On AUV actions to correctly label world information

Francesco Maurelli, Zeyn Saigol, David Lane
Ocean Systems Laboratory
School of Engineering and Physical Sciences
Heriot-Watt University
EH14 4AS Edinburgh, Scotland (UK)
Email: f.maurelli@ieee.org

Michael Cashmore, Bram Ridder, Daniele Magazzeni
Department of Informatics
Kings College Londons
WC2R 2LS, London, UK
Email: firstname.lastname@kcl.ac.uk

***Abstract*—An Autonomous Underwater Vehicle (AUV) needs to demonstrate a number of capabilities, in order to carry on autonomous missions with success. One of the key areas is correctly understanding the surrounding environment. However, most of the state-of-the-art approaches in labelling world information are based on the analysis of a single frame, whilst - especially in scenarios where the vehicle interact with complex structures - there is the need of sensor data from multiple views, in order to correctly classify world information. This paper presents an active approach to solve this problem, with a tree-based (path)-planner which makes the vehicle executing a specific set of actions (following a specific trajectory), in order to discriminate among several possibilities. Results in simulation, with varying parameters, have shown that the algorithm is always bringing the robot to locations where it is expected that measurements would be different, according to the different environment.**

I. Introduction

The ability to recognise interesting features, objects and structures in the operating environment is becoming more and more important in the underwater robotics community. Whilst Remotely Operated Vehicles (ROVs) are routinely used for intervention tasks across the world, the use of Autonomous Underwater Vehicles (AUVs) is on the other hand mainly limited to surveys in a obstacle-free environment. However, in the ocean technology research community, interest is growing in robots able to operate persistently and autonomously in dynamic, partially-unknown and partially-observable environments. As part of the European project FP7 PANDORA [1], [2], a key focus for our research group goes towards a correct recognition of semantic concepts in the operating environment, in order to have a high-level representation, over which the planning system can reason and plan. This represents a key enabler for persistent autonomy in the marine robotics domain, which means the capability for the robot to robustly operate in those challenging environments. In order to achieve the high-level goal of AUV mission involving active intelligent decisions, correctly labelling the world, i.e. to maintain a semantic descriptions of it, alongside the geometric one, becomes an important ability.

Most of the state-of-the-art techniques for AUV navigation do not address the semantic aspects of the world and the ambiguity of possible features. Matching a sensor reading with a specific semantic concept is a quite difficult task and is often impossible with just one sensor reading. Complex objects and structures need to be observed from several points in order to be correctly classified. In addition to the noise in the sensor measurements, a partial view of an object might not be enough to associate it with a specific concept without any doubt.

There is therefore the need for the robot to undertake deliberate actions in order to plan its path close to specific objects and structures, in order to understand them and to correctly label them. According to the specific semantic concept recognised, the planner system can therefore intervene and update its original plan.

Related Work

Correctly labelling underwater object is a topic which has been widely researched, especially in Mine-Counter-Measure (MCM) missions [3], [4], [5]. However the focus of those work was mainly into correctly label a portion of the sonar image, i.e. the part which represents a possible threat and can be classified. The significant difference in the presented work relay on the AUV operating close to and interacting with complex objects and structures, such as a single sonar frame is not enough to correctly identify it. In order to represent the information about the world, the vehicle needs to maintain a proper ontology, as a mean to reason on concepts and evolutions. Work on ontologies for underwater vehicles can be found in [7], from Miguelanez *et al.*. A previous work from the author [6] addresses the uncertainty linked to processing acoustic images and its impact at ontological level, as well as other types of uncertainty. No approach however considers the integration of an active system in order to aid the classification process. A single view from any sensor, being it a sonar, a camera, or others, is often not enough to classify and specific strategies need to be employed. A previous work from Maurelli and Petillot [8] addresses the active localisation problem, i.e. the ability for the robot to actively choose a set of actions in order to localise itself. That problem represents a mirror problem to the one addressed in this paper. Whilst in [8] the world was fully known, but the vehicle's state was a finite set of possibilities, in this case the vehicle's state is known, but the world is represented by a finite set of possibilities.

Contribution

This paper aims to present a novel approach to correctly label world information, through the active choice of a specific path which would be most beneficial in order to discriminate among several possibilities. A passive navigation approach would not guarantee a correct classification of the world as no specific information is sought. As outlined in Figure 1, after one measurement (or more measurements from the same

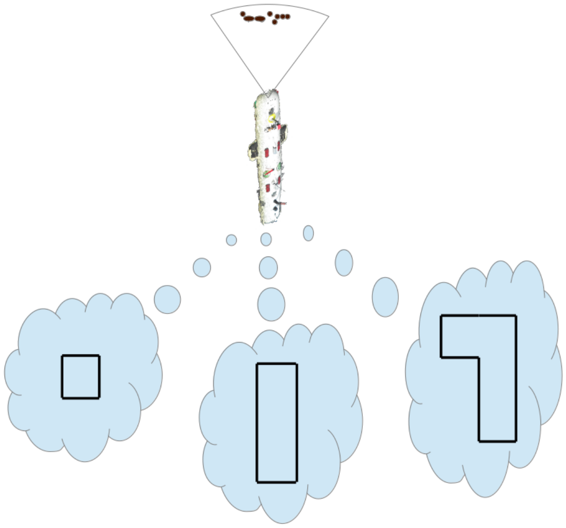


Fig. 1. A single measurement from a sensor is not enough to correctly label the world. The same sensor reading in the Figure can belong to any of the three objects below the robot. The actions undertaken by the robot can significantly vary according to the type of object and therefore a specific active navigation strategy needs to be employed.

position), the robot does not know which is the structure in front of it, and therefore needs to employ an active strategy, involving path planning and navigation, in order to correctly label it. The same sensor measurement can represent any of the three objects in the Figure. The paper is organised in the following way:
- Section II will present the Active Path Planning Module; - Section III will present numeric results in simulation, including comparison with other techniques; - Section IV will conclude and highlight the future work.

II. Active Path Planning Module

The Active Path Path Planning Module is triggered when the AUV has the need to understand which object or structure is in the world and when the initial sensor data are not enough to discriminate among several options. The module provides in output a specific path to follow in order to discriminate between them. The general principle of the module is to find a set of actions that can maximise the information gain. Basic actions a_i that the vehicle can perform are identified:

$$\mathbb{A} = \{a^1; a^2; ...; a^n\} \tag{1}$$

The actions a^i are on the format: "*go forward for x meters*", "*go backwards for x meters*", "*go left for x meters*", "*go right for x meters*", "*go up for x meters*", "*go down for x meters*", "*turn x deg clockwise*", "*turn x deg anticlockwise*".

The module produces in output a list $a_{t_0}, ..., a_{t_s}$ which represents the s actions selected to be executed at times $t_0, ..., t_s$. It is important to stress that choosing only the best single move is not enough. It is often impossible to understand which object is in front of the vehicle with only one basic step, while it is more often true that only a more complex trajectory can do that, in order to break possible similarities of the different options. A simple greedy approach of building up a new move on top of the previous best move is again not suitable, due to the possibility of local minima and local maxima. The proposed approach is thus exploring a tree of basic actions, expanding each node and calculating the reward for each action. For each cluster, a tree is built having the root initialised to the centroid position and orientation. The complexity of the tree exploration is polynomial on the number n of actions and exponential on the depth d of the tree ($O(n^d)$). However, it is possible to reduce this complexity, considering that for every basic action a^i there is another basic action a^j which produces the opposite effect. As visiting a location already visited is not providing any new information, each node will not expand the action which balances the previous one. Following the same principle, loops on the same root-to-node path are not allowed, thus reducing the final complexity. It is possible to set other constraints, in order to cut the tree. They can be related to the specific vehicle used, and to some manoeuvres which should be avoided. For simplicity, we have decided not to add any other constraints, but they are easily pluggable in the module. The presented approach deals with the information gain acquired after executing the actions $a_{t_0}, ..., a_{t_s}$. It is represented by the diversity in the simulated measurements z_t^k for each of the possible objects k, at time t, after executing all the previous actions linking the root to the node. Considering the measurements z^k represented by an array of distances (after processing the sonar $\rho\theta$ data), the reward function for a single node n executing the action a^i becomes:

$$r_{a^i}(n) = \frac{\sum_{j=1}^{m} \sigma^2_{z^{j,k}_{a^i}(n)}}{m} \tag{2}$$

where m represents the length of the array and $z^{j,k}_{a^i}(n)$ represents the measurements taken from the position of the node n, for the object k rototranslated according to the actions $a_0, ..., a_{t-1}$ (path root-node). It represents the average of the variance of the measurements z from the different objects k for each index j of the measurements. An example is in Figure 2. Each action has a cost associated, which may vary according to the specific constraints. For example, it is reasonable to assume that if the action a_t is the same than the action a_{t+1}, the cost for the vehicle is less than a completely new action. This is because there is no need to radically change the thruster behaviour. The total value assigned to a single node is given by the difference between the reward r and the cost c. The output of the module is therefore:

$$p^* = \{a^*_{t_0}, ..., a^*_{t_s}\} = argmax_{p_i}(r_{p_i} - c_{p_i}) \tag{3}$$

where p_i represent the path i^{th} and p^* represents the best path. The maximum depth of the tree is fixed *a-priori*. In a closed environment, it can be set to infinite because each branch of the tree is expanded until no further new location can be reached.

Summarising, the general principle of the module is to find a path that maximises the diversity in the observations for the different possible objects. A trajectory tree is built from the initial AUV position. Each node represents a possible basic move. The output of the module is a path root-leaf (i.e. a sequence of basic moves) which maximises the diversity in the simulated observations and thus helping to correctly label the object. The next two sections present in more details some aspects of the algorithm, like an explanation about vehicle

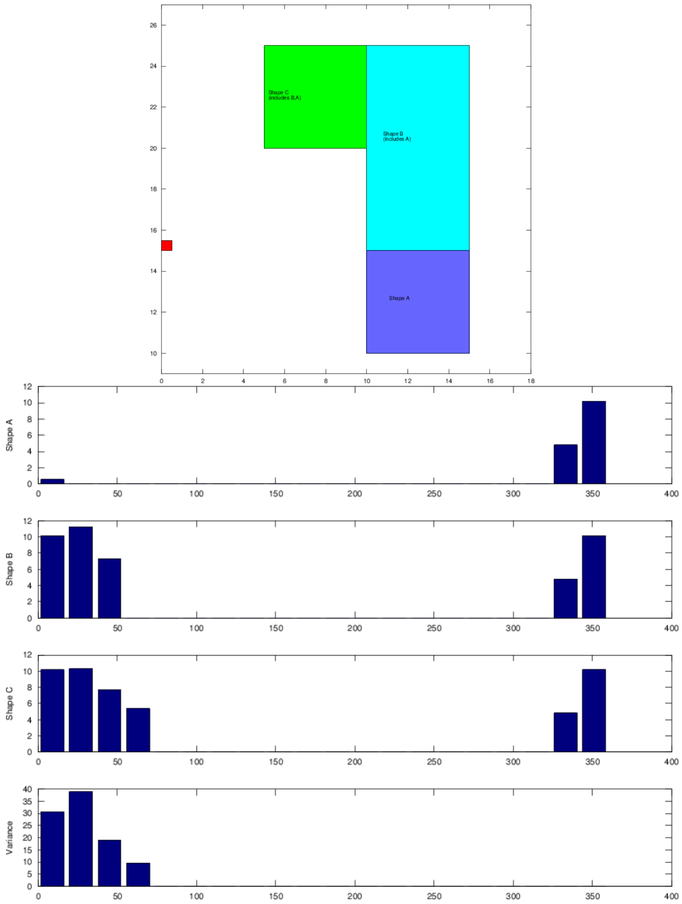


Fig. 2. An explanation of the gain function: The vehicle is positioned as in the Figure on top. From that position, it simulates the sonar sensor for each of the three possibilities, getting an array of distances. For each beam (index in the array), the variance is calculated (bottom set) and then averaged according to the number of beams. This value represents the gain of being in that position.

safety and obstacle avoidance, and an analysis of the stopping criteria of this algorithm and the completeness of the results generated.

A. *Dealing with obstacles*

A key safety constraint for the robot is to avoid obstacles whilst performing a task. This is very relevant in this case as well, because the robot does not know the world and a certain path might be obstacle-free in some cases, but impracticable in other cases. Specific sets of actions are chosen to solve the disambiguation among possible worlds, but these should ensure the robot safety. Considering that the possible world are known in advance - the robot needs to label a partially viewed object from a predefined set of possibilities, the only trajectories that can be generated are those who satisfy safety constraints for all the possible worlds. This has the side effect to further cut branches of the tree, thus making the algorithm more efficient. Trajectories which would bring the robot close to obstacles for one world and in free space for another world are to be preferred, as the reward will be greater.

B. *Stopping criteria*

The stopping criteria for the exploration of the tree are as follow:

1) a predefined depth is reached. In this case the best path is given to the path execution system;
2) the reward for a node has reached a specific value. That means that the path up to that node is able to disambiguate among the different possible objects.

The ability to stop the tree exploration at any depth is very important for in-mission executions, where time-constraints do not allow prolonged reasoning.

Can the algorithm guarantee convergence after stopping criteria is reached? The easy answer is not in all cases, like all solutions which involve robot navigation. Robot navigation is very dependent on the environment and there can be environments where the motion is severely constrained and therefore the vehicle cannot reach a point to disambiguate among the possible initial hypotheses. Another very important factor is given by the algorithm parameters: a basic movement which is too big might prevent the vehicle from being in the right spot to see the differences among the possible worlds, especially if this is in pair with a short range of the measurements. The algorithm however always terminates providing the best available trajectory - if any. In open environments, it would terminate reaching the final depth of the tree, whilst in close environments - even without considering the depth of the tree, at a certain point, it would not be able to expand any more leaf of the tree, without visiting an already visited location, which is forbidden. It is possible however to detect cases where it is not possible to correctly label the world, given the type and amount of information available, and therefore to notify to the upper vehicle's layer about the impossibility to classify the object with certainty. This happens when the reward for each explored node is below a certain noise threshold, thus practically detecting similarities in the possible worlds which are impossible to solve.

III. Simulated Results

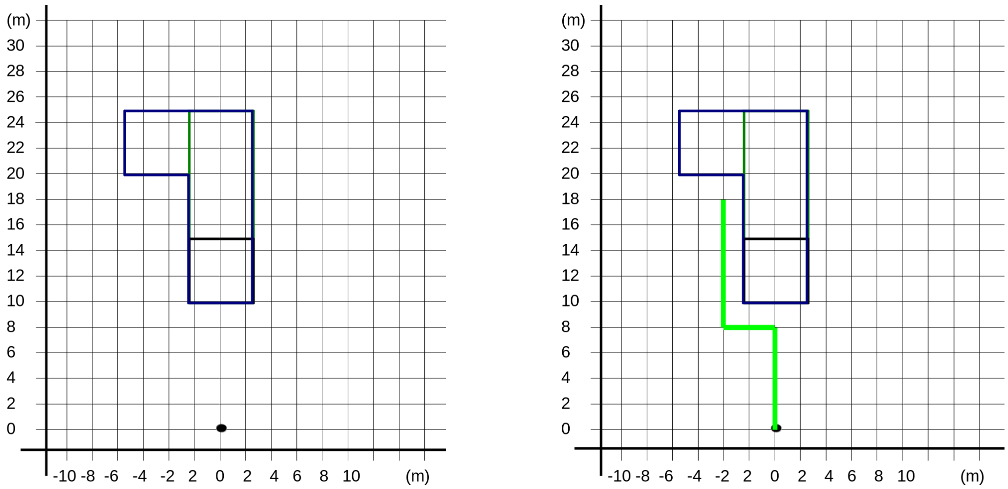
The scenario used is the one outlined in Figure 1. There are three possible labels to the object partially perceived by the robot. The goal is to arrive to a position where the simulated measurements for the three possibilities are as different as possible, in order to help a correct identification. The vehicle is equipped with a sonar, with scanning capability of $360\,\mathrm{deg}$. The set of basic actions is represented by:

$$\mathbb{A} = \text{move}\{\text{forward, backwards, left, right}\} \qquad \text{for x meters} \tag{4}$$

The length x of a single move is constant at current stage of the work. Considering that the sonar has a full field of view, no rotation was considered in the set of basic actions. Hydrodynamic considerations are also out of the scope of this paper. Once the algorithm generates the path to follow, composed by a set of waypoint, the vehicle can freely choose the best low-level path among waypoints. In the case in which the field of view is limited, like for example using a forward looking sonar, then the rotation basic actions are necessary. In the presented set-up, the vehicle is positioned in front of

Step (columns) Sonar range (rows)	2m	4m	6m	8m	12m
4m	1 1 1 1 3 3 1 1 1 1 1	1 1 3 1 1 3 1 1	1 3 1 1	1 3 1 1	no solution
8m	1 1 1 1 3 3 1 1 1 1 1	1 1 3 1 1 3 1 1	1 3 1 1	1 3 1 1	3 1 1
12m	1 1 1 1 3 3 1 1 1 1 1	1 1 3 1 1 3 1 1	1 3 1 1	1 3 1 1	3 1 1
16m	1 1 1 1 3 3 1 1 1 1 1	1 1 3 1 1 3 1 1	1 3 1 1	1 3 1 1	3 1 1
20m	1 1 1 1 3 3 1 1 1 1	1 1 3 1 1	1 3 1 1	1 3 1 1	3 1 1
30m	1 1 1 1 3 3 1 1 1	1 1 3 1 1	1 3 1 1	1 3 1	3 1 1
50m	1 1 1 1 3 3	1 1 3	1 3	1 3	3 1

TABLE I. TRAJECTORIES GENERATED ACCORDING TO THE CHANGE OF TWO PARAMETERS: THE SONAR RANGE AND THE BASIC GRID STEP. 1 MEANS *forward*, 2 MEANS *backwars*, 3 MEANS *left*, 4 MEANS *right*



(a) Map

(b) Path

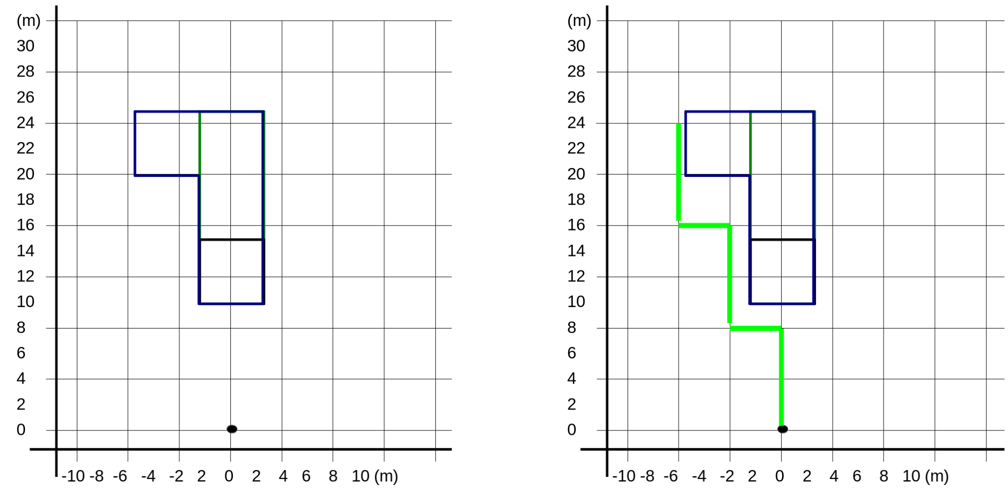
Fig. 3. (a) Vehicle's position with a grid showing the possible actions. (b) Path chosen by the robot to understand the environment. Each basic movement has a length of $2m$.

the object, at a distance of $10m$. Many tests were performed, changing two parameters:

- the sonar range, from $4m$ up to $50m$
- the length of a single step in the trajectory, from $2m$ up to $12m$.

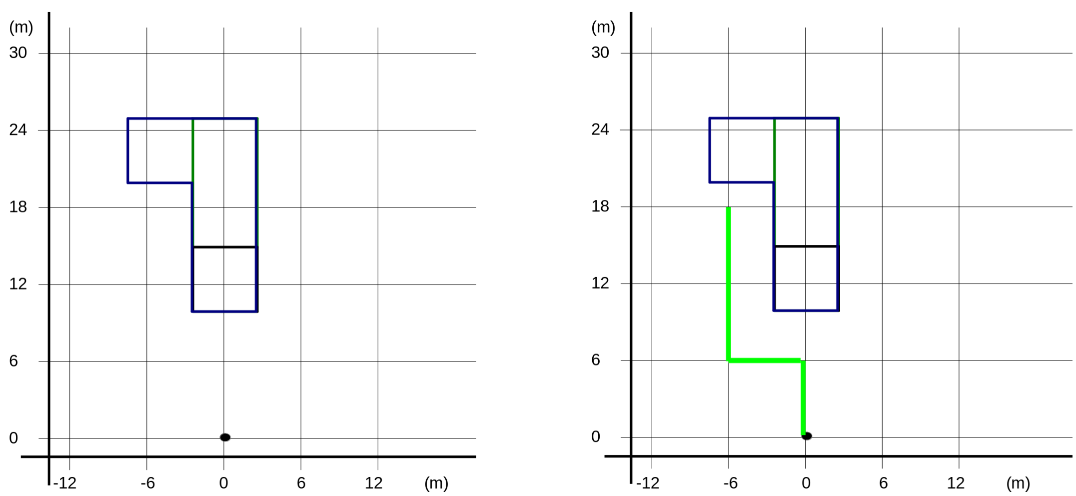
The trajectories generated are summarised in Table I. The trajectories generated are in the form of a sequence of action, each action being identified by a number. In this case, 1 represents a movement forward, 2 represents a movement backward, 3 represents a movement on the left, and 4 represents a movement on the right. As it is possible to see interpreting the data from the table or looking at Figures 3, 4, 5, 6, 7, the vehicle always tries to reach a point in the map where the observations are mostly different for each of the possible objects. In all cases, this means moving on the left and forward to arrive to a place where most differences could be perceived. Naturally the choice of the parameters influence the generated path, but as it can be seen, there are no great differences. The location the vehicle is aiming for remains always in the same area of the map. The change of sonar range influences the size of the generated path. The more the vehicle can see, the less it needs to travel in order to spot the differences among the possible scenarios. Similarly, the size of each basic movement influences the length of path - intended as number of basic movements. This does not always have an impact on the geometric size of the path, which could remain pretty much similar - or even the same - whilst changing the grid size. Having a smaller basic movement gives the possibility to model the possible movements in greater details and reach positions which would not be reachable otherwise. On the other hand, the gain on a more detailed possible robot position/locations is paid with higher computational cost expanding the tree.

Looking at the different trajectories, it can be seen that the algorithm returns a path which tries to follow the structure, and tries to end as close as possible to one of the possible world, as this is a very good place to disambiguate among different options. Considering this, a question would naturally arise: would it be sufficient to instruct the robot to simply follow a structure, in order to correctly label it? Although in some cases it would be sufficient, the resulted path might not be as efficient as the one produced by the algorithm presented. First of all, it would be important to decide in which direction the structure would need to be followed. In the example provided, similar results would have been reached if a *clockwise* direction is chosen. In case the real object is a rectangle or an L-shaped object, choosing a *anti-clockwise* trajectory would have resulted in a much more long route to undertake, in order to correctly assign a label to the object. There are also cases in which following the structure is always a more expensive choice than the proposed approach, even if the best direction is chosen. This is the case for example for complex non-convex structures, where a direct path would be more efficient than following step by step the object surface.



(a) Map (b) Path

Fig. 4. (a) Vehicle's position with a grid showing the possible actions. (b) Path chosen by the robot to understand the environment. Each basic movement has a length of $4m$.



(a) Map (b) Path

Fig. 5. (a) Vehicle's position with a grid showing the possible actions. (b) Path chosen by the robot to understand the environment. Each basic movement has a length of $6m$.

IV. Conclusions and Future Work

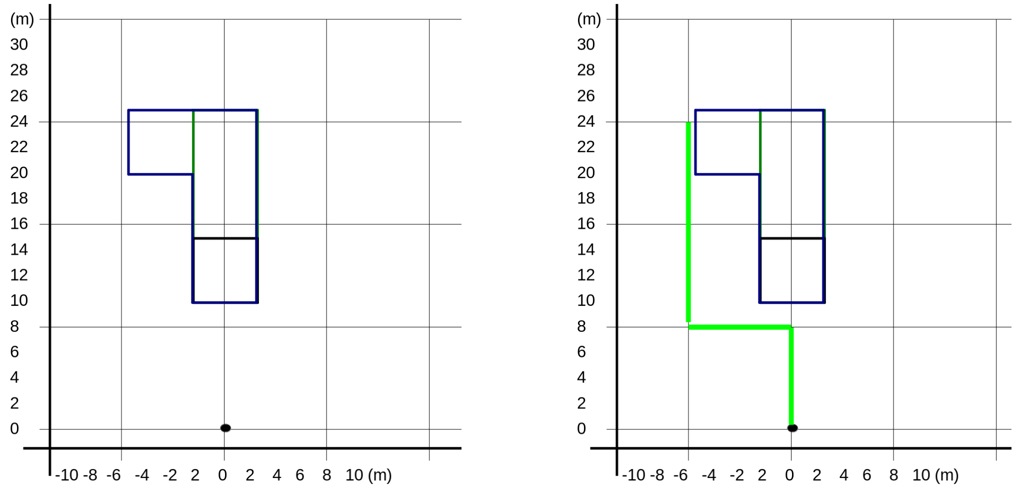
This paper has presented a novel approach in order to correctly label world information. Although the scenario envisaged is for underwater robotics, the proposed approach can be applied in other fields as well. The vehicle has initially only a partial view of the object. Analysing this view, it determines a finite set of possible objects. Through a tree-based path planner, the vehicle then is able to determine the best path in order to disambiguate among different world and correctly label the objects. For simplicity, only 2D environments have been tested, in order to reduce the set of basic moves. The 3D extension is very straightforward, as the only change is the number of elements in the set of possible actions. Conceptually and practically, there is no difference, apart from the computational time, which is however relatively low. An interesting development for this work would be to be able to change the length of each basic move dynamically, without it being predefined. This would give the possibility to have long movements in relatively free space and shorter movements close to the structure being inspected, and therefore being more computationally efficient.

Acknowledgment

The authors would like to thank all the other members of the Ocean Systems Laboratory at Heriot-Watt University. The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 - Challenge 2 - Cognitive Systems, Interaction, Robotics - under grant agreement No 288273 - PANDORA

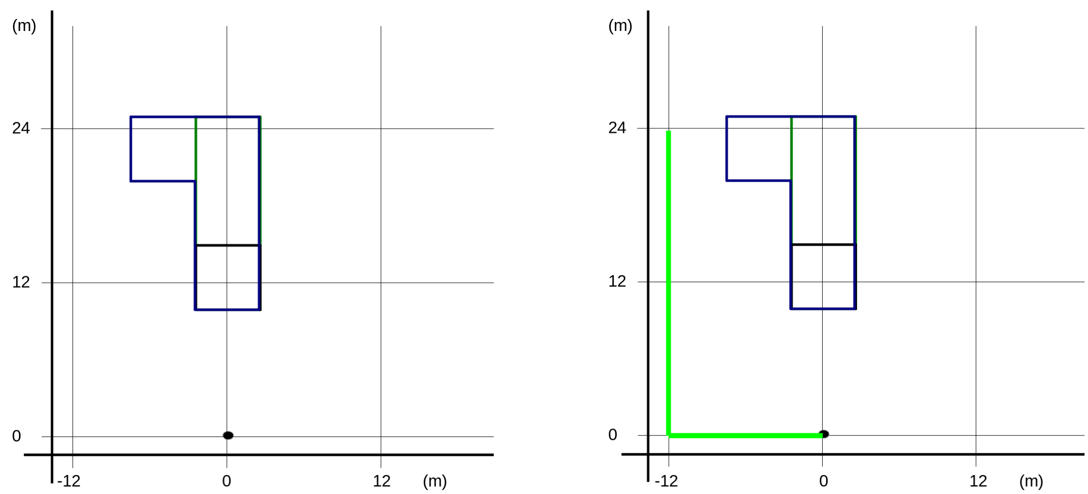
References

[1] David M. Lane, Francesco Maurelli, Tom Larkworthy, Darwin Caldwell, Joaquim Salvi, Maria Fox, and Konstantinos Kyriakopoulos. PANDORA:



(a) Map (b) Path

Fig. 6. (a) Vehicle's position with a grid showing the possible actions. (b) Path chosen by the robot to understand the environment. Each basic movement has a length of $8m$.



(a) Map (b) Path

Fig. 7. (a) Vehicle's position with a grid showing the possible actions. (b) Path chosen by the robot to understand the environment. Each basic movement has a length of $12m$.

Persistent autonomy through learning, adaptation, observation and replanning. In *Proceedings of IFAC NGCUV 2012 - Navigation, Guidance and Control of Underwater Vehicles*, 2012.

[2] David M. Lane, Francesco Maurelli, Petar Kormushev, Marc Carreras, Maria Fox, and Konstantinos Kyriakopoulos. Persistent autonomy: the challenges of the PANDORA project. In *Proceedings of IFAC MCMC 2012 - Manoeuvring and Control of Marine Craft*, 2012.

[3] Francesco Maurelli, Pedro Patrón, Joel Cartwright, Jamil Sawas, Yvan Petillot, and David Lane. Integrated MCM missions using heterogeneous fleets of AUVs. In *Proceedings of IEEE-MTS Oceans'12, Yesou, South Korea*, 2012.

[4] Jamil Sawas and Yvan Petillot. Cascade of boosted classifiers for automatic target recognition in synthetic aperture sonar imagery. In *Proceedings of Meetings on Acoustics*, volume 17. Acoustical Society of America, 2013.

[5] M.R. Azimi-Sadjadi, De Yao, Qiang Huang, and G.J. Dobeck. Underwater target classification using wavelet packets and neural networks. *Neural Networks, IEEE Transactions on*, 11(3):784–794, May 2000.

[6] Francesco Maurelli, Zeyn Saigol, Georgios Papadimitriou, Tom Larkworthy, Valerio De Carolis, and David Lane. Probabilistic approaches in ontologies: joining semantics and uncertainty for AUV persistent autonomy. In *Proceedings of IEEE-MTS Oceans'13, San Diego, USA*, 2013.

[7] Emilio Miguelanez, Pedro Patron, Keith E. Brown, Yvan R. Petillot, and David M. Lane. Semantic knowledge-based framework to improve the situation awareness of autonomous underwater vehicles. *IEEE Transactions on Knowledge and Data Engineering*, 23(5):759–773, May 2011.

[8] F. Maurelli and Y. Petillot. Particle diversity reduction for AUV's active localisation. In *IEEE Oceans'10, Sydney*, 2010.