REAL-TIME LOCALISATION OF A LOW-COST MOBILE ROBOT WITH POOR ULTRASONIC DATA.

P. Hoppenot, Doctor ; E. Colle, Professor.

CEMIF, Systèmes Complexes, 40 rue du Pelvoux, 91020 Evry Cedex (France)
Tel : Colle [(+33-1) 69-47-75-57], Hoppenot [(+33-1) 69-47-75-04]
email : hoppenot, ecolle,...@cemif.univ-evry.fr

Abstract : In order to open the field of autonomous mobile robotics to new applications such as the provision of assistance to disabled people, the research is being focused upon low-cost solutions. That implies the use of poor perception systems and low computing power. In such a context, the algorithms used have to be simple, if they are to be executed in real time, and proof against the weaknesses of the sensing systems. The localisation approach presented here is based on the fact that the higher the localisation algorithm speed is, the lower the error in the position and the orientation, due to the odometry. Any systematic errors in the relative localisation using odometry are corrected on-line by using a limited set of ultrasonic data. If a non-systematic error occurs, a more complex procedure is necessary.

Both simulation and experimentation show that the systematic odometric errors become bounded, thanks to those algorithms. Moreover, they are robust to a high rate of false ultrasonic measures.

Keywords : Mobile robots, medical robotics, ultrasonic transducers, range data, least-squares algorithm, localisation estimation, localisation errors.

1. Introduction

Mobile robot displacements require the provision of correct position and the orientation knowledge by the localisation function. The classical approach uses proprioceptive devices, e.g. an odometer, for the relative localisation, and a more complex exteroceptive system to periodically correct the relative localisation. Many authors have studied the localisation problem in known and structured environments, and with an advanced but expensive perception systems, e.g. a laser range-finder or camera(s). Therefore new mobile robot applications will require substantial reductions in the costs involved.

The application descibed in this paper deals with medical robotic aids, and is aimed at the provision of a conveyance and manipulation assistance, for severely disabled people (Hoppenot et al, 1996). The system is composed of a mobile robot which plays the part of the carrier manipulator arm. The mobile robot is built around a wheelchair platform that integrates the computer and perception modules. The perception system is reduced to an odometer and a ring of eight ultrasonic range-finders. The computer system is a multiprocessor architecture.

Odometry solves the relative localisation at low cost but presents different error sources, that fit into two categories (Borenstein, 1996):

- Systematic errors, principally due to the unequal wheel diameters, the misalignment of wheels, and the uncertainty about the contact between the wheels and the floor.

- Non-systematic errors, due to rough floors and wheel-slippage.

The localisation strategy distinguishes betwenn two kinds of errors. The systematic errors are corrected on-line by a high-speed algorithm, using a small set of exteroceptive measures. A non-systematic error requires a more complex and more time-consuming procedure to localise the robot, using a large set of exteroceptive measures.

Among the range-finders that are currently available, ultrasonic sensors respect the low cost constraint but present several significant sensing problems. Specular reflection implies that a surface that is non orthogonal to the direction of acoustic propagation will not be detectable. Multiple reflections produce erroneous measures. Such poor perception systems require algorithms that are proof against many erroneous measures.

This paper presents robust solutions for the correction of systematic odometric errors on-line. Algorithms are applied to a simulated then a real robot.

In Section 2, the general localisation problem is presented. In Section 3, two algorithms to correct systematic odometric errors are described. Section 4 gives the experimental conditions. In Section 5, the simulation results are demonstrated and in Section 6 experimental results are given. Section 7 discusses these results, and Section 8 concludes this work.

2. The localisation problem

Much work has been carried out on this subject. This problem is very close to the map-building one, and most publications deal with both mapping and localisation.

Different techniques have been presented in the literature. In (Crowley, 1989 ; Kröse et al., 1993) a measured point is associated with an already built segment if the distance between them is less than a predefined threshold. The environment is built up during the movement of the robot, and a new point is used to locate the position of the segment precisely. In (Cox, 1991), the environment is known a priori. A measure is matched with the closest known segment. This approach does not need a sensor model. The computation is not very complex, and is undertaken in real time. Another solution consists of using grids to represent the environment (Elfes, 1986; Elfes, 1990; Matthies and Elfes, 1988). Each measure naturally matches one cell of the grid. In (Mandelbaum and Mintz, 1993), the environment is modelled by segments and a grid. The grid is used for matching, and each cell contains information (e.g. « occupied » or « empty ») for use in obstacle avoidance. A pointer to a list of features is associated with each cell. Higher-level tasks such as localisation need more precise information. A measure belonging to a cell is directly matched with the feature pointed to by that cell. The drawback of this method is the large amount of basic information that is introduced into the model of the environment. In (Schiele and Crowley, 1994), another use of both segments and grids is based on several local maps and a global maps. Each map can be represented by a grid or a set of segments. The main issue is the matching of one local map againts the global one. Four cases are encountered: segment-segment, segment-grid, grid-segment and grid-grid.

According to the papers cited above, the matching problem is one of the more important issues in the localisation of mobile robots. In this paper, the approach is based on Cox's (1991) work. Each measurement is matched with the nearest segment of the environment. The main interest of this choice is that the matching is made independently for each measurement. The problem is less complex than in the other approaches, and easier to solve.

In order to respect the low-cost constraint, the perception system is composed of a ring of eight ultrasonic sensors. In contrary to the quite accurate laser measurements used by Cox (1991), ultrasonic sensors provide many erroneous measurements due to specularity, multiple echoes and large solid angles (Wilkes et al., 1993). This requires an adaptation of Cox's (1991) idea. The solution that matches the measurements with the known environement is developed in detail in Section 3.1.

This matching is made possible only by the assumption that the odometric error is not too important. With reference to (Borenstein, 1996), there exist two types of odometrical errors. Non-systematic ones are not predictable. Systematic errors increase over time. So, the higher the speed of the localisation algorithm is, the higher the acceptable systematic error of the odometry. The solution to the matching problem permits localisation algorithm to be run in real time (7.3), which is important in any on-line localisation problem.

3. The Localisation algorithms

Two algorithms are presented here. The first takes account of only one burst of measurements, that is to say 7 measurements, one from each sensor (see Figure 4). In the second one, a memory effect covering the 10 last bursts is used, so 70 measurements are available.

However, before developing these algorithms, it is important to explain in detail how to match the measurements with the known environment.

3.1 The matching solution

This is the main contribution of this paper. The objective is to simplify the localisation algorithms by matchnig the dead-reckoning localisation against the points of impact of the measurements. According to the odometry, the position of the robot is known. For each measurement, an impact point is calculated, using this odometric position of the robot. The idea is then to match the point of impact with the nearest segment of the known environment.

Cox (1991) works with laser measurements. These are almost perfect, in comparison with ultrasonic measures. In this paper, not only is the nearest segment of the environment used, but a threshold is also defined to reject wrong measures due to specularity, multiple echoes and large solid angles.

In fact, two thresholds are used, depending on the relative positions of the ultrasonic point of impact and the segment of the environment. Three areas are defined: A_0, A_1 and A_2 (Figure 1). If the point of impact of the measure is in A_0, a threshold T_0 is used. In A_1 and A_2, another threshold T_{12} is defined, smaller than T_0.

The distinction can be explained as follows. In A_0, the distance is computed between the impact of the measurement and the segment of the modelled environment, in a direction perpendicular to the segment (see eq. 3 in the following paragraph). This is consistent with ultrasonic measurements, which always give the shortest distance to a wall. In A_1 or A_2, the distance is computed between the impact and the closest end of the segment (eq. 4) in the following paragraph. That choice is less consistent with ultrasonic measurements; but allows a smoothing of the matching. The thresholding takes account of the area distinction and filters more points of impacts in A1 or A2 than in A0. A compromise is found with T_0=0.25m and T_{12}=0.05m.

3.2 The pin-point algorithm

3.2.1 Presentation of the algorithm

Position (x, y, θ) is given by the odometry. Distances are measured by the 7 forward sensors in the ring (Figure 4). When the robot moves in a known environment the best robot position minimises the sum of the square measured distances to the walls. The function to be minimised is complex if all the possible robot positions are considered. Each measurement can be matched with known walls in the environment. The computing complexity is higher than this algorithm.

Indeed, use of the matching described above simplifies the localisation algorithms. The odometric position error must stay less than predefined threshold (25 cm). That leads to the following algorithm, based on a least-squares one:

```
1- Measure the seven distances
2- Find the impact
3- Match impact with the environment
4- Compute the gradient of $F$
5- Compute the global position correction
6- Estimate the cost function $F$
    6-1-**if** $F < \varepsilon$, **or** $\partial F < \varepsilon'$ **or** $n > N$ **goto** 7
    6-2-**else goto** 2.
7-**if** $n > N$,
    - return the original position
    - **if** $\partial F/\partial x < \varepsilon'$ use $x$ correction
    - **if** $\partial F/\partial y < \varepsilon'$ use $y$ correction
    - **if** $\partial F/\partial \theta < \varepsilon'$ use $\theta$ correction
8- **else** use the global correction
```

where F : cost function (eq. 5),
∂F : $\partial F/\partial x$ or $\partial F/\partial y$ or $\partial F/\partial \theta$,
n : iteration number,
N : maximum number of iteration.

If the exit condition is due to too large a number of iterations (step 7), the correction is done in only the directions (x, y, θ) in which the gradient is less than a predefined threshold ε'.

3.2.2 The cost function

As a least-squares algorithm is used, the cost function must be continuously derivable.

Let $P(x, y, \theta)$ be any position of the robot. Let $M_i(x_i, y_i, \theta_i)$ be any impact of ultrasonic sensor number *i*, where θ_i is the direction of measurement in the robot reference (Figure 4). So the co-ordinates of the impact *i* are :

$$\begin{cases} x_i = x + d\cos(\theta + \theta_i) \\ y_i = y + d\sin(\theta + \theta_i) \end{cases} \quad (1)$$

where d is the measured distance.

Let S, supported by the straight line of equation $ax + by + c = 0$, be the segment matched with M_i. Let $S1(x_1, y_1)$ and $S2(x_2, y_2)$ be the two extremities of S. Let $H(x_h, y_h)$ be the orthogonal projection of M_i on the straight line supporting S, whose co-ordinates are :

$$\begin{cases} x_h = \left(b^2 x_i - aby_i - ac\right) \big/ \left(a^2 + b^2\right) \\ y_h = \left(a^2 y_i - abx_i - bc\right) \big/ \left(a^2 + b^2\right) \end{cases} \quad (2).$$

Then F_i is defined as follows :

- if H belongs to S **then** $F_i = \dfrac{(ax_i + by_i + c)^2}{(a^2 + b^2)}$ (3) which is the squared distance between M and H, corresponding to the area A_0 defined above,

$$\textbf{- else } F_i = \min \begin{cases} (x_i - x_1)^2 + (y_i - y_1)^2 \\ (x_i - x_2)^2 + (y_i - y_2)^2 \end{cases} \quad (4)$$

which is the minimum squared distance between M and S_1 and between M and S_2, corresponding to the areas A_1 and A_2.

The algorithm requires a continuously derivable function.

F_i is continuous everywhere in x_i and y_i. As x_i and y_i are continuous in x, y and θ, F_i is continuous everywhere in x, y and θ.

The derivatives of F_i have to be continuous, specially between the two parts of the function. The derivatives of each form of F_i with regard to x_i when H equals S_1 are calculated. The first form (3) gives $\dfrac{\partial F}{\partial x_i} = \dfrac{2a(ax_i + by_i + c)}{(a^2 + b^2)}$ and the second (4) gives

$\frac{\partial F_i}{\partial x_i} = 2\left(x_i - x_1\right)$. Now, as H equals S_1, (2) gives $\begin{cases} x_1 = \left(b^2 x_i - aby_i - ac\right) \big/ \left(a^2 + b^2\right) \\ y_1 = \left(a^2 y_i - abx_i - bc\right) \big/ \left(a^2 + b^2\right) \end{cases}$.

So $\left(x_i - x_1\right) = \frac{a\left(ax + by + c\right)}{a^2 + b^2}$ and F_i is derivable everywhere with regard to x_i. The derived forms show that $\frac{\partial F_i}{\partial x_i}$ is continuous too. The same result can be demonstrated for $\frac{\partial F_i}{\partial y_i}$.

The final point is to derive F_i with regard to x, y and θ. Now x_i and y_i are continuously derivable with regard to x, y and θ (1). So F_i is continuously derivable with regard to x, y and θ.

Now, $F = \sum_i F_i$, (5).

So F is continuously derivable with regard to x, y and θ.

3.3 Pin-point algorithm with memory effect

In this algorithm the last ten measurements of the seven sensors of the ring are memorised in order to build segment features. Then the segments are matched with the modelled environment.

How is one to build the segments? The first step is to choose which segment the measure number *i* belongs to. For example in (Crowley, 1989) and (Kröse *et al.*, 1993) a new segment is built when three points are aligned with a certain predefined tolerance. A new point belongs to a segment if the distance to it is smaller than a predefined threshold. In (Mc Kerrow, 1993), an ultrasonic measure is stored as a circle arc. Two different measurements come from the same plane if there is a common tangent to the two arcs. In both cases the computation is quite complex and time-consuming.

As in the previous method, a sensor measurement is matched with a segment of the environment, taking the odometric position into account. When a set of points has been associated with the same segment of the environment, a segment is computed by using a linear regression between these points (Figure 2-a). So two segments are available: the known one in the model of the environment, and the computed one. It is very important to point out that each measured segment has an associated segment in the known environment: the matching problem is solved. In this method segments are represented according to Crowley's formalism (Crowley, 1989).

The correction is performed in two steps. First, only the orientation is corrected (Figure 2-b). This is the sum of the differences between the calculated segment orientations and the known ones, divided by the number of segments used. Then the (x,y) position is corrected (Figure 2-c) by minimising the sum of the distances between the middle of the measured segment, and the known segment of the environment. The same least-squares algorithm is used a above.

4. Experimental conditions

4.1 Environment and robot characteristics

The known environment (Figure 3) is composed of a room with a door aperture and a smooth ground surface. The task to be performed is a movement from the source to a goal, across sub-goals. The robot, called RMI (French abbreviation for « Intelligent Mobile Robot »), is a circular robot with two drive wheels. The perception system integrates a ring of eight Polaroid ultrasonic sensors (Figure 4) and an odometric device.

A static kinematic model of the robot is used. The direct model is : $\begin{cases} V = R \cdot \left(\omega_l + \omega_r\right) \big/ 2 \\ \Omega = R \cdot \left(\omega_l - \omega_r\right) \big/ \left(2 \cdot E\right) \end{cases}$

where *V* is the linear speed, Ω the angular speed, *R* the radius of the wheels, *E* the distance between the two wheels, ω_l the rotation speed of the left wheel and ω_r the rotation speed of the right wheel. The inverse model is given by : $\begin{cases} \omega_{l=(V+E\Omega)/R} \\ \omega_{r=(V-E\Omega)/R} \end{cases}$.

4.2 Experimental protocol

The algorithms are applied first on a simulated robot, and then on a real one.

4.2.1 Case of the simulated robot

To compare the localisation algorithms the experimental protocol performed is:

1- Move the real robot and memorise the odometric and ultrasonic data.
2- Degrade the odometric data.
3- Simulate robot movements using real ultrasonic and modified odometric data.
4- Execute the localisation algorithm at each position.
5- Draw the paths followed by the real robot, the modified odometric robot and the modified odometric robot after the position corrections.

4.2.2 Case of the real robot

The real robot is then used to test the two algorithms. As it is difficult to make an over-inflated tyre, the odometry is degraded when the position is calculated. The error is then well known; so a comparison between the simulation and experiment can be established.

In these experimental conditions, the assumption that the odometer (when not degraded) provides correct information about the robot position is verified.

4.3 Degradation of the dometric data

With reference to (Borenstein, 1996), systematic errors are corrected. A constant bias is applied to the wheel diameter. The deviation taht is taken into account at each iteration ends in complete confusion of the robot. Two cases have been treated:

- asymmetrical inflation e.g. the left tyre is over-inflated and the other sub-inflated
- symmetrical but incorrect inflation

The first case is the more difficult to solve. All the results are presented for that case.

5. Simulation results

All the simulation results are presented for the same set of odometric and ultrasonic data. Each algorithm is first presented without the disturbing odometry, in order to show that the correction does not impair correct odometric localisation by too much. Then a degradation is introduced to display the method’s limits. Finally, a table of maximum errors for the x and y axes allows a comparison between the different methods to be made.

5.1 Without odometric disturbance in a completely known environment

Ultrasonic sensors have a wide aperture angle (30 degrees for Polaroid transducers) and a low distance accuracy, typically 3 centimetres. Therefor, using them to localise the robot, even when the odometry is correct, will introduce an error. That error must not be too important, and must be compared with the improvement when the odometry is disturbed.

5.1.1 Algorithm without memory

The distance error goes up to 20 centimetres (Figure 5) after the doorway, because of the unknown environment. This error at the end of the trajectory will be noticed in each case (Figure 6).

e is the localisation error, measured in metres. This is the difference between the actual trajectory and the reference trajectory without odometric degradation and without relocalisation. t is the run time of a trajectory. One graduating of the time scale is about 0.5s, corresponding to one complete cycle of the measurement and localisation algorithm.

5.1.2 Algorithm with memory

If the odometry is correct, the localisation degradation due to the defaults of the ultrasonic sensors is less than 0.15 metres in a completely known environment after a 4-metre displacement.

5.2 With a constant odometric disturbance in a completely known environment.

5.2.1 Algorithm without memory

Figure 7 shows the result with a constant error caused by an 8% sub-inflated wheel. DD is the cumulative error without correction, and dD with the correction. The error in the corrected trajectory is not more than T_0 (25 cm) after a 4-metre trip, whereas the error of the non-corrected trajectory goes up to 1.8 metres. The aim is not to locate the robot very precisely, but to avoid becoming lost, so this result is satisfactory (Figure 7). With a constant 9% sub-inflated wheel the corrective action is insufficient. Until the thirtieth step the correction is good. However, if the error is over 20 centimetres, the matching algorithm no longer works, so there is no further localisation and the error increases.

5.2.2 Algorithm with memory

Using the memory method, an 11% disturbance can be corrected in the worst case. The corrected trajectory follows the reference one (trajectory without odometrical degradation) quite well (Figure 8).

5.3 Without odometric disturbance in a partially known environment. Algorithm with memory.

In the next experiment, the environment is composed of the same room and an unknown obstacle. That poses a matching problem between measurements and known modelled segments of the environment. Matching conditions are less strict than previously, to permit a sufficient number of measurements to be used. Results are presented only for the algorithm with memory because of its better robustness to uncorrect ultrasonic measures. Up to 50% (Figure 9) of them are unusable.

5.4 With a constant odometric degradation in a partially known environment.

The same degradation percentage as in a completely known environment cannot be reached. The obstacle occludes some important segments in the environment.

The best result that can be obtained is a correction of a 4.5% degradation in the worst case (Figure 11).

5.5 Discussion

Only the worst case is presented here. It corresponds to an asymmetrical inflation of the tyres, which induces localisation errors in x, y and θ .

The pin-point method with a memory effect presents the possibility of adjusting different parameters. It is possible, for example, to adjust the depth of memory, the association distance, the maximum angle of measurements and the localisation correction rate. Thanks to that adjustment ability, the same algorithm operates in both known and partially known worlds. The dynamic modification of the parameters can be driven by the rate of recognition of the environment.

6. Experimental results

The algorithm with a memory effect is evaluated in real situations: first in the same condition as seen previously (Figure 3) then in a more complex environment.

6.1 With a constant odometric degradation in a totally known environment.

In the worst case, a correction of a 9% degradation is performed (Figure 12). It is a little less than the simulation results, but not by too much.

The line called the real distance is the distance between the real position of the robot and the reference position (trajectory without odometric degradation). Indeed, the computed position is not the real position of the robot, and it is interesting to compare them.

6.2 With a partially known environment.

The same 4.5% degradation (as in the simulation results) can be corrected (Figure 13).

6.3 Case of a more complex environment

It is interesting to see how this algorithm works in a more complex environment (Figure 14). This is a room with three cupboards, a special furniture unit in the top right-hand corner and an unmodelled area composed of tables and chairs.

Several trajectories have been performed. As a summary, the localisation algorithm works well for a 3% odometric degradation. The maximum error at the end is less than 25cm in x and y, and 15° in θ after a 15-metre trip. Above this level, wrong localisations appear e.g. with a 4% odometric degradation, the robot gets lost twice in 11 different trajectories.

6.4 Discussion.

Those results with a real robot show that the simulation conditions were near enough to the reality.

The results are presented only for the second algorithm. That choice was guided by the possible adjustment of that method (5.5), and the robustness to the high rate of wrong ultrasonic measures.

7. Discussion

7.1 Comparison of the algorithms

Table 1 presents the comparative results of the two algorithms, in simulation and in reality. Generally, the authors estimate a real odometrical error at around a few percent on a smooth surface.

	(1)	**(2)**	**(3)**
without obstacles	8%	11%	9%
with obstacles	3%	4.5%	4.5%

Table 1: *Acceptable odometric error.*

In the table, the colums represent:

(1) pin-point algorithm without memory, simulation ;
(2) pin-point algorithm with memory, simulation ;
(3) pin-point algorithm with memory, real results.

The method with memory is better than that without memory, and also presents a larger margin of evolution.

7.2 Room occupied by an obstacle

Figure 16 shows the evolution rate of the robot vision field masked by the obstacle in the environment of Figure 17. It explains the drop in the algorithm’s performance. Indeed, at the beginning, the obstacle is just in front of the robot, so the wall behind it is not seen. At the end, the obstacle is no longer in the robot’s field of vision.

Figure 16 shows change in the field of vision of the robot, masked by the obstacle. This ratio, in fact α_O/α_T, where α_O is defined in Figure 17 and $\alpha_T=\pi$, corresponds to the fact that the sensors are fixed on the front semi circle of the robot. This evolution (Figure 16) is presented for a trajectory without odometic error.

7.3 Real-time considerations

With regard to real time, the timing of the algorithm is at least 10 times less than the sensors data-acquisition timing which is about 0.5s on a 16 MHz Intel 80196 microcontrollor. In fact, most of the time is used by the time of flight of the ultrasonic wave. More precisely, the mean time of the algorithm without memory is 2.5 ms, whereas the mean time of the algorithm with memory is 50 ms on a 133 Mhz pentium PC. In the second case, it is interesting to point out that 10% of the time is used for the matching operation, 70% for the linear regression calculation to build up the calculated segments, and 20% for the localisation computation. So 90% of the time is required by the leastsquares algorithm (which is used in both the linear regression and the localisation computation). The aim of this work was to show that this kind of matching gives interesting results.

8. conclusion

Generally, knowledge of the position and orientation of a mobile robot uses two functions, known as relative and absolute localisation. The former is looked after by the odometry, and is simple and inexpensive. Its disadvantage is an unbounded accumulation of errors. The latter requires a more complex system, based on a laser range-finder and/or camera(s) to correct the odometry from time to time.

With a poor perception system, the strategy must be different, and must take account of the categories of odometric errors.

In the approach described in this paper, a real-time algorithm limits the accumulation of systematic errors by using a limited set of ultrasonic measuremants. The call to absolute localisation is no longer necessary unless if a non-systematic error occurs. In that case a more complex procedure, based on a large set of ultrasonic measurements is required. That procedure is now under consideration.

The algorithm is able to correct a 9% systematic error in a known environment, and a 4.5% error in a partially known one. In the latter case, the obstacle can mask up to 40% of the robot’s field of vision. These values have to be compared to the 2 to 3% error attributed to the odometry in the literature for an indoor environment.

Furthr more, the algorithms present a high insensitivity to erroneous and inaccurate ultrasonic measures. The rate of false measurements due to specularity, multiple echoes and the large solid angle can reache up to 50%.

At this stage the approach does not take account of the inadequate knowledge of the orientation position, or of the position of the robot at the commencement of the task.

The addition of the effect of memory to the pin-point method gives the ability to adjust the parameters of the algorithm dynamically to the type of environment.

9. References

Borenstein J. (1996) : « Measurement and correction of systematic odometry errors in mobile robots » - IEEE Trans on Rob and Auto, vol. 12, N°6, pp869-880.

Cox J. (1991) : « Blanche - an experiment in guidance and navigation of an autonomous robot vehicle" - IEEE Trans on Rob and Aut, vol 7, n°2.

Crowley J.L. (1989) : « World modeling and position estimation for a mobile robot using ultrasonoic ranging » - IEEE Int. Conf. on Robotics and Automation.

Schiele B. and Crowley J.L. (1994) : « A comparison of position estimation techniques using occupancy grids » - IEEE International Conference on Robotics and Automation, San Diego.

Elfes A. (1986) : « A sonar-based mapping and navigation system » - IEEE International conference on robotics and automation.

Elfes A. (1990) : « Occupancy grids : a stochastic spatial representation for active robot perception » - IEEE Proc 6th Conf on Uncertainty in Al.

Matthies L. and Elfes A. (1988) : « Integration of sonar and stereo range data using a grid-based representation » - IEEE Int Conf on Rob and Aut, pp 727-733.

Hoppenot P. , Benreguieg M., Maaref H., Colle E. and Barret C. : « Control of a medical aid mobile robot based on a fuzzy navigation » - IEEE Symposium on Robotics and Cybernetics, july 1996.

Kröse B.J.A., Compagner K.M. and Groen F.C.A (1993) : « Accurate estimation of environment parameters from ultrasonic data » - Robotics and Autonomous systems 11, 221-230.

Mandelbaum R. and M Mintz (1993) : « Active sensor fusion for mobile robot exploration and *navigation » - 130 SPIE vol 2059, sensor fusion VI.*

Mc Kerrow P.J. (1993) : « Echolocation - from range to outline segments » - Robotics and Autonomous systems 11, 205-211, El Sevier.

Wilkes D., G. Dudek and M. Jenkin (1993) : « Multi-transducer sonar interpretation » - IEEE Int. Conf. on Robotics and Automation, pp 392-397.

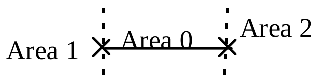

Figure 1. *Three areas defined for a segment.*

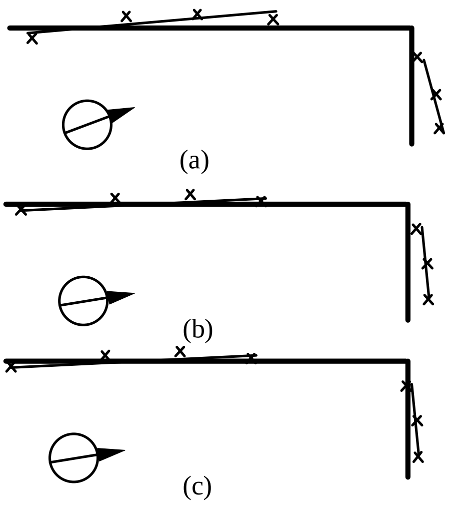

Figure 2 : *Position correction.*

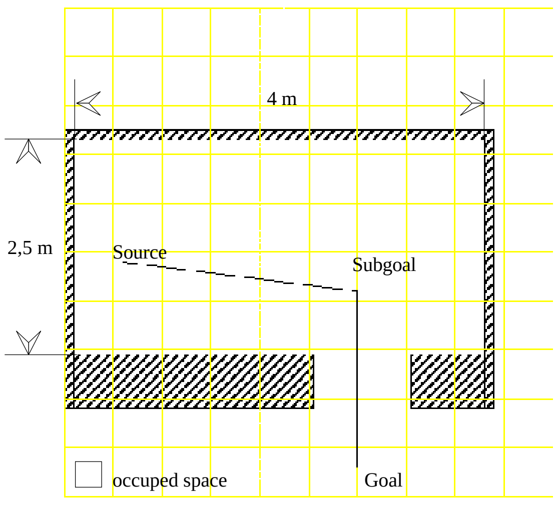

Figure 3 : *Planned path.*

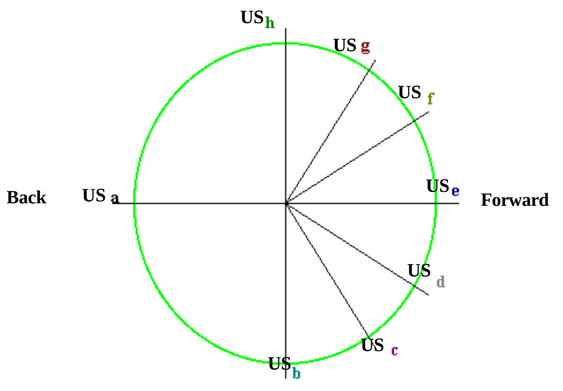

Figure 4 : *Layout of the ultrasonic sensors.*

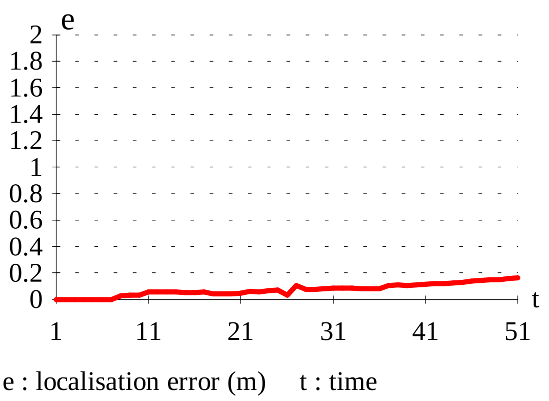

Figure 5 : *Error evolution without odometric degradation. Case of pin-point algorithm without memory.*

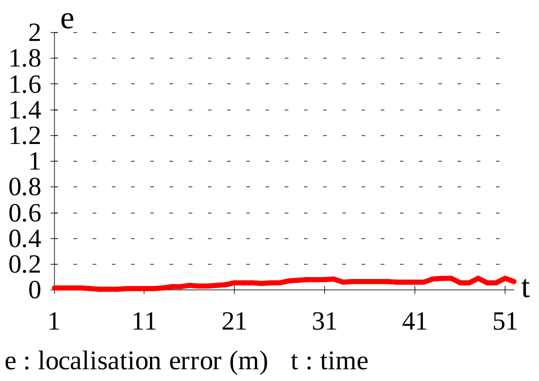

Figure 6 : *Error evolution without odometric degradation. Case of pin-point algorithm with memory.*

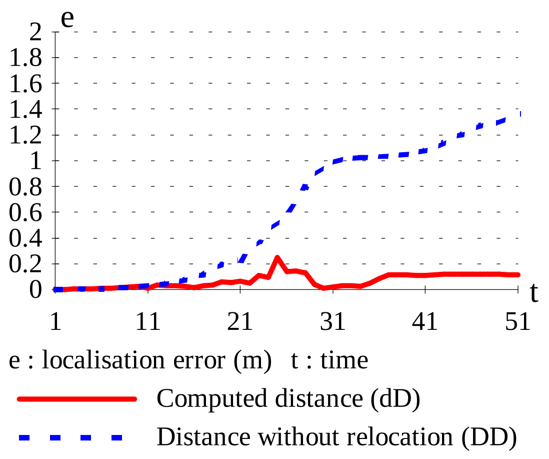

Figure 7 : *Error evolution with an 8% odometric degradation. Case of pin-point algorithm without memory.*

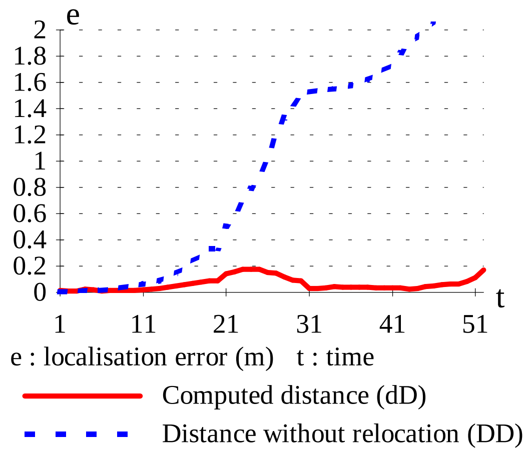

Figure 8 : *Error evolution with an 11% odometric degradation. Case of pin-point algorithm with memory.*

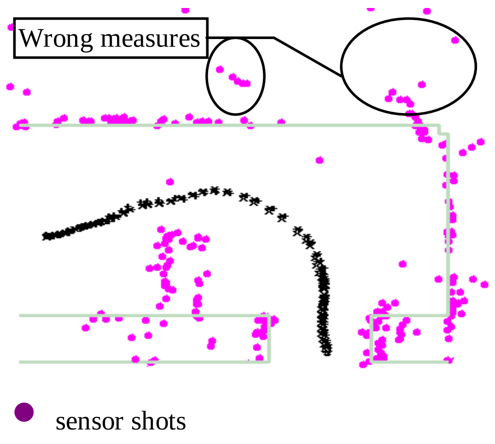

Figure 9 : *Robot trajectory without disturbance in a partially known environment.*

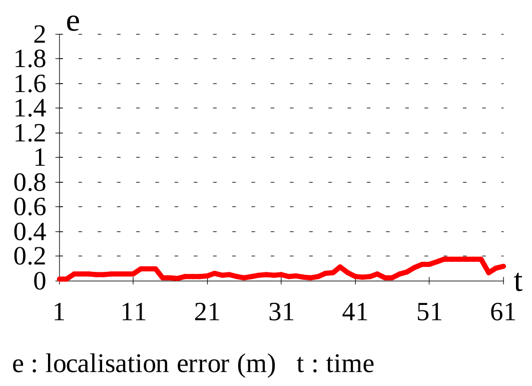

Figure 10 : *Error evaluation without odometric degradation in a partially known environment. Case of pin-point algorithm with memory.*

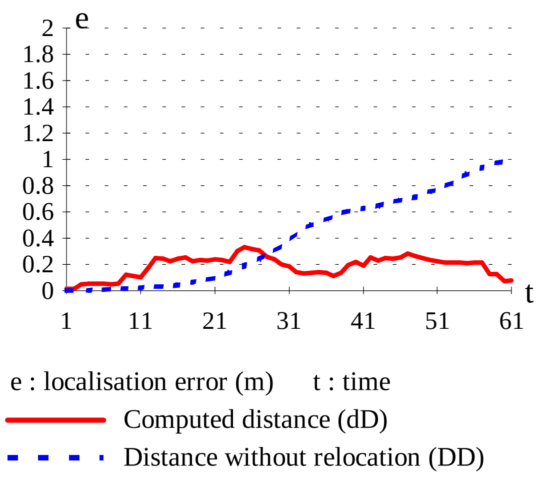

Figure 11 : *Error evaluation with a 4.5% odometric degradation in a partially known environment. Case of pin-point algorithm with memory.*

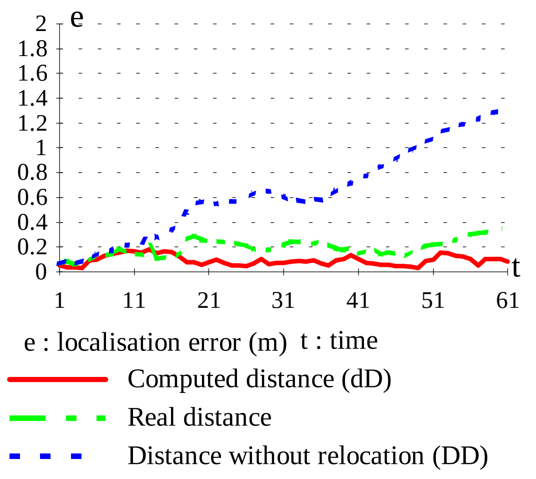

Figure 12 : *Error evaluation with a 9% odometric degradation.*

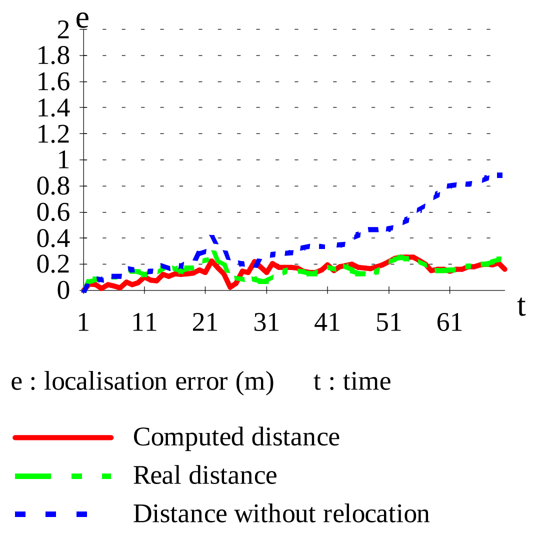

Figure 13 : *Real robot trajectory with a 4.5% odometric degradation.*

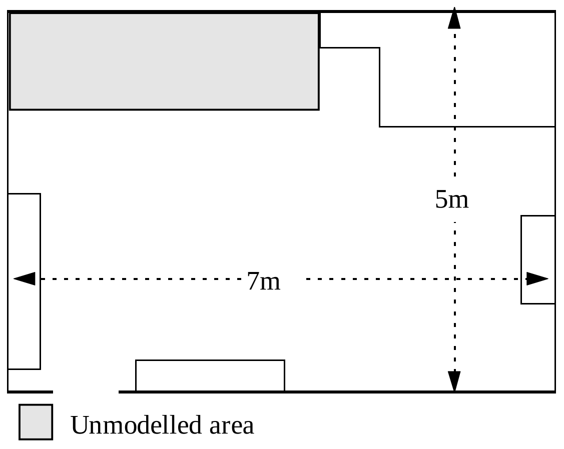

Figure 14 : *More complex environment.*

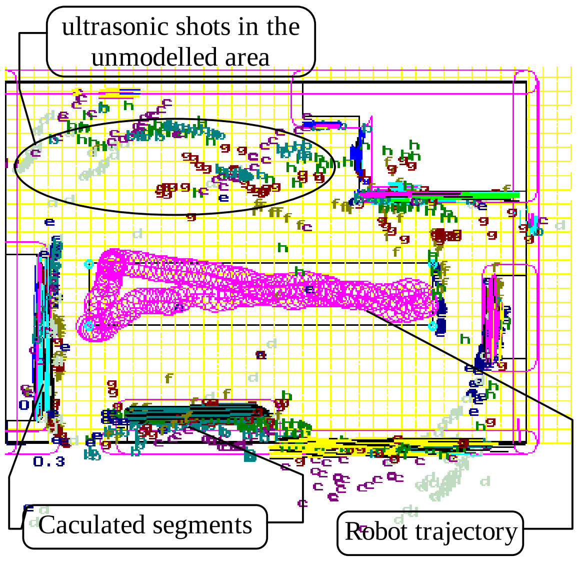

Figure 15 : *Trajectory in a complex environment.*

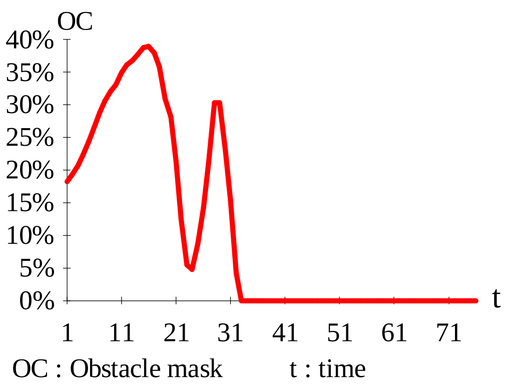

Figure 16 : *Percentage of obstacle masking.*

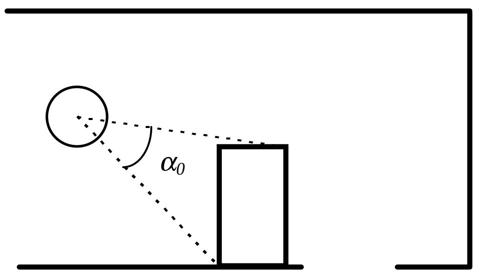

Figure 17 : *Definition of the angular masking of the environment by an obstacle.*

Figure 1. *Three areas defined for a segment.*

Figure 2 : *Position correction.*

Figure 3 : *Planned path.*

Figure 4 : *Layout of the ultrasonic sensors.*

Figure 5 : *Error evolution without odometric degradation. Case of pin-point algorithm without memory.*

Figure 6 : *Error evolution without odometric degradation. Case of pin-point algorithm with memory.*

Figure 7 : *Error evolution with an 8% odometric degradation. Case of pin-point algorithm without memory.*

Figure 8 : *Error evolution with an 11% odometric degradation. Case of pin-point algorithm with memory.*

Figure 9 : *Robot trajectory without disturbance in a partially known environment.*

Figure 10 : *Error evaluation without odometric degradation in a partially known environment. Case of pin-point algorithm with memory.*

Figure 11 : *Error evaluation with a 4.5% odometric degradation in a partially known environment. Case of pin-point algorithm with memory.*

Figure 12 : *Error evaluation with a 9% odometric degradation.*

Figure 13 : *Real robot trajectory with a 4.5% odometric degradation.*

Figure 14 : *More complex environment.*

Figure 15 : *Trajectory in a complex environment.*

Figure 16 : *Percentage of obstacle masking.*

Figure 17 : *Definition of the angular masking of the environment by an obstacle.*