Steps Towards the Automatic Evaluation of Robot Obstacle Avoidance Algorithms

I. Rañó
Dpto. de Informática e Ing. de Sistemas
Universidad de Zaragoza, Spain
irano@unizar.es

J. Minguez
I3A, Dpto. de Informática e Ing. de Sistemas
Universidad de Zaragoza, Spain
jminguez@unizar.es

***Abstract*— This paper presents the first steps towards the evaluation of obstacle avoidance techniques for mobile robots. The idea is to create a methodology to evaluate the performance of the methods given a wide range of work conditions. The work conditions usually include scenarios with very different nature (dense, complex, cluttered, etc). The performance is measured in terms of robotic parameters (robustness, optimality, safety, etc). We describe in this paper the overall methodology that we intend to apply and the first steps in the scenario characterization.**

I. Introduction

The *rationale* of the workshop explains very well the objective of this research:

> Current practice of publishing research results in robotics makes it extremely difficult not only to compare results of different approaches, but also to asses the quality of the research presented by the authors. Though for pure theoretical articles this may not be the case, typically when researchers claim that their particular algorithm or system is capable of achieving some performance, those claims are intrinsically unverifiable, either because it is their unique system or just because a lack of experimental details, including working hypothesis. (...).

It is clear that to overcome this issue, we have to find ways or processes to automatically evaluate the research with methodologies accepted by the community. Research on this topic forks into a *top-down* or *bottom-up* perspectives. On the one hand, the top-down manner consists in evaluating the complete robot performance when developing a given task (see [**?**], [**?**] and the majority of the speakers in the *First European Workshop on Benchmarks in Robotics Research* in 2006 Euron meeting). The advantage of this strategy is that it results straightforward to asses whether the robot has accomplished a given task. However, the disadvantage is that this requires a strong experimental validation with the real robots. Thus, rigorous protocols of experimentation have to be developed to deal with the repeatability problem of real experiments and to guarantee the checking of all possible situations.

On the other hand, the bottom-up manner deals with the evaluation of each single subtask individually. The aim is to explain the robot performance developing a task as the synergy of each particular performance in each of the involved subtasks. This paper follows this direction and focusses on the automatic evaluation of collision avoidance, which is a particular subtask involved in many applications of mobile robots.

This work is a part of a Spanish project. The objective is the evaluation of robots for mobility aid (like electric walkers or robotic wheelchairs). There are two main research axes: (i) the evaluation of robot motion from a robotic perspective and (ii) the evaluation of robot motion from a human-centered point of view. This paper describe the first steps towards the automatic evaluation of motion in the first axis, and the complete overview of the methodology that we will try to use.

II. Robot Obstacle Avoidance and Evaluation Perspective

This work focusses in one of the fundamental modules of sensor-based motion schemes: reactive obstacle avoidance. This module is the responsible of moving a vehicle to a given goal location while avoiding collisions with the scenario. Usually, it is the last responsible of the motion.

From a robotic perspective, there are many techniques that have been designed to address autonomous collision-free motion (sensor-based motion with obstacle avoidance). For example [3], [1], [2], [9], [5] among many others. It is clear that under the same conditions each technique generates a different motion. Nevertheless, questions like: which is the most robust one? or which of them behaves better in a determined context or condition? cannot be answered neither from a scientific nor technological point of view. In other words, once we face a mobile robotics application, the selection of a motion technique among all the existing ones is a matter of specialists and not accessible to everybody. This is because there are not objective comparisons of methods neither quantitative (in terms robustness or action parameters of such as the time or the total distance traveled) nor qualitative (in terms of security of the motion). At present, there is only one experimental comparison [4]. Nevertheless, this comparison is very old, and thus, it does not include the advances in this subject in the last 15 years. Furthermore it is based on the observation and does not present a rigorous and objective methodology to address this objective.

III. Overview of the Evaluation Framework

An obstacle avoidance technique is a mechanism that given an *obstacle configuration* and a *initial* and *goal* configurations computes the best motion to drive the vehicle to the goal while avoiding collisions with the obstacles. This process is repeated until the goal is reached. The result is a *trajectory* that joins the initial and the goal but generated online (since they work in an iterative process). Thus, the **inputs** of the problem are: (i) obstacle distribution and (ii) initial and goal configurations. The **output** is *success* if a collision free trajectory that joins the initial and goal has been computed. *Failure*: collision or the goal is not reached (usually referred as local minima). Thus, an evaluation of an obstacle avoidance mechanism should cover all the possible inputs (obstacles and initial and goal locations) and must be done on the basis of the quality of the output (collision free trajectory).

The methodology that we propose is to build a system able to generate random obstacle, initial and goal distributions and to evaluate the output of the method (trajectory) as a function of quantitative descriptors of the inputs and the outputs. The framework has the following modules (Figure 1):

1) *Scenario generation*: random generator of obstacle distributions and initial and goal locations.
2) *Scenario characterization*: extraction of the quantitative descriptors of the environment (e.g. density, clearness, etc).
3) *Collision Avoidance*: this is the technique to evaluate and is a “black-box” in the framework.
4) *Robot Simulator*: simulates the next state of the robot given the motion computed by the collision avoidance technique. To simplify the problem, in this work we assume that the robot is circular (with radius R) and holonomic without dynamic constraints. The sensor is assumed to measure range.
5) *Trajectory Evaluation*: extraction of the quantitative descriptors of the trajectory (e.g. optimality, safety, etc).

The final step is to describe the performance of the method as a function of this quantitative parameters. Notice that the evaluation measures how a given technique behaves for different environmental conditions. The environmental conditions are expressed in terms of density, confinement, clearness, etc. The behavior is described in terms of the success ratio, safety and optimality, etc.

The validation is performed exploring as much as possible the variability of the scenario. This forces the proposed validation methodology to work only on simulation, since the cost of evaluation on real environments is prohibitive. Notice that this process extrapolates for different methods giving an adequate framework for comparison.

We next describe the modules in detail.

A. Scenario generator

The scenario generator module creates a random obstacle distribution and goal and initial locations. These entities are represented geometrically as follows.

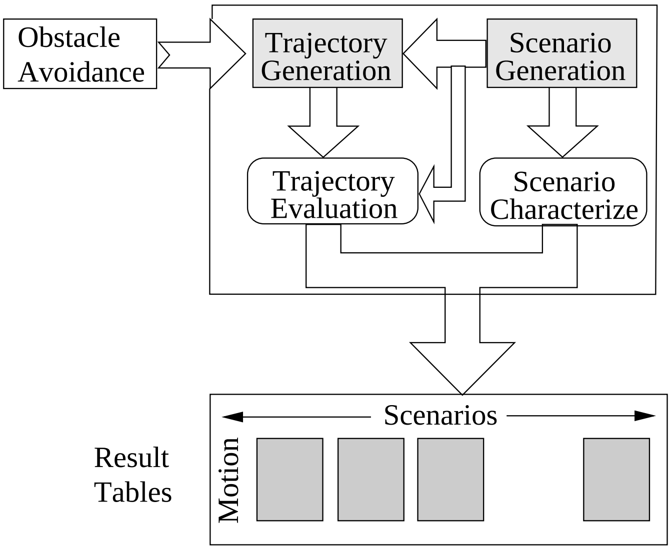


Fig. 1. This figure shows the modules of the evaluation framework

The scenario is a unitary ball with Euclidean metric. The obstacles are spheres within the unit ball. This is because spheres are a base of $\mathbf{R}^2$ and many shapes can be constructed with spheres (e.g. general polygons can be represented by circles [8]). In other words, the selection of circular obstacles is not a limitation. The number of obstacles is random. Furthermore, for each obstacle the location of the center and the radius are also random. Figure 2 shows two examples.

The initial and goal locations are also randomly generated within the unit ball. If any of the locations is within an obstacle, this location is recomputed. Furthermore, for any obstacle distribution and initial and goal locations, it is important to check the existence of a collision free path. Otherwise there is no solution for the obstacle avoidance technique. This is performed through a complete planner.

B. Scenario Characterization

This module extracts a quantitative evaluation of some descriptors of the environment. The aim of the scenario characterization is to extract numerical values that represent qualitative scenario features with a quantitative intrinsic values. The descriptors must be both related to the common

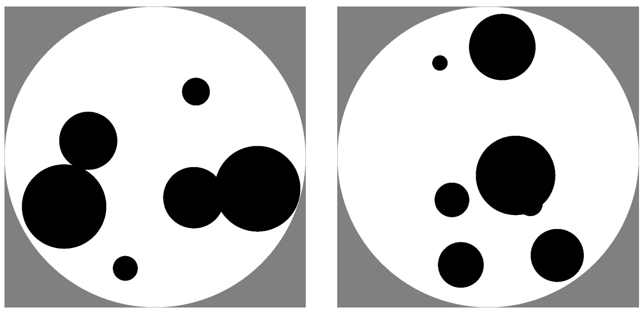


Fig. 2. Two examples of random scenarios with six and seven obstacles.

intuition of the qualitative variables that they define, and their values must be correlated with the performance of the obstacle avoidance algorithms. Table I describe the parameters and the human intuition qualitative descriptors.

Human Descriptor	*Mathematical Parameter*
Density	Density
Clearness	Dispersion
Confinement	Nearest Neighbor Metric
Uniformity	Discrepancy
Clutterness	Aleatory
Structure	Convex Hull
Others...	...

TABLE I
DESCRIPTORS OF THE SCENARIO.

We define these descriptors next. Some of them have been already developed while others need a deeper study.

a) Density: The density measures the amount of space occupied by obstacles. This is an intrinsic and global property of the environment, independent of the robot shape and size. Let be C_u the unitary sphere and lets assume a given distribution of spheres $\mathcal{C} = \{C_i\}$. The density of occupied space is:

$$\rho(\mathcal{C}) = \frac{A(\bigcup_i C_i)}{\pi} \tag{1}$$

where C_i is the i-th circle, and function $A(.)$ computes the area of a set of circles. Notice that the density is $\rho \in [0, 1]$. In other words is normalized. On one hand, when $\mathcal{C} = \{\emptyset\}$ the area is zero, hence the density is also zero. On the other hand, if $\mathcal{C} = \{C_\mathrm{u}\}$ the density of obstacles is one, which makes the scenario completely occupied and leaves no free space for the robot motion. Table II depicts some examples.

b) Clearness: The clearness is related to the maximum open space among the obstacle distribution. This descriptor depends on the size of the robot. One way of measuring open areas in distributions is the dispersion [6], since it measures the biggest obstacle free ball among the distribution. In other words, the value is the radius of the largest circle on the free space. Let be $\tilde{\mathcal{C}} = \mathcal{C} \oplus C_{\frac{\mathfrak{R}}{2}}$, where $\oplus$ is the Minkovski sum of sets and $C_{\frac{\mathfrak{R}}{2}}$ is the sphere with radius $\frac{R}{2}$. This set is the obstacle distribution enlarged the radius of the robot. The dispersion of $\mathcal{C}$ is:

$$\delta(\mathcal{C}) = \sup_{p \in C_\mathrm{u}} \left\{ \min_{C_i \in \mathcal{C}} \{||p - \tilde{C}_i||\} \right\} \tag{2}$$

where $||.||$ is the Euclidean distance from point p to the sphere $\tilde{C}_i$ ($\tilde{C}_i = C_i \oplus C_{\frac{\mathfrak{R}}{2}}$). Notice that the dispersion $\delta \in [0, 1]$. When there are no obstacles the value of the dispersion is one. However, as the number of obstacles and their radius increase the dispersion drops to zero. This characteristic captures the notion of clearness (open space) since it represents the maximum allowable distance for the robot to maneuver. Table II depicts some examples.

c) Confinement: The notion of confinement is related with the lack of space to manoeuvre (the distance between obstacles). This notion depends on the robot size reason why we use $\tilde{C}$. A measure is:

$$\kappa(\mathcal{C}) = 1 - \frac{1}{n} \sum_{i=1}^{n} \frac{d_{sph}(\tilde{C}_i - NN(\tilde{C}_i))}{2} \qquad , n > 1 \tag{3}$$

where d_{sph} is the Euclidean distance between two spheres and $NN(C_i)$ is the closest sphere to C_i (nearest neighborhood). Notice that the confinement is $\kappa \in [0, 1)$. High values of confinement explain obstacles which are very close among them, while low values are due to far obstacles.

d) Uniformity: The uniformity in the obstacle distribution refers to the match with a uniformly distributed set of obstacles. In fact this is measured by the discrepancy. Let be C_r the set of balls with radius r in the unit circle C_u. The discrepancy is:

$$\eta(\mathcal{C}) = \sup_{r \in [0,1]} \left\{ \left| \frac{A(\mathcal{C} \cap C_r)}{A(\mathcal{C})} - \frac{A(C_r)}{\pi} \right| \right\} \tag{4}$$

Notice that the discrepancy $\eta \in [0, 1]$. When there are no obstacles both $A(\mathcal{C}) = 0$ and $A(\mathcal{C} \cap C_r) = 0$ (discrepancy is not defined). Low values of discrepancy represent well distributed obstacle points. The discrepancy tends to one as the obstacles are closed and not equally distributed distributed in the space. Table II depicts some examples.

e) Cluttering: The cluttering refers to the order of the distribution. The disorder on a obstacle distribution is related with the randomness. We plan to use information theory to measure the randomness of sequences by using the tests of Martin-Lof [**?**] or Kolmogorov complexity of constructive measurement. Other works use the entropy as a descriptor [**?**] to measure the disorder. However, from our point of view entropy is a local descriptor that needs to approximate a probability distribution, which seems difficult to obtain from an obstacle distribution.

f) Structure: The structure of the scenario measures the the tendency of the scenario to approximate a polygonal world (man-made scenarios). The measure is, for all clusters, the normalized difference between a cluster of obstacles C^j (set of connected spheres) and its convex hull. The measure of the structure is

$$\gamma(\mathcal{C}) = 1 - \frac{1}{N} \sum_{j=1}^{N} (A(\hat{C}^j) - A(C^j)) \tag{5}$$

where $\hat{C}^j$ the convex-hull of C^j and N the number of clusters. Notice that $\gamma \in [0, 1]$. This measure gives an idea of the structure underlying the cluster. For example the value is one for a perfect line and zero for sparse non intersecting obstacles. We have notice that there are still some issues to fix with this measurement since it fails to represent well aligned non convex polygons. We are trying to correct it by sub-clustering each cluster with alignment criteria.

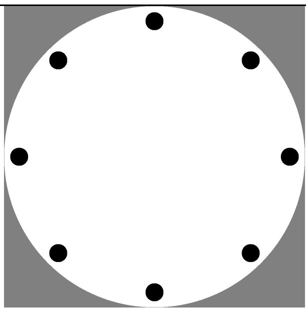
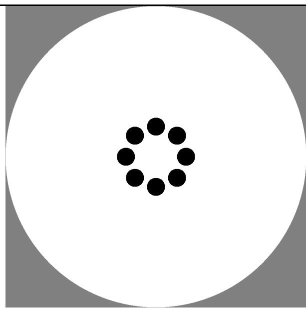
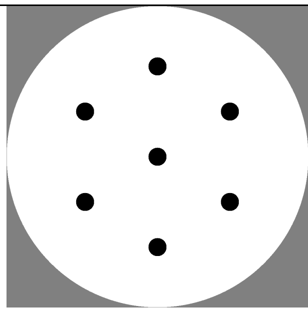
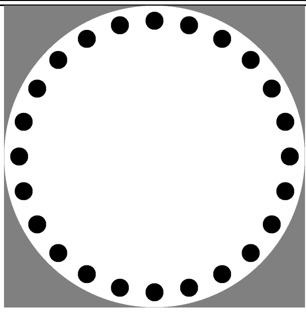
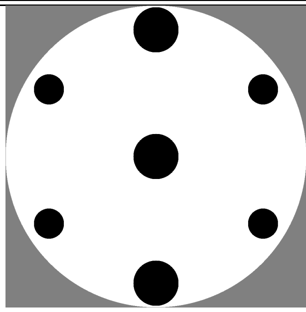
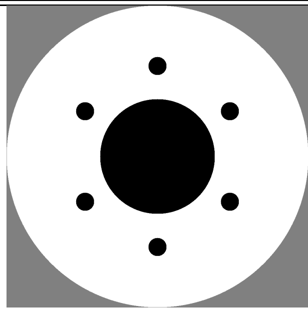
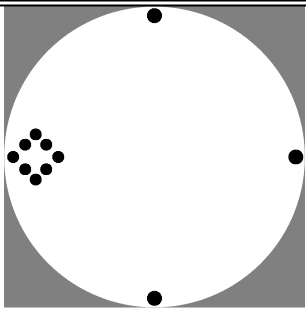
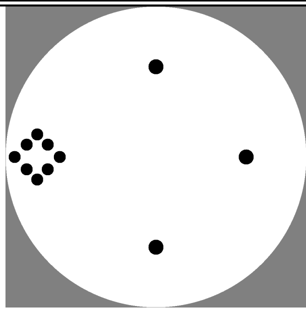
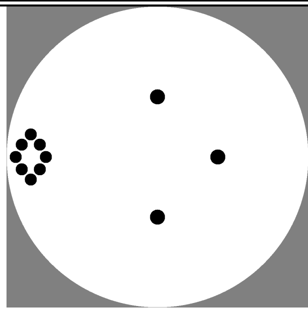
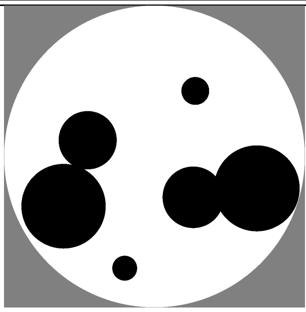
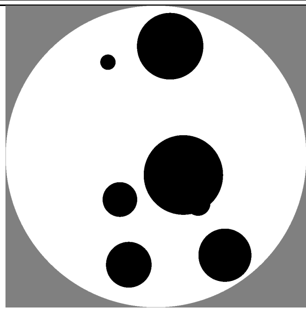
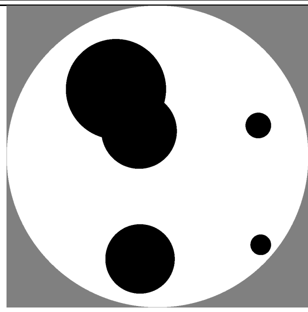
Environment			
Density	0.0287	0.0287	0.0252
Clearness	0.840	0.381	0.307
Confinement	?	?	?
Uniformity	0.138	0.447	0.184
Cluttering	?	?	?
Structure	?	?	?
Environment			
Density	0.086	0.107	0.166
Clearness	0.840	0.377	0.307
Confinement	?	?	?
Uniformity	0.137	0.139	0.244
Cluttering	?	?	?
Structure	?	?	?
Environment			
Density	0.020	0.020	0.020
Clearness	0.757	0.552	0.431
Confinement	?	?	?
Uniformity	0.390	0.390	0.425
Cluttering	?	?	?
Structure	?	?	?
Environment			
Density	0.251	0.189	0.210
Clearness	0.374	0.437	0.400
Confinement	?	?	?
Uniformity	0.461	0.379	0.444
Cluttering	?	?	?
Structure	?	?	?

TABLE II

CHARACTERIZATION OF SCENARIOS. THE SYMBOL ? MEANS THAT THE VALUE HAS NOT BEEN COMPUTED YET.

C. Robot Simulator

This module emulates the sensory and motion processes of the motion:

1) Sensory process: given the obstacle distribution and the current vehicle location, this module computes the *generalized visibility polygon* as generic range sensory measurement.
2) Motion process: given the motion computed by the obstacle avoidance method, this module computes the next robot state in a given period of time δt.

D. Trajectory Evaluation

This module extracts quantitative descriptors of quality of the trajectory. The aim of this characterisation is to extract numerical values that represent a qualitative measurement of the trajectory. We denote the trajectory generated by the method:

$$\begin{array}{rcl} \chi : [0,1] & \mapsto & C_u - \mathcal{C} \\ \phi & \mapsto & \chi(\phi) = \mathbf{q} \end{array} \tag{6}$$

where $\chi(0) = \mathbf{q_{init}}$ and $\chi(1) = \mathbf{q_{goal}}$. We describe next some of the parameters.

a) Success: This is the most important parameter since it describes the success of the task. Notice that the collision avoidance techniques are local techniques, and thus they could get trapped in local minima (not reaching the goal) or even to have collisions. In both cases ($\chi(1) \neq \mathbf{q_{goal}}$ or $\exists \phi$ such that $\chi(\phi) \in \mathcal{C}$) the result is *failure*. Then the success is $\eta(\mathcal{C}, \mathbf{q_{init}}, \mathbf{q_{goal}}) = [\{0\}, \{1\}]$. If the obstacle avoidance mechanism fails $\eta = 0$ otherwise $\eta = 1$. From now on the rest of the parameters are defined when $\eta = 1$.

b) Optimality: This parameters measures how the trajectory matches the optimal path. In order to compute the optimal trajectory χ_{opt} we have adapted the visibility graph technique [7] to work in spherical worlds[1]. Let be Φ an *optimality* function defined over the space, such that at each $\mathbf{q}$, $\Phi(\mathbf{q})$ is the length of the path with minimum length that joins $\mathbf{q}$ and the optimal trajectory χ (without lying in $\mathcal{C}$). Then there are some concepts that give the "difference" with the optimal path like the difference of *lengths* of the trajectories, or how the trajectory differs from the optimal by integrating $\Phi(\chi(\phi))$. We plan to implement both parameters.

Figure 3a,c show two different scenarios and the trajectories computed by an obstacle avoidance method (potential field method) and the optimal trajectory to the problem.

c) Safety: The safety measures how close the trajectory matches the safest trajectory. Notice that usually safest is far different from optimal (an optimal trajectory usually graze the obstacles, which is the un-safest trajectory). In order to compute the safest trajectory, we compute the Voronoi diagram of the obstacle distribution. Then, we define a Voronoi function

[1]The path is a sequence of straight lines and circular arcs instead of straight lines

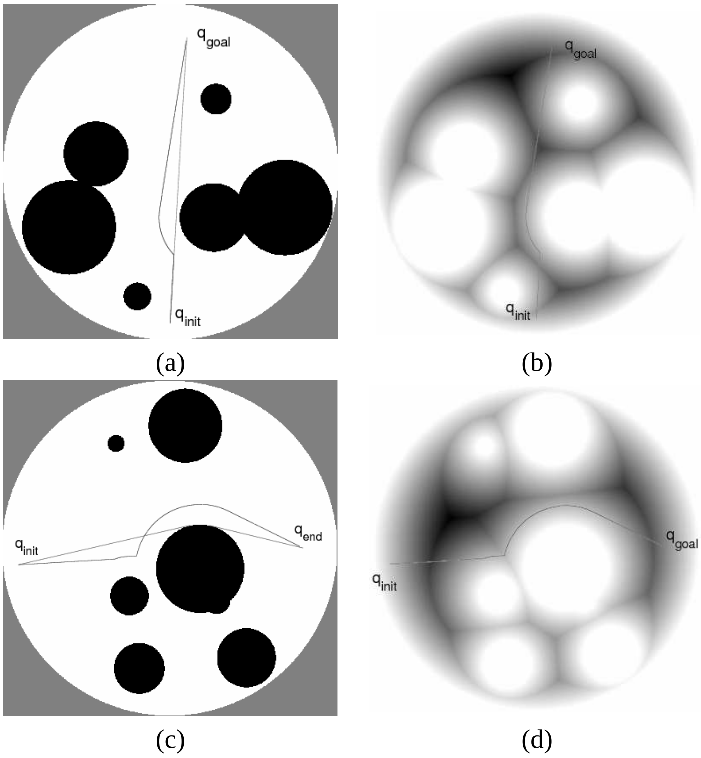


Fig. 3. This figures display the optimality function and the Voronoi function, and real trajectory generated with a PFM method on scenario from two scenarios.

$Vor(\mathbf{q})$ that takes values in all the space. The function is the length of the path with minimum length that joins $\mathbf{q}$ and the Voronoi diagram (without lying in $\mathcal{C}$).

Then there are some concepts that give the "difference" with the safest path (the path on the Voronoi diagram) like the difference of *lengths* of the trajectories, or how the trajectory differs from the safest by integrating $Vor(\chi(\phi))$. We plan to implement both parameters.

Figure 3b,d show two different scenarios, the trajectory computed by an obstacle avoidance method (potential field method) and an approximation of the Vor function.

d) Other Characteristics: Other parameters could be defined to characterize the trajectories, but they need to be matched against a ground truth. In general a good avoidance algorithm must display a tradeoff over the above mentioned characteristics, since for instance safety and path length optimality are usually a tradeoff. The characteristics have been restricted to the cinematic domain, other parameters like time optimality of a path are clearly dependent on the dynamics of the robot, and cannot be directly expressed under the present assumptions.

While the scenarios considered here are static, another interesting characteristic to take into account for dynamic environments is the speed of the response to an unexpected change. The latency of the algorithm is indeed an important issue in dynamic environments, which is related to the amount of history information internally stored on the algorithm. A purely reactive algorithm will show a better latency equal to

Density (ρ) VS Success	$[0, 0.25)$	$[0.25, 0.5)$	$[0.5, 0.75)$	$[0.75, 1]$
Success	95%	80%	68%	41%
Failure	5%	20%	32%	59%

TABLE III

METHOD 1: DENSITY VERSUS SUCCESS RATE.

Density (ρ) VS Success	$[0, 0.25)$	$[0.25, 0.5)$	$[0.5, 0.75)$	$[0.75, 1]$
Success	95%	95%	60%	20%
Failure	5%	5%	40%	80%

TABLE IV

METHOD 2: DENSITY VERSUS SUCCESS RATE.

Density VS Optimality Param.	$[0, 0.25)$	$[0.25, 0.5)$	$[0.5, 0.75)$	$[0.75, 1]$
$[0, 0.25)$	65%	50%	40%	25%
$[0.25, 0.5)$	20%	25%	10%	10%
$[0.5, 75)$	10%	15%	5%	5%
$[0.75, 1]$	5%	10%	3%	1%

TABLE V

METHOD 1: DENSITY VERSUS OPTIMALITY.

the computation time. This delay in the response is also related with a parameter that can be measured on the framework, even it has not been yet mentioned, the execution time of the avoidance cycle.

E. Final evaluation

The above mentioned process is repeated for a significant number of scenarios (obstacle distributions, initial and goal locations) for each of the methods X_i to evaluate. The final results should express the working conditions of each method. For each pair of descriptors (scenario / trajectory) the benchmark procedure produces a table.

We next describe some examples based on imaginary data. Table III and Table IV represent the scenario density versus the success ratio for methods X_1 and X_2. These tables describe how the algorithms behave for a different range of scenario density (as usual, the robustness of the method decreases as the density increases). This is very useful for comparison and selection. For example, if in the application the range of density is low, Method 2 is more robust than Method 1. However, if the density is larger, the Method 1 become more robust. If the density of the scenario can be a priori estimated, the selection is clear from these tables (even for non experts).

Table V represents the normalized optimality parameter for different density ranges. This table describes how the optimality in length of the paths generated change as a function of the density of the scenario.

Notice that from this evaluation one can extract conclusions about a given method and compare the performance of the methods among them.

IV. CONCLUSIONS AND FURTHER WORK

This paper presents the first steps towards the evaluation of obstacle avoidance algorithms for mobile robots. We understand that the benefits of this evaluation are twofold. On the one hand, it is useful for researchers and developers to have rigorous evaluation tools as a objective feedback of their designs. On the other hand, for non technical experts, the decision of what method is well suited for a given application is a matter of understanding the working conditions of the application, but not the technical details of each technique.

The project that includes this work is currently starting, hence only first steps results are presented. Even the overall evaluation methodology has been depicted, only parts of the scenario characterization has been actually obtained. We are aware that more characteristics must be defined as this is an actually ongoing project. However the defined characteristics seem to properly catch intuitive concepts about environments, which is a key element of obstacle avoidance evaluation.

REFERENCES

[1] J. Borenstein and Y. Koren. The Vector Field Histogram–Fast Obstacle Avoidance for Mobile Robots. *IEEE Transactions on Robotics and Automation*, 7:278–288, 1991.

[2] D. Fox, W. Burgard, and S. Thrun. The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics and Automation Magazine*, 4(1), 1997.

[3] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. Journal of Robotics Research*, 5:90–98, 1986.

[4] A. Manz, R. Liscano, and D. Green. A comparison of realtime obstacle avoidance methods for mobile robots. In R. Chatila and G. Hirzinger, editors, *Experimental Robotics II*, pages 301–316. Springer-Velag, 1991.

[5] J. Minguez and L. Montano. Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios. *IEEE Transactions on Robotics and Automation*, 20(1):45–59, 2004.

[6] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[7] N.J. Nilsson. A mobile automaton: An application of artificial intelligence techniques. In *Proceedings of the 1st International Joint Conference on Artificial Intelligence*, pages 509–520, 1969.

[8] J. O'Rourke and N. Badler. Decomposition of three-dimensional objects into shperes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(3):295–305, 1979.

[9] R. Simmons. The Curvature-Velocity Method for Local Obstacle Avoidance. In *IEEE Int. Conf. on Robotics and Automation*, pages 3375–3382, Minneapolis, USA, 1996.