Benchmarking Collision Avoidance Schemes for Dynamic Environments

Luis Martinez-Gomez, Thierry Fraichard

ICRA2009
Workshops & Tutorials

Benchmarking Collision Avoidance Schemes for Dynamic Environments

Luis Martinez-Gomez† and Thierry Fraichard†

***Abstract*— This paper evaluates and compare three state-of-the-art collision avoidance schemes designed to operate in dynamic environments. The first one is an extension of the popular Dynamic Window approach; it is henceforth called TVDW which stands for *Time-Varying Dynamic Window*. The second one called NLVO builds upon the concept of *Non Linear Velocity Obstacle* which is a generalization of the Velocity Obstacle concept. The last one is called ICS-AVOID, it draws upon the concept of *Inevitable Collision States*, *ie* states for which, no matter what the future trajectory of the robotic system is, a collision eventually occurs. The results obtained show that, when provided with the same amount of information about the future evolution of the environment, ICS-AVOID outperforms the other two schemes. The primary reason for this has to do with the extent to which each collision avoidance scheme reasons about the future. The second reason has to do with the ability of each collision avoidance scheme to find a safe control if one exists. ICS-AVOID is the only one which is complete in this respect thanks to the concept of Safe Control Kernel.**

***Index Terms*— Motion Safety; Collision Avoidance; Dynamic Environments; Inevitable Collision States, Velocity Obstacles, Dynamic Window.**

I. INTRODUCTION

A. Background and Motivations

Autonomous mobile robots/vehicles navigation has a long history by now. Remember Shakey's pioneering efforts in the late sixties [1]. Today, the situation has dramatically changed as illustrated rather brilliantly by the 2007 DARPA Urban Challenge[1]. The challenge called for autonomous car-like vehicles to drive 96 kilometers through an urban environment amidst other vehicles (11 self-driving and 50 human-driven). Six autonomous vehicles finished the race thus proving that autonomous urban driving could become a reality. Note however that, despite their strengths, the Urban Challenge vehicles have not yet met the challenge of fully autonomous urban driving (how about handling traffic lights or pedestrians for instance?).

Another point worth mentioning is that at least one collision took place between two competitors. This unfortunate mishap raises the important issue of *motion safety*, *ie* the ability for an autonomous robotic system to avoid collision with the objects of its environment. The size and the dynamics of the Urban Challenge vehicles make them potentially dangerous for themselves and their environment (especially when driving at high-speed). Therefore, before letting such autonomous systems transport around or move among people, it is vital to assert their ability to avoid collisions.

In the last forty years, the number and variety of autonomous navigation schemes that have been proposed is huge (*cf* [2]). In general, these navigation schemes intend to fulfill two key purposes: reach a goal and avoid collision with the objects of the environment. When it comes to collision avoidance, once again, many collision avoidance schemes have been proposed. Their aim of course is to ensure the robotic systems' safety. However, the analysis carried out in [3] of the most prominent navigation schemes (*ie* the ones currently used by robotics systems operating in real environments, *eg* [4]–[7]) shows that, especially in environments featuring moving objects, *motion safety is not guaranteed* (in the sense that collisions can occur even if they have full knowledge of the environment future evolution: no uncertainty or spurious information). As shown in [3], collision avoidance in dynamic environments is complex since it requires to explicitly reason about the *future behaviour* of the moving objects with a *time horizon*, *ie* the duration over which the future is taken into account, which is determined by the nature of both the moving objects and the robotic system at hand. Failure to do so yields collision avoidance schemes with insufficient motion safety guarantees.

B. Contributions

The primary purpose of this paper is to explore this time horizon issue and to show how important it is in the design of a truly safe collision avoidance scheme. To that end, this paper will evaluate and compare three state-of-the-art collision avoidance schemes that have been explicitly designed to handle dynamic environments. The first one is from [8] and is henceforth called *Time-Varying Dynamic Window* (TVDW), it is a straightforward extension of the popular Dynamic Window approach [6]. The second one builds upon the concept of *Non Linear Velocity Obstacle* (NLVO) [9] which is a generalization of the Velocity Obstacle concept [7]. The last one, ICS-AVOID [10], draws upon the concept of *Inevitable Collision States* developed in [11] (*aka* Obstacle Shadow [12] or Region of Inevitable Collision [13], [14]). The three collision avoidance schemes do reason about the future evolution of the environments but they do so differently, each scheme has its own time horizon.

When placed in the same environment and provided with exactly the same amount of information about the future, the results we have obtained show that ICS-AVOID performs significantly better than the other two schemes.

†INRIA, CNRS-LIG & Grenoble University, France.

[1]http://www.darpa.mil/grandchallenge.

The primary reason for this has to do with the way each collision avoidance scheme uses the information about the future, thus emphasizing the fact that, reasoning about the future is not nearly enough, it must be done with an appropriate time horizon. In contrast with TVDW and NLVO, ICS-AVOID is the only scheme that reasons over an infinite time-horizon. The analysis carried out in [10] shows that if ICS-AVOID were provided with full knowledge about the future, it would guarantee motion safety no matter what. Now, it could be argued that infinite knowledge about the future is not available in realistic cases (which is true). The fact remains that ICS-AVOID is the only scheme that is able to make full use of all the information about the future which is provided.

The second reason has to do with the decision part of each collision avoidance scheme. In all cases, their operating principle is to first characterize forbidden regions in a given control space and then select an admissible control, *ie* one which is not forbidden. Accordingly motion safety also depends on the ability of the collision avoidance scheme at hand to find such admissible control. In the absence of a formal characterization of the forbidden regions, all schemes resort to sampling (with the inherent risk of missing the admissible regions). In contrast, ICS-AVOID through the concept of *Safe Control Kernel* is the only one for which it is guaranteed that, if an admissible control exists, it will be part of the sampling set.

C. Outline of the Paper

The paper is organized as follows: Section II gives an overview of the collision avoidance schemes used for the comparative evaluation: TVDW, NLVO and ICS-AVOID. Afterwards, Section III details the way each collision avoidance scheme reasons about the future. Section IV describes the benchmarking and simulation setup. The benchmark results are presented in Section V. Discussion and concluding remarks are made in Section VI.

II. STATE-OF-THE-ART COLLISION AVOIDANCE SCHEMES

As exposed in the introduction, the benchmarking concerns TVDW, NLVO and ICS-AVOID. The first two are extensions to popular collision avoidance schemes used in real-world applications: Dynamic Window (DW) and Velocity Obstacles (VO). DW has been demonstrated at relatively high speeds (up to $1\ m/s$) in complex environments with Minerva [15], Rhino [16] and Robox [17], robotic tour-guides that have operated for different time periods in different places in the United States, Germany and Switzerland. VO has been tested with MAid [18], an automated wheelchair that navigated in the concourse of the central station in Ulm (DE) and during the German exhibition Hanover Fair'98. ICS-AVOID, is the continuation of the work done around the ICS concept for safe motion planning in dynamic environments [19], [20] with applications in driverless vehicles [21], [22].

A. Time Varying Dynamic Window

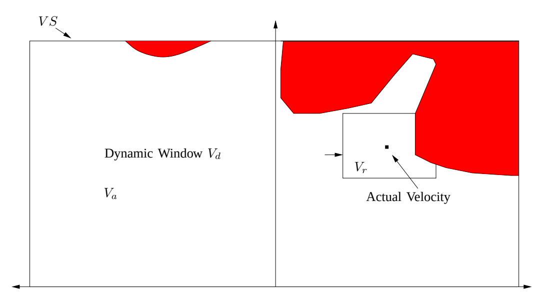
The Dynamic Window approach is a velocity space based local reactive avoidance scheme where search for admissible controls is carried out directly in the linear and angular velocity space [6]. The search space is reduced by the system kinematic and dynamic constraints to a set of reachable velocities (V_r) in a short time interval (Δt) around the current velocity vector (Fig.1a):

$$V_r = \{(v,\omega) | v \in [v_c - \dot{v_b}\Delta t, v_c + \dot{v_a}\Delta t] \wedge \\ \omega \in [\omega_c - \dot{\omega_b}\Delta t, \omega_c + \dot{\omega_a}\Delta t]\} \quad (1)$$

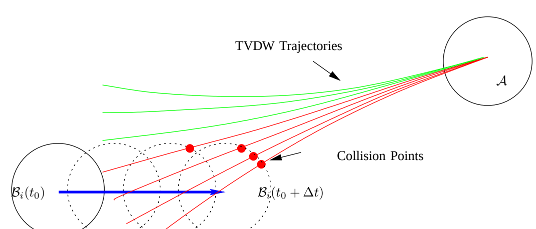
where $\dot{v_a}$, $\dot{\omega_a}$, $\dot{v_b}$ and $\dot{\omega_b}$ are maximal translational/rotational accelerations and breaking decelerations. A velocity is admissible (V_a) if it allows the system to stop before hitting an object:

$$V_a = \{v, \omega \leq \sqrt{2\rho_{min}(v,\omega)\dot{v_b}} \wedge \sqrt{2\rho_{min}(v,\omega)\dot{\omega_b}}\} \quad (2)$$

An admissible velocity optimizing a given cost function is selected at each time step. This approach considers the objects in the environment as static. TVDW extends this scheme by calculating at each instant a set of immediate future obstacles trajectories in order to check for collision in the short term [8]. In this respect TVDW is superior to DW because it reasons about the future behaviour of the obstacles. The extent of the look ahead time is set to equal the time it takes to the robotic system to stop, if no collision occurs during that time the velocity is considered admissible (Fig.1b).



(a) Dynamic Window.



(b) Time-Varying Dynamic Window.

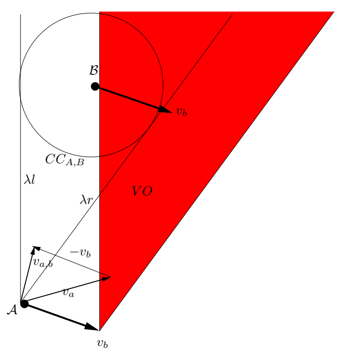
Fig. 1: Dynamic Window based approaches.

B. Non-Linear Velocity Obstacles

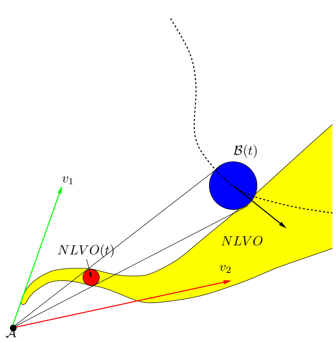
Velocity Obstacles is a reactive approach that operates in the Cartesian velocity space of the robotic system considered [7]. VO takes into account the velocity of the moving objects (assumed to be moving with a constant linear velocity). Each object yields a set of forbidden velocities whose shape is that of a cone (Fig.2a depicts the linear velocity space of the robotic system, the red conical region on the right is the set of forbidden velocities that would yield a collision between the robot $\mathcal{A}$ and the moving object $\mathcal{B}$). Should the robotic system select a forbidden velocity, it would collide with the moving object at a later time (possibly infinite) in the future. In practice, velocities yielding a collision occurring after a given time horizon (t_H) are considered as admissible. NLVO is an extension of VO that considers known arbitrary velocity profiles for the moving objects [9]. NLVO consist of all velocities of $\mathcal{A}$ at t_0 that would result in collision with $\mathcal{B}$ at any time $t_0 \leq t \leq t_H$. As depicted in Figure.2b, $NLVO(t)$ is a scaled down $\mathcal{B}$, bounded by the cone formed between $\mathcal{A}$ and $\mathcal{B}(t)$, thus, NLVO is a warped cone with apex at $\mathcal{A}$ and formally defined as:

$$NLVO = \bigcup_{t_0 \leq t \leq t_H} \frac{\mathcal{B}(t)}{t - t_0} \quad (3)$$

where $\frac{\mathcal{B}(t)}{t-t_0}$ is the set $\mathcal{B}(t)$ scale down by $(t - t_0)$. One issue (often overlooked) with the VO representation is that, in a closed environment, every velocity is forbidden since it eventually yield a collision. For that reason, both VO and NLVO require a time horizon t_H that cannot be arbitrarily large.



(a) Velocity Obstacles.



(b) Non Linear Velocity Obstacles.

Fig. 2: Velocity Obstacles based approaches.

C. ICS-AVOID

ICS-AVOID is a reactive navigation approach based upon the concept of Inevitable Collision State (ICS) [11]. An ICS is defined as a state for which, no matter what the future trajectory followed by the system is, a collision eventually occurs. ICS-AVOID searches the control space of the system for a control to apply at the next time step. A control is admissible if it drives the system to a non-ICS state. To test for ICS-ness the ICS-Checker presented in [23] is used. If the current state is not an ICS then it is guaranteed that ICS-AVOID will find and return an admissible control (Safe Control Kernel) [10].

III. REASONING ABOUT THE FUTURE

All the collision avoidance schemes used in the benchmarking make use of a model of the future, that is, they take into account the future behaviour of the obstacles in the environment. The different extent in which they use the available information have an impact in the decisions they made and consequently in their overall performance. TVDW considers as look ahead the braking time t_B (the time it takes to the system to go from its current velocity to a halt). This time is then state dependent and upper-bounded. NLVO use as look ahead an arbitrarily set time horizon (t_H), in other words, there is no clear guideline on how to choose it an is not a function of the system dynamics nor current state. Furthermore, it can't be set to a very large value because in closed environments it will render all velocities inadmissibles. ICS-AVOID in accordance to the ICS defintion reasons in terms of infinite duration. It uses the available information about the unfolding of the environment up to infinity. The different look ahead of the collision avoidance schemes is illustrated in Fig.3 to emphasize the fact that both TVDW and NLVO truncate their future model and disregard any information beyond t_B and t_H respectively (even if it's available). In contrast, this isn't the case for ICS-AVOID.

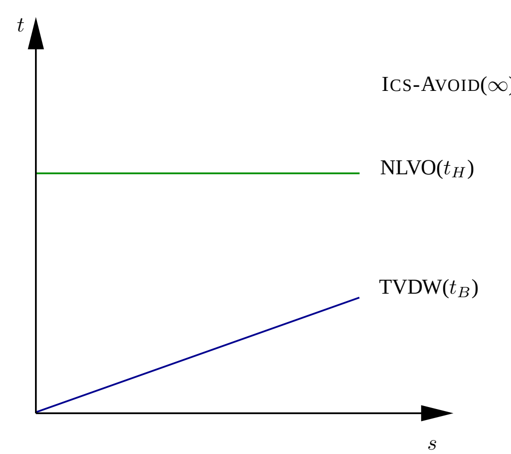


Fig. 3: Look-ahead of the different schemes.

IV. BENCHMARK AND SIMULATION SETUP

To assess the performance of the collision avoidance schemes just presented a comparative evaluation was conducted. A simulation environment capable of reproducing the same conditions for all the schemes was chosen to conduce the benchmarking. The robotic system, environment setup and implementation is discussed next.

1) Robotic System: Point Mass Model: Let $\mathcal{A}$ be modeled as a disk with point mass non-dissipative dynamics. A *state* of $\mathcal{A}$ is defined as $s = (x, y, v_x, v_y)$ where (x, y) are the coordinates of the center of the disk and v_x, v_y are the axial components of the velocity. A control of $\mathcal{A}$ is defined by the pair (u_x, u_y) which denote the force exerted by the actuators along the x- and y-axis respectively. The motion of $\mathcal{A}$ is governed by the following differential equations:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{v_x} \\ \dot{v_y} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_y \qquad (4)$$

with a bound in the control given by the maximum acceleration: $\frac{u_x^2+u_y^2}{m^2} \leq a_{max}^2$ where m is the robot mass.

2) Workspace Model: $\mathcal{A}$ moves in a closed 2D workspace $\mathcal{W}$ (100 by 100 meters), cluttered up with disk-shaped moving objects (grown by the radius of $\mathcal{A}$). A total of twenty three objects move with random constant speeds (between 1 to 10 m/s) along complex cyclic trajectories (closed B-splines with 10 random control knots). Figure 4 shows the trajectories of the objects to illustrate the complexity of the environment. This setup can theoretically provide future information about the behaviour of the moving objects up to infinity. In practice, knowledge is provided until a fixed time in the future t_F after which constant linear motion is assumed (Fig. 5). This to resemble realistic cases where prediction quality degrades as time pass by.

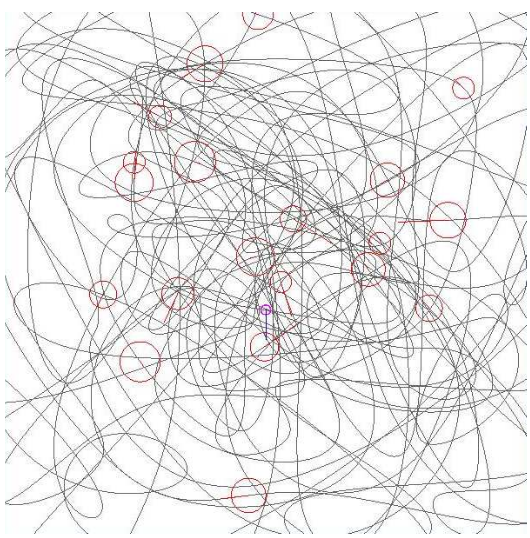


Fig. 4: Workspace example, 23 obstacles (represented by circles) with random generated velocities and B-Splines trajectories.

3) Implementation: The simulation environment and collision schemes were programmed entirely in C++ using OpenGL as rendering engine. The random number generator employed to produce the obstacles trajectories and velocities was seeded with a set of identical numbers to achieve an identical reproduction of simulation conditions for each of the collision avoidance schemes in the benchmark. The information about the future behaviour of the objects in the environment was made available to all the schemes with a limit of $t_F = 1$, 3 and 5 seconds into the future.

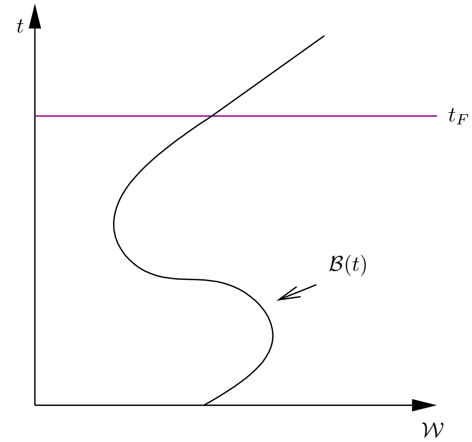


Fig. 5: World Model of the future.

V. BENCHMARK

The collision avoidance schemes were tested on a set of five runs with a duration of two minutes each. We varied the amount of available information about the future behaviour of the obstacles in the environment with $t_F = 1$, 3 and 5 seconds. For each run the number of collisions between $\mathcal{A}$ and the objects $\mathcal{B}_i$ are recorded in Table I.

Scheme	Run	Collisions TF=1(s)	Collisions TF=3(s)	Collisions TF=5(s)
TVDW	1	5	6	3
	2	12	4	4
	3	5	7	3
	4	12	2	4
	5	12	2	4
Average:		9.2	4.2	3.6
NLVO	1	10	2	0
	2	8	2	0
	3	12	2	0
	4	3	3	2
	5	7	2	2
Average:		8.0	2.2	0.8
ICS-AVOID	1	7	0	0
	2	0	0	0
	3	1	0	0
	4	1	0	0
	5	1	0	0
Average:		2.0	0.0	0.0

TABLE I: Benchmarking of collision avoidance schemes.

TVDW (Fig. 6) performs poorly in comparison with the other two schemes. One of the main causes of failure is the limited extent in which the scheme use the information available about the future trajectories of the objects: as explained before it limits itself to a small fraction of the time at hand (t_B). In contrast, NLVO (Fig. 7) exploits better the given information. In these runs t_H was set equal to t_F so all the available information could be taken into account. NLVO averages less of one collision per run in the 5 second setup, nonetheless, it fails to guarantee the safety of the system when provided with less information. ICS-AVOID (Fig. 8) has the best performance in all the time setups. ICS-AVOID is designed to reason in terms of infinite duration but even when dealing with minimal information about the future (1 second) it outperfomed the other two schemes. When given more information (3 and 5 seconds) not a single collision occured. The results show the importance of the look ahead time, when a colllision avoidance scheme disregard available information its performance is lower compared to those that use more.

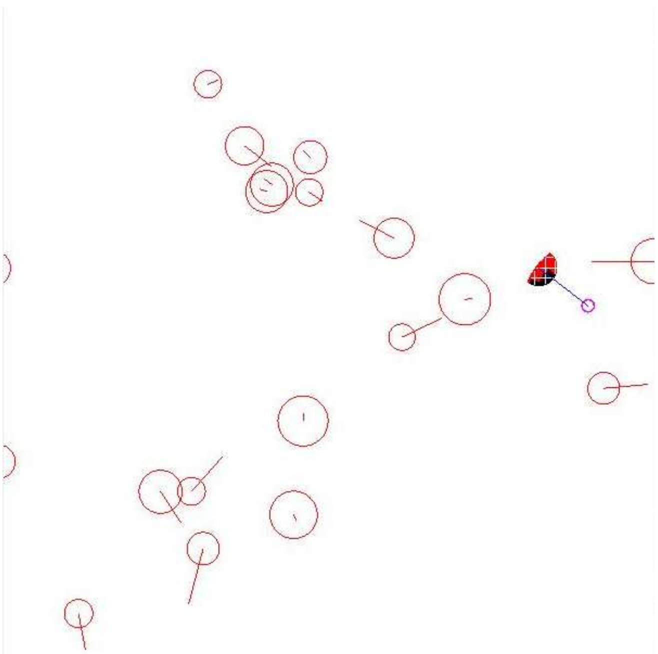


Fig. 6: TVDW. Admissible velocities (V_a) are represented in black, velocities in red are forbidden.

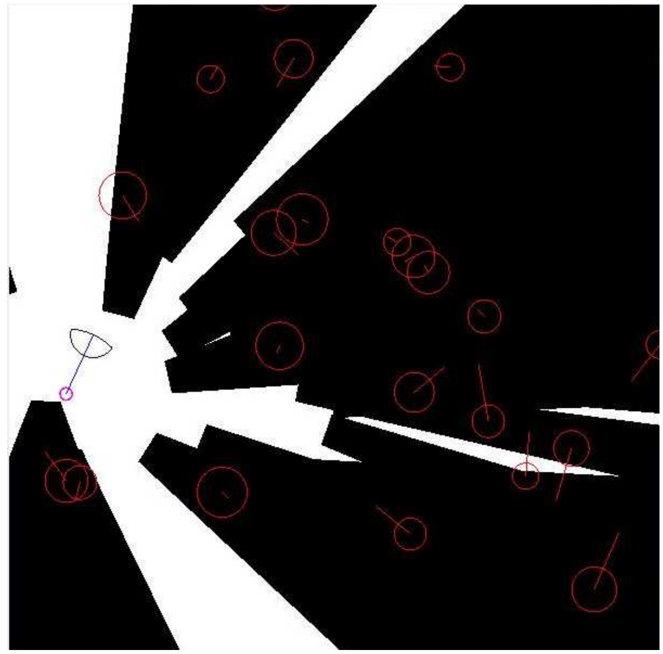


Fig. 7: NLVO. Black warped cones are forbidden velocities for the robotic system.

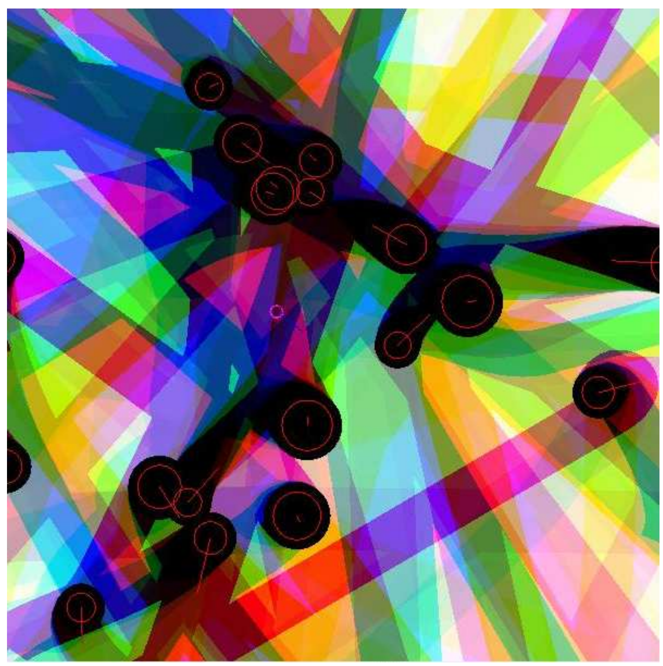


Fig. 8: ICS-AVOID. Black regions are forbidden states (ICS).

VI. CONCLUSION

We have presented a comparative evaluation with three state-of-the-art collision avoidance schemes designed to handle complex dynamic environments. The results show that, when provided with the same amount of information about the future evolution of the environment, ICS-AVOID outperforms the others. The reason for this has to do with the extent to which each collision avoidance scheme reasons about the future.

ACKNOWLEDGEMENTS

This work has been partially supported by the European Commission contract "Have-It FP7-IST-2007-212154".

REFERENCES

[1] N. J. Nilsson, "Shakey the robot," AI Center, SRI International, Menlo Park, CA (US), Technical note 323, Apr. 1984.
[2] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.
[3] T. Fraichard, "A short paper about motion safety," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Roma (IT), Apr. 2007.
[4] J. Borenstein and Y. Korem, "The vector field histogram — fast obstacle avoidance for mobile robts," *IEEE Trans. on Robotics and Automation*, vol. 7, no. 3, June 1991.
[5] J. Minguez and L. Montano, "Nearness diagram (ND) navigation: collision avoidance in troublesome scenarios," *IEEE Trans. on Robotics and Automation*, vol. 20, no. 1, pp. 45–59, Feb. 2004.
[6] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, Mar. 1997.

[7] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. Journal of Robotics Research*, vol. 17, no. 7, July 1998.

[8] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," *Proc. of the Int. Conf. on Robotics and Automation*, Apr. 2007.

[9] F. Large, C. Laugier, and Z. Shiller, "Navigation among moving obstacles using the NLVO : Principles and applications to intelligent vehicles," *Autonomous Robots Journal*, vol. 19, no. 2, pp. 159–171, September 2005.

[10] L. Martinez-Gomez and T. Fraichard, "Collision avoidance in dynamic environments: an ics-based solution and its comparative evaluation," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Kobe (JP), May 2009.

[11] T. Fraichard and H. Asama, "Inevitable collision states. a step towards safer robots?" *Advanced Robotics*, vol. 18, no. 10, pp. 1001–1024, 2004.

[12] J. Reif and M. sharir, "Motion planning in the presence of moving obstacles," in *Proc. of the IEEE Int. Symp. on Foundations of Computer Science*, Portland, OR (US), Oct. 1985.

[13] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Detroit, MI (US), May 1999.

[14] N. Chan, M. Zucker, and J. Kuffner, "Towards safe motion planning for dynamic systems using regions of inevitable collision," in *Proc. of the workshop on Collision-free Motion Planning for Dynamic Systems*, Rome (IT), Apr. 2007, workshop held in association with the IEEE Int. Conf. on Robotics and Automation.

[15] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Minerva: A second generation mobile tour-guide robot," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Detroit, MI (US), May 1999.

[16] W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence*, vol. 114, no. 1-2, 2000.

[17] R. Philippsen and R. Siegwart, "Smooth and efficient obstacle avoidance for a tour-guide robot," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Taiwan (TW), Sept. 2003.

[18] E. Prassler, J. Scholz, and P. Fiorini, "A robotic wheelchair for crowded public environments," *IEEE Robotics and Automation Magazine*, vol. 8, no. 1, pp. 38–45, Mar. 2001.

[19] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, AB (CA), Aug. 2005.

[20] R. Parthasarathi and T. Fraichard, "An inevitable collision state-checker for a car-like vehicle," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Roma (IT), Apr. 2007.

[21] R. Benenson, S. Petti, T. Fraichard, and M. Parent, "Toward urban driverless vehicles," *Int. Journal of Vehicle Autonomous Systems*, vol. 6, no. 1/2, 2008.

[22] L. Bouraoui, S. Petti, A. Laouiti, T. Fraichard, and M. Parent, "Cybercar cooperation for safe intersections," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, Toronto, ON (CA), September 2006.

[23] L. Martinez-Gomez and T. Fraichard, "An efficient and generic 2d inevitable collision state-checker," in *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Nice (FR), Sept. 2008.