A tree-based planner for active localisation: applications to Autonomous Underwater Vehicles

Yvan Petillot[1], Francesco Maurelli[1]
[1]School of Engineering and Physical Sciences
Heriot-Watt University
Edinburgh, EH14 4AS
email: Y.R.Petillot@hw.ac.uk

***Abstract*—Autonomous Underwater Vehicle (AUV) are moving to a new phase with the development of light intervention systems. New vehicles will be equipped with lightweight manipulators and operate around subsea infrastructures. One of the key capabilities to safely perform such mission is robust and accurate autonomous localisation, i.e. the ability for the AUV to estimate correctly its position and orientation in the environment. Most of the current approaches to localisation are "passive", i.e., with no active control of the vehicle to improve localisation performances based on the current knowledge of the environment and the current estimate of the vehicle position. The "active" localisation framework aims at incorporating the control of the robot motion in the localisation process by finding the best path to follow in order to reduce the uncertainty in the position state estimation. This paper aims at presenting a novel approach to the active localisation problem underwater using a priori maps of the environment or maps previously built using SLAM or mosaicing techniques. This is very relevant to the Trident project which aims at developing and demonstrating technologies for light intervention using an AUV. In the proposed framework, the position of the vehicle is estimated using Monte Carlo localisation techniques (the state of the vehicle is represented by particles) and the motion of the vehicle is optmised to reach a single cluster of the particles (the vehicle knows where it is) by minimizing the expected entropy of the move. Both simulation results and tank trials showing the advantages of using this technique in realistic environments are presented here.**

I. Introduction

Underwater robots now play a key role in the exploitation of marine resources (offshore), conservation of marine environments (environment assessment) and security applications (harbour protection) The current generation of vehicles, Remotely Operated Vehicles (ROVs) are safely and routinely used in the off-shore industry. They are teleoperated and require human pilots. There is currently a shortage of trained pilots and deep-water and under-ice operations are new areas of operation where umbilical cables are inefficient. Autonomous Underwater Vehicles are increasingly able to perform more complex missions including survey and inspection tasks in complex environments. However, a main barrier to their more widespread use is the localisation problem. Localisation techniques are required for many underwater applications involving autonomous and semi-autonomous robots. They are used, for example, to perform docking tasks, in order to determine the relative state of the vehicle with respect to the docking station or to navigate around underwater structures, for inspection or intervention. For both survey-class and intervention-class AUVs, a localisation system is a prerequisite for almost every task. Many approaches have been proposed to address the AUV localisation problem. Most of the approaches are *passive*. There is no robot motion control involved. They are based exclusively on the analysis of the sensor data (usually motion estimation and measures of the environment) gathered during a set of predefined operations. Another possible approach is the so-called *active* localisation. This approach incorporates the control of the robot motion, to explicitly perform the sensing required to improve the localisation accuracy of the robot. The authors want to clarify that by *active* localisation, they refer to the vehicle being *actively* sensing, i.e. capable of taking decisions and controlling the vehicle in order to localize. It is not related to localisation with *active* features (e.g. active beacons, lights), as seen in [1].

Related Work

Most of the proposed approaches to localisation are *passive*.The estimation problem has been considered in most cases to be unrelated to the robot motion control problem, which assumes usually a knowledge of the robot position. However, there is some work in the field of *active* localisation in land robotics.

[2] proposed an active choice of the landmark to be observed. However, this approach does not solve the global localisation problem, but helps tracking the position of the robot. If the uncertainty grows, a specific module is called for global localisation, but there is no specific strategy to discriminate between different poses.

This idea is similar to the one proposed in [3], where the landmarks are actively selected, by a supervisor module. This approach is unsuitable for our purposes. First it is landmark-based and it is very difficult to identify reliable landmarks in sonar imagery. Second, it does not address the initial pose estimation and the disambiguation between multiple hypotesis.

[4] proposed an approach based on multiple hypothesis Kalman filter based pose tracking. Using a topological map, the decision on the robot motion is determined by the maximization of the expected number of new features observed in the next possible moves. This approach works well when it is possible to identify a clear set of features. Again this can be difficult underwater and we want to address the

problem in a general way, regardless of the specific structure of the environment. In [4], the exploration is driven by some heuristics preventing the visit of the same location twice. We have also made this assumption, as visiting the same location is not providing any new information.

A cornerstone work in the field of mobile robot active localisation is represented by [5] and [6]. In their work they select actions by maximizing the weighted sum of the expected decrease in uncertainty (entropy) and the costs of moving to the target point. Target points are specified relative to the current robot position and can represent an arbitrary point in the space. Path planning is not involved in the active localisation module. The output is the next position of the robot. Position probability grids are used to estimate the vehicle position.

[7] proposes active sensing, clustering the particles of a particle filter into groups and calculating the total expected entropy for the particle filter using a weighted average of the expected entropy for each group.

With respect to the underwater world, there is very little work in the field of active localisation. The approach of [8] uses active beacons deployed in the environment, in order to help the localisation process. The active strategy is however pretty limited, related only to the disambiguation between the two standard solutions in the beacon localisation problem.

The work carried on by [9] represents an important contribution. It uses active localisation on top of the map previously constructed by a SLAM approach. The set of possible actions are represented by the heading of the vehicle for the following 30 m. The action is selected in order to choose the most discriminative part of the map. The vehicle state is represented using a particle filter, and only a subset of particles are used to evaluate the best action. This is the closest technique to our approach.

[10] propose an approach based on the active movement of an electric field emitter. Apart from the biology-inspired idea of using electric fields, which is out of our scope, the vehicle state is estimate with a particle filter and the control option chosen minimizes the expected variance of the particles at the next step.

Contribution

This paper aims to present a novel approach to the active localisation problem underwater using a priori maps of the environment or maps previously built using SLAM or video mosaicing. With respect to passive approaches, the proposed technique actively decided on the next vehicle move in real time, allowing the vehicle to decide to follow a specific path in order to determine uniquely its position in the state space. This is very important as a random path is generally not able to determine the vehicle position, if unique distinctive features are sparsely present in the environment. In comparison to other active localisation approaches, the proposed method is considering a full trajectory, composed of a set of basic moves. Considering only one basic move, like many approaches described in the previous section, (e.g. [10]) is often not enough. For example, if disambiguating the robots position requires the robot to move to a remote location, greedy single-step algorithms can fail to make the robot move there. In this respect, we fully agree with the work of [6]. However, our approach overcomes some of its limitations. We incorporate the building of a trajectory in our approach, as the *pdf* can change significantly according to the single moves the vehicle chooses in order to reach a predefined location. The reward of an action is expressed by the expectation of the entropy whose calculation is time consuming. We agree with [9] about the computational complexity of the entropy expectation calculation and propose a very computationally efficient solution, based on the diversity of the measurements.

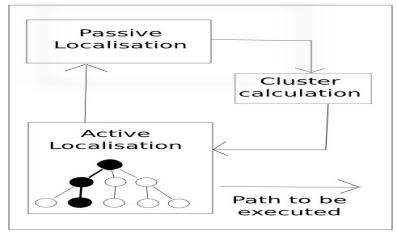


Fig. 1. The general architecture of the navigation system: the passive localisation module is running. According to the probability distribution of the state, the particles are clustered and the centroids are in input to the active localisation module. Through an exploration tree structure, it outputs the path to be executed by the vehile and gives the control back to the passive localisation.

II. AUV Localisation Module

The proposed approach consists of a two-module localisation architecture: the first one is passive, while the second one is active, i.e. with real time control of the vehicle's motion for localisation purposes. The first module is running by default: it receives all the information from the vehicle's sensors (doppler velocity log, compass, depth estimation, sonar measures) and tries to combine them in order to estimate correctly the vehicle state. When there is the need to actively localise, the second module starts and outputs the best estimate of the path to be followed, in order to reduce the uncertainty in localisation. The decision to switch from passive localisation to active localisation is taken by the mission planner, based on the current goals and vehicle state estimation. The navigation architecture is highlighted in Fig. 1.

A. Passive Localisation

This module is based on an improved Particle Filter algorithm. It is assumed that the vehicle has a general (although not perfect) knowledge of the environment. Any particle filter rely on a Monte Carlo approximation of the conditional density $p(x_t|z_{0:t})$ using a finite set of points ξ_t^i in the state space called *particles*. The approximation is of the form

$$p(x_t|z_{0:t}) \simeq \sum_{i=1}^{N} w_t^i \delta_{\xi_t^i}(x_t) \quad \text{where} \quad \sum_{i=1}^{N} w_t^i = 1 \qquad (1)$$

where N represents the number of particles, w_t^i represents the weight of the particle i at time t and δ is the Dirac

function. It can be interpreted in the following way: the denser the particles in a region of the state space and the higher their weights, the higher the probability that the state lies in this region. The most commonly used particle filter is the *sampling with importance resampling* (SIR) algorithm, whose performances are better ([11]) than the first *bootstrap filter*, introduced by [12]. The three steps of iteration $t \geq 1$ of the SIR algorithm are as follows.

1) **Selection:** Generate $\tau_t^i \sim (w_{t-1}^1, \ldots, w_{t-1}^N)$
2) **Propagation:** Generate $\xi_t^i \sim p(x_t|\xi_{t-1}^{\tau_t^i})$
3) **Correction:** Set $w_t^i \propto p(z_t|\xi_t^i)$

More details about the authors previous work on passive localisation using particle filter can be found in [13] and [14].

As part of the filtering process, the particles converge to the most likely positions in the map. According to the map and the motion error, it is possible that more than one position is likely according to the observations and the motion estimation. This is the case, for example, when similar features are present in different parts of the map. In industrial off-shore oil applications, the map is often composed by a sequence of same elements (pipes, rises, underwater structures), while the navigation error from one site to another one is usually bigger than the distance between two same (or similar) elements, thus preventing a straightforward localisation. This can also be the case in unstructured environments. In this case, the mission planner triggers the second module in order to decide where to go for uncertainty minimization in the vehicle's state estimation. It is important to underline that the number of particle clusters is not predefined, but determined dynamically.

B. Active Localisation

We assume here that the vehicle as a finite number of possible positions before the active localisation module is started. This means that the vehicle is already close to known structures and has observed the environment for at least a short period of time. In cases where the vehicle is far from any known structure, the active localisation module could be used to move the vehicle towards recognisable structures for localisation. This has not been considered here. Therefore, the prerequisite to start the active localisation module is to have a finite number of possible current states. The output is a specific path to follow in order to discriminate between these possible states. The input parameters of the this module are given by a clusterization of the particles. The active localisation module is triggered when there is a clear grouping in the particles and when the vehicle needs to find out precisely its state to carry on its mission. The general principle of the module is to find a set of actions that minimizes the expectation of the entropy in the particle distribution. Basic actions a_i that the vehicle can perform are identified:

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

The actions a^i are in the following 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 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 discriminate between possible positions, with only one basic step. It is more often true that only a more complex trajectory can do that, in order to break the symmetries of the environment. 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 initialized 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. Loop closing is a key feature of other navigation algoritms such as SLAM and global bundle adjustment where the navigation error can be bounded by re-observing the same structures. This is not imcompatible with the active localisation process here. In the active localiusation framework, we assume known structures and the paths proposed for localisation are assumed relatively short (navigation accuracy during the execution of these trajectories is good). What we want to avoid is to revisit a area that was already recently visited and for which the information gain is limited. In the SLAM approach, loops are traditionally closed after long periods and navigation accuracy is the issue. Revisiting the same area in that case is critical. It would perfectly legitimate to use active localisation in the SLAM framework to perform accurate loop closing (i.e. the active localisation algorithm is used to confirm we are closing the correct loop).

It is also 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. Modeling the robot behavior as a set of basic actions is important in order to consider that different paths to the same location can produce a different probability density function of the vehicle state. The reward function is thus a critical one, as it needs to be evaluated for each node of the tree. Minimizing the expected entropy means minimizing the function:

$$\begin{aligned} E_{a_i}[H] = -\int\int p(z/x) Bel_{a_i}(x) * \\ log[p(z/x) Bel_{a_i}(x) p(z)^{-1}] dx dz \end{aligned} \tag{3}$$

for every action a^i, as explained in [6]. This calculation is however time consuming ([9]). Our 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 measurements z_t^k from each cluster position k, at time t, after executing all the previous actions linking the root to the node. The sensor measures are simulated from each cluster and from each cluster a *path tree* is generated in order to simulate the sensors over all the possible list of basic actions a^i. It is important to notice that this has the effect of minimizing the expected entropy, without the need of calculating it explicitely.

1) Measurement model: Underwater, the most common sensing modality is sonar because of its good propagation properties in water, enabling long ranges. Here, the sensor used is a profiling sonar which gathers range profiles of the environment as a laser range finder would do in air. The measurements z^k therefore represent an array of distances for each direction that the sonar insonifies. The reward function for a single node n executing the action a^i becomes:

$$r_{a^i}(n) = \frac{\sum_{j=1}^{m} \sigma^2_{z_{a^i}^{j,k}(n)}}{m} \tag{4}$$

where m represents the length of the array and $z_{a^i}^{j,k}(n)$ represents the measures taken from the position at node n, given by the centroid of the cluster k rotated and translated 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 clusters k for each index j of the measures. Each action has a cost associated, which may vary according to the specific constraints. For example, it is reasonable to assume that if action a_t is identical to 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:

$$p^* = \{a^*_{t_0}, ..., a^*_{t_s}\} = argmax_{p_i}(r_{p_i} - c_{p_i}) \tag{5}$$

where p_i represent the path i^{th} and p^* represents the best path. The maximum depth of the tree is fixed *a-priori*. The module can stop either when the maximum depth of the tree is reached, returning the best node (and consequently the path root-node, to be executed by the vehicle) or if a node value (given by the diversity of the measures from the different clusters minus the cost of arriving to that node) is higher than a predefined threshold. This is because the need of a full exploration of the tree might not arise, if enough distinctive features have been recognized quickly.

Summarizing, the general principle of the module is to find a path that maximizes the diversity in the observations from the different initial possible positions. From the centre of each cluster, a trajectory tree is built. 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 maximizes the diversity in the observations and thus minimizing the expected entropy.

III. Experimental Results

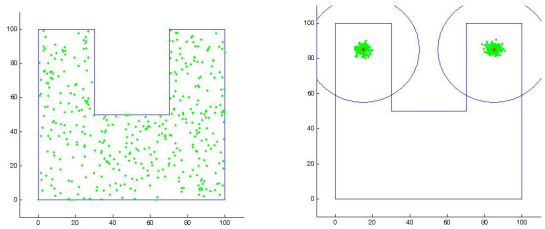
A. Simulated setup

The algorithm has first been tested in a simulated environment. For simplicity, only 2D environments have been tested, in order to reduce the set of basic moves. The 3D extension is 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. The sensor modeled here is a Tritech Micron, currently mounted on our vehicle *Nessie IV*. It is a mechanically scanning imaging sonar (MSIS), with a 360 deg field of view. For this reason, we have not set any rotation in our set of basic moves, as it does not provide any more information about the environment. In the case in which the field of view is limited, then the rotation basic actions are necessary. The set of basic actions is thus represented by:

$$\mathbb{A} = \text{move}\{\text{forward, backwards, left, right}\} \quad \text{for 6 meters} \tag{6}$$

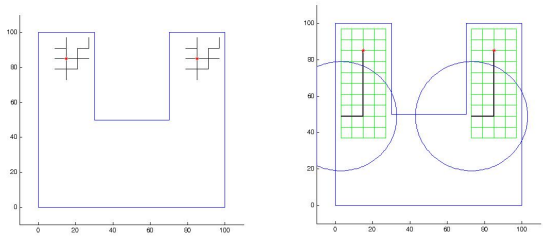
The first environment has a *U configuration*. The vehicle is either at the end of one of the two legs of the *U*. With the vehicle positioned on the left leg, the particles, initially spread all over the environment, quickly converge into the two possible locations. Of course, the same result appears with the vehicle positioned on the right leg. As the observation model from the two points is identical, it is not possible to distinguish between the two hypotesis with classical passive techniques. The control is then given to the active localisation module which takes the two locations of the cluster centroids as inputs. The dimensions of the environment are 100x100 meters. Each leg is 50 meters long and 30 meters wide. The two cluster centroids are located at $(15;85)$ and at $(85;85)$ respectively. With a sonar range of 40 meters, the output of the module is a path composed by six basic moves, all going backwards. This is actually the best path in order to discriminate between the two solutions, as the diversity in the environment can be sensed on the bottom of the *U*. Reducing the range to 30 meters, the output is composed by eight basic moves: six backwards and then two on the left. This is consistent with the expectations, as the generated path arrived to a point very close to the borders of the environemnt (presence of obstacles) for one cluster, while for the other cluster, the final point was far from any obstacle. This simulated environment shows the possibility to apply active localisation in closed environment, like it is often the case for man-made environments, like marinas. All the steps are highlighted in Fig. 2.

The second environment has been chosen to be closer to open sea conditions and off-shore applications. There are no boundaries, just three objects, which can represent an underwater site. Assuming that the vehicle is traveling from one operational site to another, it is very likely that the navigation error will grow bigger than the distance between two objects in the scene. The vehicle will need to discriminate between the possible locations(in this case, three). This case is also interesting as it shows how a small change in the



(a) Initial state: uniform distribution of the vehicle state

(b) Clusterization of the particle



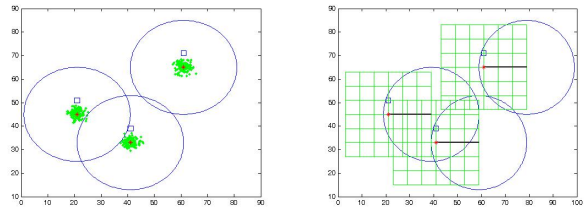
(c) Tree built from each centroid

(d) Path chosen to maximize the information gain

Fig. 2. All the steps of the active localisation process: clusterization, tree construction and path building. First scenario: U-like closed environment

parameters of the sonar can change significantly the results. Due to the location of the underwater objects, a sonar range of 27 meters can discriminate between the positions without the need for any active localisation. Reducing the range gradually, the generated path changes significantly. Between 26 and 27 meters, one move is enough to distinguish between the three hypotesis and the selected move is to go backwards. For the top right centroid, this has the effect to go nearer the two middle objects. When the range drops to 23 meters, the selected move is to go right. For the left centroid, this has the effect to go nearer the central object. Reducing again the range, the required trajectory is composed by two steps and again the first choice is to go backwards, creating a measure discrepancy between the top centroid and the other two. At 20 meteres range, the two steps are on the right. Up to 17 meters range, the planned path is to go on the right for three steps: this helps to discriminate the left centroid (very near to the central object) with respect to the other two objects. It is now interesting to see what happens for sonar range below 17 meters. In this case, the Active Localisation module needs to fully explore the tree of possibility until the maximum tree depth (fixed at nine) is reached. However, the best path that the algorithm can find is not a path of length 9, but a path of length 3 corresponding to the last path generated for ranges between 17 and 20 meters. A representation of the environment, with particle clustering and chosen path is highlighted in Fig. 3.

A complex simulated test - a labyrinth-style environment seen in Fig. 4 - represents our third simulated environment. It proves the validity of the algorithm whist testing a possible real scenario such as an underwater cave inspection. For such



(a) Particle clustered in front of the three objects

(b) Path chosen to maximize the information gain

Fig. 3. Second scenario: three objects in an open environment.

complex scenario with so many constraints, given by the walls, the algorithm needs to be applied iteratively, in order to get to a unique position of the robot pose. The first path generated by the algorithm drops the number of clusters from the inital six to three. The second path drops it from three to two, while the third path provides a unique solution. It is to be noted that the last path is actually a degenerated path, with the robot deciding not to move.

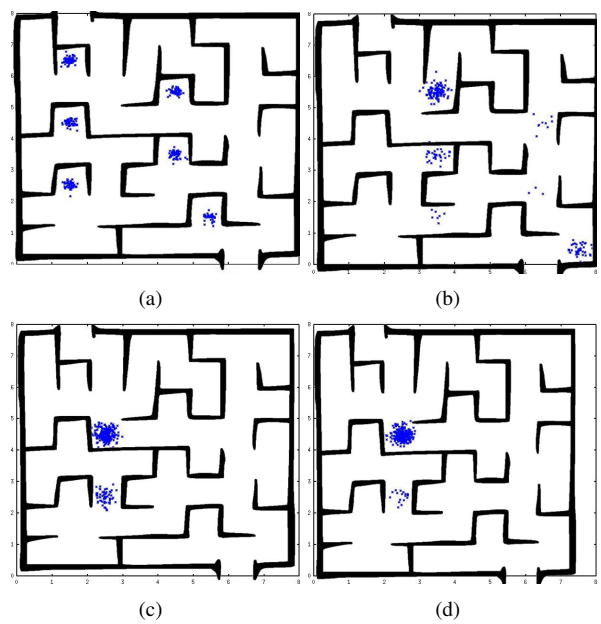


Fig. 4. (a) inital distribution (b) pose estimation after first path (down-right-right) is execured. Three clusters are dropped, (c) pose estimation after second path is executed (left-down). An additional cluster is dropped. (d) execution of the third path (stay still): particle convergence.

B. Tank Trials

The algorithm has been succesfully tested in several tank trials, using the facilities at Heriot-Watt University. The test platform was the Autonomous Underwater Vehicle *Nessie IV* [15], equipped with a Tritech Micron sonar to sense the environment, and Doppler Velocity Log (DVL) for motion estimation. A first set of tests has taken place in a 3x4 m tank. A panel has been put into the tank in order to create two identical parts in the environment, similar to the *U-scenario*

described in the simulated setup. The results are highlighted in Fig. 5.

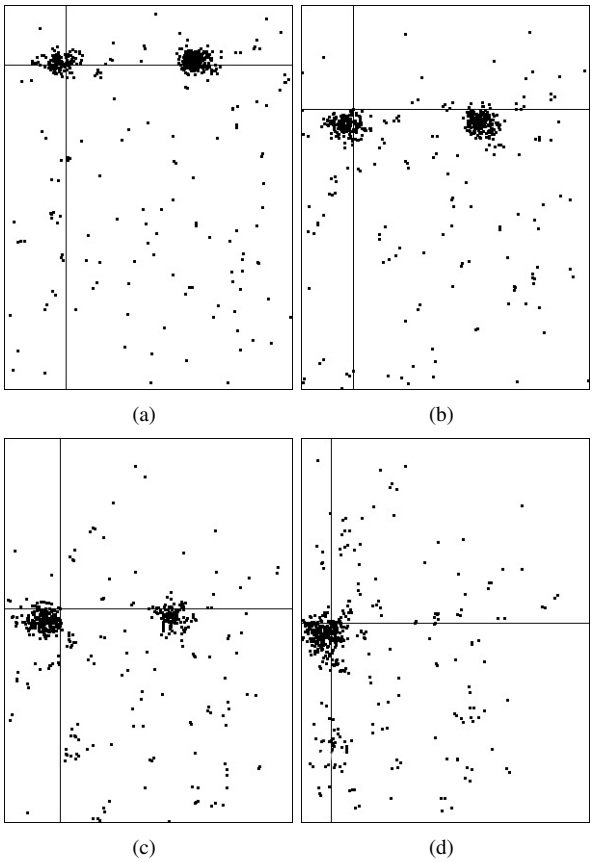


Fig. 5. (a) Initial distribution (b-c) executing the path (down-down-down-down-left-left) (d) Convergence after execution of the path generated by the Actve Localisation module.

The second set of tests has taken place in a 10x12 m tank. Several environments have been tested. In this paper we only present one set-up, highlighted in the attached video. The environment is composed by the tank walls plus four panels making two identical sections. The robot starts with no initial knowledge of its position, so particles are spread over all the environment. After convergence, a clear and stable clustering of the particles is reached. The active localisation module is triggered and the vehicle computes the path to be executed. After executing the path, the vehicle's position is determined without doubts.

IV. Conclusion Future work

This paper has presented a novel approach to the active localisation problem. Driven by the maximization of the information gain, the algorithm has proven to be reliable and efficient. The key achievement is the ability to efficiently and autonomously build a path to discriminate between possible poses and eliminate all but one possibe solutions. The experimental results have shown that the propossed algorithms performs well in different scenarios, always providing excellent results. The next step will be to test the algorithm in a realistic off-shore scenario, using the *ARF* simulator ([16]). Real tests will then be performed during a long mission in a Scottish Loch. In the context of the EC funded Trident project, accurate video mosaics of the seabed will be produced. They will provide a feature reach prior map of the environment that can be used in exactly the same way to relocalise the vehicle. It can also be used to find again a specific location for target reacquisition. This will be tested in near future.

Acknowledgment

This research was sponsored by the Trident project EC-248497, the Marie Curie Research Training Network FREEsubNET (MRTN−CT−2006−036186) and by the Scottish Funding Council for the Joint Research Institute in Signal and Image Processing between the University of Edinburgh and Heriot-Watt University (part of the Edinburgh Research Partnership in Engineering and Mathematics - ERPem).

References

[1] F. Giuffrida, P. Morasso, G. Vercelli, and R. Zaccaria. Integration of active localization systems in vehicles control for real-time trajectory tracking. pages 549–556, Dec 1996.

[2] A. Arsenio and M.I. Ribeiro. Active range sensing for mobile robot localization. 2:1066–1071 vol.2, Oct 1998.

[3] C. Tessier, C. Debain, R. Chapuis, and F. Chausse. Active perception strategy for vehicle localisation and guidance. pages 1–6, June 2006.

[4] P. Jensfelt and S. Kristensen. Active global localization for a mobile robot using multiple hypothesis tracking. *Robotics and Automation, IEEE Transactions on*, 17(5):748–760, Oct 2001.

[5] Wolfram Burgard, Dieter Fox, and Sebastian Thrun. Active mobile robot localization. In *In Proceedings of IJCAI-97. IJCAI, Inc*. Morgan Kaufmann, 1997.

[6] D. Fox, W. Burgard, and S. Thrun. Active markov localization for mobile robots. volume 25, pages 195–207, 1998.

[7] Rainer Kümmerle, Patrick Pfaff, Rudolph Triebel, and Wolfram Burgard. Active monte carlo localization in outdoor terrains using multi-level surface maps. In *AMS*, pages 29–35, 2007.

[8] E. Olson, J. Leonard, and S. Teller. Robust range-only beacon localization. In *Autonomous Underwater Vehicles, 2004 IEEE/OES*, pages 66–75, June 2004.

[9] Nathaniel Fairfield and David Wettergreen. Active localization on the ocean floor with multibeam sonar. In *Proceedings of MTS/IEEE OCEANS*, 2008.

[10] James R. Solberg, Kevin M. Lynch, and Malcolm A. MacIver. Robotic electrolocation: Active underwater target localization with electric fields. In *ICRA*, pages 4879–4886, 2007.

[11] W. Burgard, D. Fox, and S. Thrun. On sequential monte carlo sampling methods for bayesian filtering. *Journal of Artificial Intelligence*, 2001.

[12] N. Gordon, D. Salmond, and C. Ewing. A novell approach to nonlinear nongaussian bayesian estimation. In *Proc. IEEE Radar and Signal Processing*, 1993.

[13] F. Maurelli, S. Krupiński, A. Mallios, Y. Petillot, and P. Ridao. Sonar-based auv localization using an improved particle filter algorithm. In *Proceedings of IEEE Oceans '09, Bremen, Germany*, 2009.

[14] F. Maurelli, A. Mallios, D. Ribas, P. Ridao, and Y. Petillot. Particle filter based auv localization using imaging sonar. In *Proceedings of IFAC MCMC '09, Sao Paulo, Brazil*, 2009.

[15] F. Maurelli, J. Cartwright, N. Johnson, G. Bossant, P.L. Garmier, P. Regis, J. Sawas, and Y. Petillot. Nessie iv autonomous underwater vehicle. In *Proceedings of UUVS 2009, Southampton, UK*, 2009.

[16] Benjamin C. Davis, Pedro Patron, Miguel Arredondo, and David M. Lane. Augmented reality and data fusion techniques for enhanced Situational awareness of the underwater domain. In *OCEANS 2007 - EUROPE, VOLS 1-3*, pages 1090–1095, 2007. Oceans 2007 Europe International Conference, Aberdeen, SCOTLAND, JUN 18-21, 2007.