESULTS FOR TEST 1

Tables II, III, and IV show the results obtained by the three exploration strategies in the first, second, and third test respectively.

In test 2 (Table II), the Visser strategy achieves the best performance. Despite the pre-selection of the candidates, f-MCDM is strongly penalized by the long time needed to compute the path lengths. This is also confirmed by the relatively small number of visited locations. As a consequence, the total area covered in the 20 minutes of the run is limited. Also the a-MCDM strategy is outperformed by the Visser's one in this case. What emerges here is a clear example of how using a too simple estimate (i.e., the Euclidean distance) can penalize performance. Indeed, the a-MCDM strategy selects locations that, once the real path to reach them is computed, are not as close as expected. For instance, we observed that the exploration of a corridor has been interrupted to continue in the parallel one. In the other two tests, a-MCDM outperforms the other two strategies. In these cases, minimizing the time to make a decision turns out to be the most important factor. Particularly interesting is the large difference between the number of locations visited by a-MCDM and by the other strategies. In Fig. 2, the maps generated by the robots in the second test are reported, as they are visualized in the controller (unknown area is blue, free area is white, and safe area is grey; the path followed by the robot is depicted as a green dotted line).

	Visser	f-MCDM	a-MCDM
locations	547	562	987
free area (m^2)	449.68	378.96	556.5
safe area (m^2)	202.98	167.22	272.66
victims	2	2	4

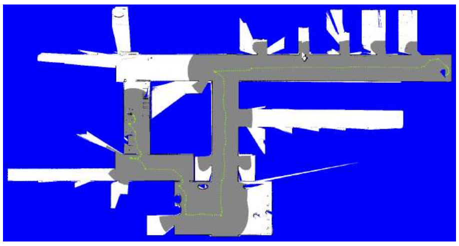
TABLE III
RESULTS FOR TEST 2

	Visser	f-MCDM	a-MCDM
locations	602	506	953
free area (m^2)	385.35	400.33	525.26
safe area (m^2)	181.07	147.26	243.07
victims	2	2	2

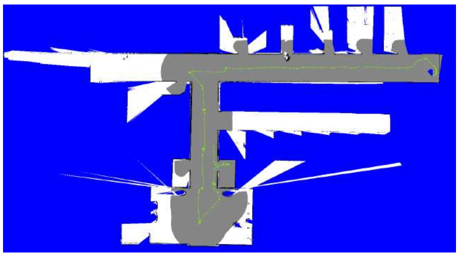
TABLE IV
RESULTS FOR TEST 3

Although these results are still very preliminary, some remarks can be attempted. The Visser strategy mostly aims at driving the robot to explore cluttered spaces, such as rooms. Differently, MCDM-based strategies are likely to select zones with wide open spaces. From the one hand, this behavior could provide an advantage in terms total mapped area; from the other hand, it can require a lot of time to check for victims within rooms or near end zones of the environment (such as corners). The a-MCDM strategy is very fast in making decisions but requires the robot to cover unnecessary long distances (i.e., to visit a lot of locations). This could be a drawback with respect to efficiency in power consumption (this is why we plan to introduce a criterion related to battery). The f-MCDM strategy avoids this problem allowing the robot to make more informed decisions. However, the long time needed to compute utilities makes this strategy not completely suitable for search and rescue scenarios with time constraints.

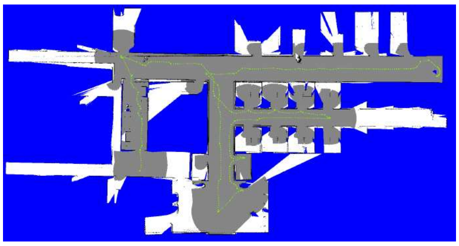
From these preliminary experimental activities we identified two main issues that we shall address. The first one is the need to avoid the problem of incomplete exploration of some parts of the environment. This can be solved by introducing



(a) Visser



(b) f-MCDM



(c) a-MCDM

Fig. 2. Maps generated in the second test

some criteria related to high-level knowledge extracted from the map, e.g., recognizing a room or a corridor. The second requirement is related to the use of more accurate estimates for the candidate locations' distances. A possible method to provide good estimates of $d()$ within a reasonable amount of time could exploit a topological representation of the map. Selected observation locations could be stored in a graph data structure, that is used to compute paths.

VI. Conclusions

In this paper, we have presented the application of the MCDM decision-theoretic approach to the definition of exploration strategies for search and rescue. Preliminary experimental results have shown that MCDM-based exploration strategies offer a performance comparable with that obtained with exploration strategies based on *ad hoc* utility functions.

Encouraged by these initial results we are conducting a more complete experimental analysis of the potential of MCDM-based exploration strategies, involving other environments, other exploration strategies (not necessarily devised for search and rescue), and multiple robots. Our hope is to obtain more conclusive evidence that assesses the viability of the MCDM approach, which represents a theoretically-grounded approach to define exploration strategies that contrasts with those proposed in literature so far. We are also planning to make our code publicly available for fostering experimental reproducibility and comparison. Other future works include the development of automatic techniques to set the values of weights in MCDM, in order to ease the inclusion of other criteria in the evaluation of candidate locations, and the application of MCDM-based strategies to other domains, for example to patrolling.

VII. Acknowledgements

The authors would like to thank Michele Medioli for his contributions to the experimental activities.

References

[1] H. Gonzáles-Baños and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, 2002.

[2] S. Balakirsky, C. Scrapper, S. Carpin, and M. Lewis, "Usarsim: a robocup virtual urban search and rescue competition," in *Proc. of SPIE*, vol. 6561, 2007, p. 65611M.

[3] N. Basilico and F. Amigoni, "Exploration strategies based on multi-criteria decision making for an autonomous mobile robot." in *Proc. ECMR*, 2009, pp. 259–264.

[4] A. Visser and B. A. Slamet, "Including communication success in the estimation of information gain for multi-robot exploration," in *Proc. WiOPT*, 2008, pp. 680–687.

[5] E. Bourque and G. Dudek, "Viewpoint selection - an autonomous robotic system for virtual environment creation," in *Proc. IROS*, 1998, pp. 526–532.

[6] J. Leonard and H. Feder, "A computationally efficient method for large-scale concurrent mapping and localization," in *Proc. Int'l Symposium on Robotics Research*, 1999, pp. 169–176.

[7] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. CIRA*, 1997, pp. 146–151.

[8] B. Yamauchi, A. Schultz, W. Adams, and K. Graves, "Integrating map learning, localization and planning in a mobile robot," in *Proc. ISIC*, 1998, pp. 331–336.

[9] C. Stachniss and W. Burgard, "Exploring unknown environments with mobile robots using coverage maps," in *Proc. IJCAI*, 2003, pp. 1127–1134.

[10] F. Amigoni, V. Caglioti, and U. Galtarossa, "A mobile robot mapping system with an information-based exploration strategy," in *Proc. ICINCO*, 2004, pp. 71–78.

[11] B. Tovar, L. Munoz-Gomez, R. Murrieta-Cid, M. Alencastre-Miranda, R. Monroy, and S. Hutchinson, "Planning exploration strategies for simultaneous localization and mapping," *Robotics and Autonomous Systems*, vol. 54, no. 4, pp. 314 – 331, 2006.

[12] F. Amigoni and A. Gallo, "A multi-objective exploration strategy for mobile robots," in *Proc. ICRA*, 2005, pp. 3861–3866.

[13] M. Grabisch, "The application of fuzzy integrals in multicriteria decision making," *European Journal of Operational Research*, vol. 89, no. 3, pp. 445–456, 1996.

[14] M. Grabisch and C. Labreuche, "A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid," *4OR A Quarterly Journal of Operations Research*, vol. 6, no. 1, pp. 1–44, 2008.

[15] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper, "Usarsim: a robot simulator for research and education," in *Proc. ICRA*, 2007, pp. 1400–1405.

[16] A. Visser, G. de Buy Wenniger, H. Nijhuis, F. Alnajar, B. Huijten, M. van der Velden, W. Josemans, B. Terwijn, R. Sobolewski, H. Flynn, and J. de Hoog, "Amsterdam Oxford joint rescue forces - team description paper - Virtual Robot competition - Rescue simulation league - robocup 2009," in *Proceedings CD of the 13th RoboCup Symposium*, June – July 2009.

[17] A. Visser, M. V. Ittersum, L. A. G. Jaime, Laurentiu, and A. Stancu, "Beyond frontier exploration," in *Proc. of the 11th RoboCup International Symposium*, 2007, pp. 113–123.

[18] F. Amigoni, "Experimental evaluation of some exploration strategies for mobile robots," in *Proc. ICRA*, 2008, pp. 2818–2823.