A Framework for Reactive Motion and Sensing Planning: A Critical Events-based Approach

R. Murrieta-Cid[1], A. Sarmiento[2], T. Muppirala[2], S. Hutchinson[2], R. Monroy[1], M. Alencastre[1], L. Muñoz[1], and R. Swain[1]

[1] Tec de Monterrey Campus Estado de México,
{rafael.murrieta, raulm, malencastre, lmunoz, rswain}@itesm.mx
[2] University of Illinois at Urbana-Champaign
{asarmien, muppiral, seth}@uiuc.edu

Abstract. We propose a framework for reactive motion and sensing planning based on critical events. A critical event amounts to crossing a critical curve, which divides the environment. We have applied our approach to two different problems: i) object finding and ii) pursuit-evasion. We claim that the proposed framework is in general useful for reactive motion planning based on information provided by sensors. We generalize and formalize the approach and suggest other possible applications.

1 Introduction

We propose a framework for reactive motion and sensing planning based on critical events. A critical event amounts to crossing a critical curve, which divides the environment. We work at the frontiers of computational geometry algorithms and control algorithms. The originality and the strength of this project is to bring both issues together.

We divide the environment in finitely many parts, using a discretization function which takes as input sensor information. Thus, in our approach, planning corresponds to switching among a finite number of control actions considering sensor input. This approach naturally allows us to deal with obstacles.

We have applied our approach to several different problems, here for lack of space we only present two: i) object finding and ii) pursuit-evasion. In object finding, our approach produces a continuous path, which is optimal in that it minimizes the expected time taken to find the object. In pursuit-evasion, we have dealt with computing the motions of a mobile robot pursuer in order to maintain visibility of a moving evader in an environment with obstacles.

Our solutions to these two problems have been published elsewhere. In this paper we show that these solutions actually rely on the same general framework. We claim that the proposed framework is in general useful for reactive motion planning based on information provided by sensors. We generalize and formalize the approach and suggest other possible applications.

A. Gelbukh, A. de Albornoz, and H. Terashima (Eds.): MICAI 2005, LNAI 3789, pp. 990–1000, 2005.

2 A General Framework

The crux of our approach consists of relating critical events with both the controls to be applied on the robot and the robot environment representation. A *critical event* signals that the robot has crossed a critical curve drawn on the robot workspace, $\boldsymbol{W}$. It corresponds to changes in the sensors information readings, driving our algorithms.

We use $\boldsymbol{C}$ and U to respectively denote the robot configuration space and the robot control space (velocity vectors applied to the robot). $P \subset \boldsymbol{C} \times U$, the robot phase space, is the cross product of $\boldsymbol{C}$ and U. Critical curves are projections on the workspace of P. This means that even if a configuration is valid to accomplish a task, it may not be valid due to the velocity related with that configuration. Hence, the critical curves may change their location according to a given robot velocity.

Let Y denote the observation space, which corresponds to all possible sensor readings. The robot state space is $X \subset \boldsymbol{C} \times E$, in which E is the set of all possible environments where the robot might be [12]. Evidently, there is a relation between the robot state $x(t)$ and the robot observation state $y(t)$ which is a function of time. Thus, $y(t) = h(x(t))$, where $y \in Y$ and $x \in X$. The robot information state is defined as $i_t = (u_0, \ldots, u_{t-1}, y_o, \ldots, y_t)$. i_t is the history of all sensor readings up to time t and all controls that have been applied to the robot up to time $t-1$ [2]. The *information space* I is defined as the set of all possibles information states [12]. We underline that the critical events and the robot objective lie over I. That is, a robot objective amounts to achieving a specific task defined on the information state. Two example robot objectives are maintaining visibility of a moving evader and reaching a robot configuration given in terms of a specific robot sensor reading.

Critical events may be of several types. A type of critical event is systematically associated to a type of control. Mainly, to accomplish a robotic task means to answer the following question: what control should be applied on the robot given some i_t?. Thus, planning corresponds to a discrete mapping $\boldsymbol{ce} : i_t \rightarrow u$ between i_t and u, triggered by the critical event $\boldsymbol{ce}$. The output controls correspond to at the very worst case locally optimal polices that solve the robotic task.

Note that instead of using the robot state space, X, we use the critical events to make a decision on control should be applied. $\boldsymbol{ce}$ actually encodes the most relevant information on X. In addition, it relates observations with the best control that can be applied. $\boldsymbol{ce}$ is built using local information but, if necessary, it may involve global one.

We want to use our framework to generate, whenever possible, optimal controls to accomplish a robotic task. As mentioned earlier, planning corresponds to relate a critical events with a control. However, some problems may be history dependent. That means that the performance of a control to be applied not only depends on the current action and a sensor reading, but it also depends on all previous sensor readings and their associated controls. In history dependent problems, the concatenation of locally optimal controls triggered by independent critical events does not necessarily generate a globally optimal solution. For instance, we have shown that object finding is history dependent and moreover NP-hard.

To deal with history dependent problems, we have proposed a two layer approach. The high level, *combinatoric* layer attempts to find a "suitable" order of reaching critical events. The low level, *continuous* layer takes an ordering input by the upper one and finds how to best visit the regions defined by critical curves. This decoupling approach makes the problem tractable, but at the expense of missing global optimality. For the combinatorial level, we have proposed to use a *utility function based heuristic*, given as the ratio of a gain over a cost. This utility function is used to drive a greedy algorithm in a reduced search space that is able to explore several steps ahead but without evaluating all possibles combinations.

In no history dependent problems, such as finding a minimal length path in an environment without holes [11], the Bellman's principle of optimality holds and thus the concatenation of locally optimal paths will result in a globally optimal one. The navigation approach presented in [11] is also based on critical events. But, differently to the ones presented in this paper, it is based on closed loop sensor feed-back.

3 Object Finding

We have used critical events to finding time optimal search paths in known environments. In particular, we have searched a known environment for an object whose unknown location is characterized by a known probability density function (pdf).

In this problem, we deal with *continuous sensing* in a continuous space. We assume that the robot is sensing the environment as it moves. A continuous trajectory is said to *cover* [9] a polygon P if each point $p \in P$ is visible from some point along the trajectory. Any trajectory that covers a simple (without holes) polygon must visit each subset of the polygon that is bounded by the aspect graph lines associated to non-convex vertices of the polygon.

We call the area bounded by these aspect graph lines the *corner guard regions.* A continuous trajectory that covers a simple polygon needs to have at least one point inside the region associated to "outlying" non-convex vertices (non-convex vertices in polygon ears), like A and C in Fig. 1 a). Since these points need to be connected with a continuous path, a covering trajectory will cross all the other corner guard regions, like the one associated to vertex B.

Since a continuous trajectory needs to visit all the corner guard regions, it is important to decide in which order they are to be visited. The problem can be abstracted to finding an specific order of visiting nodes in a graph that minimizes the expected value of time to find an object. [6] shows that the discrete version of this problem is NP-hard. For this reason, to generate continuous trajectories we propose an approach with two layers that solve specific parts of the problem. This one is described below (see 3.4).

3.1 Continuous Sensing in the Base Case

The simplest case for a continuous sensing robot is that shown in Fig. 1 b). Then, the robot has to move around a non-convex vertex (corner) to explore the unseen area A'. For now, we assume that this is the only unseen portion of the environment.

As the robot follows any given trajectory S, it will sense new portions of the environment. The rate at which new environment is seen determines the expected value of the time required to find the object along that route. In particular, consider the following definition of expectation for a non-negative random variable [5]:

$$E[T|S] = \int_0^\infty P(T > t)\ dt. \tag{1}$$

3.2 Expected Value of Time Along any Trajectory

In the simple environment shown in Fig. 1 b) the robot's trajectory is expressed as a function in polar coordinates with the origin on the non-convex vertex. We assume that the robot will have a starting position such that its line of sight will only sweep the horizontal edge E_1. As mentioned before, the expected value of the time to find an object depends on the area A' not yet seen by the robot.

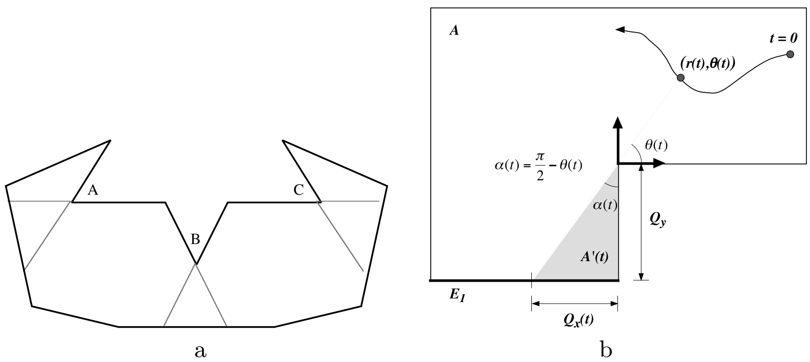


Fig. 1. a) convex corners b) base case

Assuming that the probability density function of the object's location over the environment is constant, the probability of not having seen the object at time t is

$$P(T > t) = \frac{A'(t)}{A} = \frac{{Q_y}^2}{2A\ \tan\left(\theta(t)\right)}, \tag{2}$$

where A is the area of the whole environment (for more details, see [7]) . Note that the reference frame used to define the equation 2 is *local*. It is defined with respect to the reflex vertex (this with interior angle larger than π). From (1) and (2),

$$E[T|S] = \frac{{Q_y}^2}{2A} \int_0^{t_f} \frac{dt}{\tan(\theta(t))}. \tag{3}$$

Equation (3) is useful for calculating the expected value of the time to find an object given a robot trajectory S expressed as a parametric function $\theta(t)$.

3.3 Minimization Using Calculus of Variations

The Calculus of Variations [3] is a mathematical tool employed to find stationary values (usually a minimum or a maximum) of integrals of the form

$$I = \int_a^b F(x, y, y')\, dx, \tag{4}$$

where x and y are the independent and dependent variables respectively.

The integral in (4) has a stationary value if and only if the Euler-Lagrange equation is satisfied,

$$\frac{\partial F}{\partial y} - \frac{d}{dx}\left(\frac{\partial F}{\partial y'}\right) = 0. \tag{5}$$

It is possible to express the differential of time as a function of a differential of θ. This will allow us to rewrite the parametric equation as a function in which θ and r are the independent and dependent variables respectively, The resulting equation is as follows:

$$E[T|S] = \frac{{Q_y}^2}{2A} \int_{\theta_i}^{\theta_f} \frac{1}{\tan(\theta)} \left({r'}^2 + r^2\right)^{\frac{1}{2}}\, d\theta. \tag{6}$$

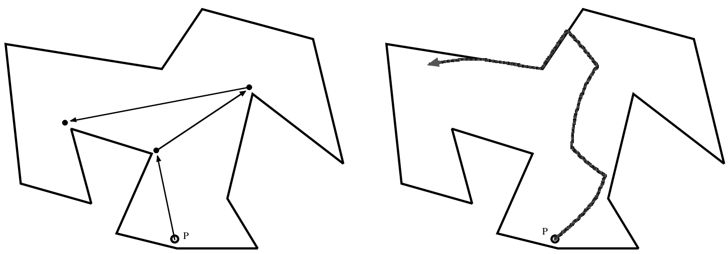
To find stationary values of (6), we use (5) with $x = \theta$, $y = r$ and $F = \frac{1}{\tan\theta}\left({r'}^2 + r^2\right)^{\frac{1}{2}}$. After simplification, this yields the following second order non-linear differential equation,

$$r'' = r + \frac{2{r'}^2}{r} + \frac{2}{\sin(2\theta)}\left(r' + \frac{{r'}^3}{r^2}\right). \tag{7}$$

This equation describes the route to move around a non-convex vertex (corner) to search the area on the other side optimally (according to the expected value of time). We have solved equation (7) numerically using an adaptive step-size Runge-Kutta method. The Runge-Kutta algorithm has been coupled with a globally convergent Newton-Raphson method [7].

3.4 Choosing an Ordering of Regions

To cover a simple polygon, it is sufficient that a trajectory visits at least one point inside each corner guard region (as defined in section 3) associated to reflex vertices of the polygon. The high level, combinatoric layer attempts to find an ordering for the robot to visit these corner guard regions such that the expected value of the time to find an object in the environment is reduced. Note that the discretization defined with critical events is needed because the form of the integral that define the expected value of the time may change according to the shape of the region. To find a suitable ordering, we have defined a point guard inside each corner guard region and used the approach of [6] for sensing at specific locations. The algorithm yields an ordering for visiting corner guard regions (associated to non-convex vertices) that attempts to reduce the expected value of the time to find an object.



a) Expected time 136.9 b) Expected time 115.3

Fig. 2. a) concatenation of straight line paths b) concatenation of locally optimal paths

Once an ordering has been established, the lower level, continuous layer uses the sequence of non-convex vertices to perform locally optimal motions around each of them, thus generating a complete trajectory that covers the polygonal environment. We know that any trajectory generated in this fashion will not be globally optimal in the general case. The main reason of lacking global optimality is that any partition of the problem into locally optimal portions does not guarantee global optimality (Bellman's principle of optimality does not apply). However, through simulation experiments, we have found that the quality of the routes generated by our algorithm is close to the optimal solutions (more details can be found in [6] and [7]).

Figure 2 shows two routes for exploring the environment. The first one, 2 a), is composed by straight lines. The second one, 2 b), is based on the concatenation of locally optimal path generated through an appropriate ordering of reaching critical events defined with our approach. The path generated with our approach produces a smaller average time to find the object. Note that a zig-zag motion is not necessarily bad because a good trajectory must find a compromise between advancing to the next guard and sensing a larger portion of the environment as soon as possible.

4 Pursuit-Evasion

In this section, we consider the surveillance problem of maintaining visibility at a fixed distance of a mobile evader (the target) using a mobile robot equipped with sensors (the observer).

We address the problem of maintaining visibility of the target in the presence of obstacles. We assume that obstacles produce both motion and visibility constraints. We consider that both the observer and the target have bounded velocity. We assume that the pursuer can react instantaneously to evader motion. The *visibility between the target and the observer is represented as a line segment and it is called the rod (or bar).* This rod is emulating the visual sensor capabilities of the observer. The constant rod length is modeling a fixed sensor range.

This problem has at least two important aspects. The first one is to find an optimal motion for the target to escape and symmetrically to determine the optimal strategy for the observer to always maintain visibility of the evader. The second aspect is to determine the necessary and sufficient conditions for the existence of a solution. In this section, we address the first aspect of the problem. That is, to determine the optimal motion strategy, which corresponds to define how the evader and pursuer should move. We have numerically found which are the optimal controls (velocity vectors) that the target has to apply to escape observer surveillance. We have also found which are the optimal controls that the observer must apply to prevent the escape of the target.

4.1 Geometric Modeling

We have expressed the constraints on the observer dynamics (velocity bounds and kinematics constraints) geometrically, as a function of the geometry of the workspace and the surveillance distance. Our approach consists in partitioning the phase space P and the workspace in non-critical regions separated by critical curves. These critical curves define all possible types of contacts of the rod with the obstacles [8]. These curves bound forbidden rod configurations. These rod configurations are forbidden either because they generate a violation of the visibility constraint (corresponding to a collision of the rod with an obstacle in the environment) or because they require the observer to move with speed greater than its maximum.

In order to avoid a forbidden rod configuration, the pursuer must change the rod configuration to prevent the target to escape. We call this pursuer motion the rotational motion. If the observer has bounded speed then the rotational motion has to be started far enough for any forbidden rod configuration. The pursuer must have enough time to change the rod configuration before the evader brings the rod to a forbidden one. There are *critical events* that tell the pursuer to start changing the rod configuration before it is too late. We have defined an escape point as a point on a critical curve bounding forbidden rod configuration sets (escapable cells), or a point in a region bounding an obstacle. This region is bounding either a reflex vertex or a segment of the polygonal workspace. We use D^* to denote the distance from an escape point such that, if the evader is further than D^* from the escape point, the observer will have sufficient time to react and prevent escape. Thus, it is only when the evader is nearer than D^* to an escape point that the observer must take special care. Thus, the critical events are to D^* distance from the escape points.

4.2 Optimal Target and Observer Motions

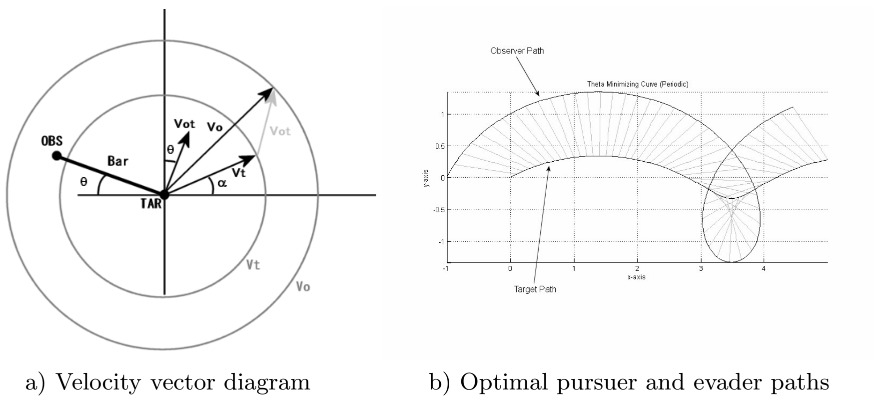
Thus, the optimal control problem is to determine D^*. We solve it using the Pontryagin's minimum principle with free terminal time [1].

Take the global Cartesian axis to be defined such that the origin is the target's initial position, and the x-axis is the line connecting the target's initial position and the escape point. Note that the *reference frame is local.* It is defined with respect to the escape point. The target and observer velocities are saturated at V_t and V_o respectively, and because the rod length must be fixed at all times, the relative velocity V_{ot} must be perpendicular to the rod. This information yields the following velocity vector diagram (see figure 3 a)). θ is the angle between the rod and x axis, α represents the direction of the evader velocity vector used to escape. The rate of change of θ can be found to be [4]:

$$\frac{d\theta}{dt} = \frac{Vot}{L} = \frac{-V_t \sin(\alpha+\theta) \pm \sqrt{{V_o}^2 - {V_t}^2 \cos^2(\alpha+\theta)}}{L} \tag{8}$$

Because the boundary conditions of the geometry are defined in terms of x, a more useful derivative would be:

$$\frac{d\theta}{dx} = \frac{d\theta}{dt}\left(\frac{dx}{dt}\right)^{-1} = \frac{-R\sin(\alpha+\theta) \pm \sqrt{1 - R^2\cos^2(\alpha+\theta)}}{LR\cos(\alpha)} \quad \text{Where} \quad R = \frac{V_t}{V_o} < 1 \tag{9}$$



a) Velocity vector diagram b) Optimal pursuer and evader paths

Fig. 3. Pursuit-evasion

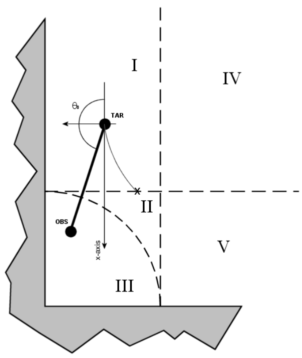


Fig. 4. The evader tries to confine the pursuer in a corner

The optimal path for the target can be defined as follows: Find for a given initial rod angle θ_0, the distance D^* to an escape point x_1 such that the final rod configuration is at a specified final angle θ_1 and the corresponding target motion $\alpha(x),\ x \in [0, x_1]$.

The natural representation would be:

$$\frac{d\theta}{dx} = \frac{-R\sin(\alpha+\theta) \pm \sqrt{1 - R^2\cos^2(\alpha+\theta)}}{LR\cos(\alpha)} = f_1 \quad \frac{dy}{dx} = V_t \tan\alpha = f_2 \quad (10)$$

$$\theta(0) = \theta_0, \theta(x_1) = \theta_1, y(0) = y(1) = 0 \quad (11)$$

To maximize x_1, the appropriate cost function is:

$$V = -\int_0^{x_1} dx \quad (12)$$

The minimum principle establishes:

$$u(*) = argmin\ H(x^*(t), p(t), u, t)$$

$$H = l + p^T f, \quad \dot{p} = -\nabla_x H$$

Where p is the co-state vector, H the system Hamiltonian, l the cost function, f the function state and $u(*)$ the optimal control. The optimal control problem can be stated using 4 conditions:

1. There exists two functions of x, p_1 and p_2 such that $\alpha^* = argmin[p_1 f_1 + p_2 f_2 - 1]$ is satisfied pointwise for all x.
2. The state vector satisfies the state equations and 4 boundary conditions above.
3. The final value x_1 satisfies $p_1(x_1)f_1(x_1) + p_2(x_1)f_2(x_1) - 1 = 0$.
4. The state equations for are given by:

$$\frac{dp_1}{dx} = -\frac{\partial f_1}{\partial \theta} p_1 = \frac{R\cos(\alpha+\theta) \pm \frac{R^2 \cos(\alpha+\theta)\sin(\alpha+\theta)}{\sqrt{1-R^2\cos^2(\alpha+\theta)}}}{LR\cos\alpha} p_1 \tag{13}$$

$$\frac{dp_2}{dx} = -\frac{\partial f_2}{\partial y} p_2 \equiv 0 \rightarrow p_2(x) = p_2 \quad (constant) \tag{14}$$

If we set p_2 to zero, the minimization condition 1 simplifies to

$$\alpha^*(x) = argmin[p_1 f_1 + p_2 f_2 - 1] \rightarrow \frac{\partial}{\partial \alpha}[p_1 f_1 - 1] = 0 \quad \rightarrow \frac{\partial f_1}{\partial \alpha} = 0 \tag{15}$$

In the case when the escape condition (critical event) is defined by a straight line (such as when the evader tries to run the pursuer into a wall, see figure 4), a strategy to generate the solution to the boundary value problem is as follows. Generate the θ-Minimizing Curve by integrating the observer and target positions forward in x, choosing the minimizer α^* at every step Δx. Select two points on the curve and let the line through them represent the new x-axis. The two points can be chosen so that the initial and final angle conditions are satisfied (as measured with respect to the new x-axis), and the optimal path is then the section of the θ-Minimizing Curve connecting the two points. The distance between the two points is x_1, the critical distance D^* to the escape point.

Note that the optimal path for the evader to escape is not a straight line (see figure 3 b). This happens because of the kinematic constraints (bounded speeds and fixed surveillance distance), there is a trade-off between minimizing the time taken for the target to reach the escape point and maximizing the time taken for the observer to change the rod configuration. Therefore, the optimal target path is the one that minimizes the amount of angle that the observer can make in its rotation up until the target reaches the escape point. Figure 4 shows a case when the evader tries to escape pursuer surveillance by confining it against a wall in a concave corner. The critical curves are the dashed lines.

5 Discussion and Future Work

We proposed a framework for reactive motion and sensing planning based on critical events. In this approach planning corresponds to associate critical events with controls. The resulting controls correspond to locally optimal polices for history dependent problems and globally optimal polices for no history dependent ones.

The techniques used to compute optimal paths are open loop methods. However, because of the manner we are using them, a global reference frame is not required. The reference frames are local. For object finding the local reference frame is fixed with respect to the reflex vertices. For pursuit evasion the local reference frame is defined respect to escape points. Therefore, we can compute the controls and resulting paths based on information obtained online directly from the sensors. The required information to compute the controls lies on the information space I.

We underline that there is a similarity between our approach and techniques that have been reported in the literature to compute optimal paths to accomplish robotic tasks. For instance in [10], an approach to compute minimal length paths in the absence of obstacles for non-holonomic robots is presented. The optimal paths correspond to the concatenation of locally optimal ones delimited by critical curves. Those critical curves correspond to the saturation on the admissible robot controls. One important difference is that our robotic tasks are focused on sensing the environment (sensing planning) and we can base our approach on the critical events detected by the sensors.

For future work we want to consider uncertainty in both sensing and control. We believe the use of local reference frames and robot motion planning based on information obtained directly from sensors will result in a robust manner of dealing with uncertainty. We also want to extent our approach to 3D environments. A first effort in the research topic has been already published.

References

1. T. Başar, S.P. Meyn, and W.R. Perkins. *Control system theory and desing.* University of Illinois, 2004.
2. D-P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models.* prentice, 1987.
3. Gelfran. I.M. and S.V. Fomin. *Calculus of Variations.* Prentice Hall, 1963.
4. T. Muppirala, R. Murrieta-Cid, and S. Hutchinson. Optimal motion strategies based on critical events to maintain visibility of a moving target. In *IEEE Int. Conf. on Robotics and Automation*, 2005.
5. S.M. Ross. *Introduction to Probability and Statistics for Engineers and Scientists.* Wiley, 1987.
6. A. Sarmiento, R. Murrieta-Cid, and S. Hutchinson. An efficient strategy for rapidly finding an object in a polygonal world. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2003.
7. A. Sarmiento, R. Murrieta-Cid, and S. Hutchinson. Planning expected-time optimal paths for searching known environments. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2004.
8. J.-T. Schwartz and M. Sharir. On the piano movers problem: I. the case of a two-dimensional rigid polygon body moving amidst polygonal barriers. *Communications on Applied Mathematics*, 36:345–398, 1983.
9. T.C. Shermer. Recent results in art galleries. *Proc. of the IEEE*, 80(9), 1992.
10. P. Souères and J.-P. Laumond. Shortest paths synthesis for a car-like robot. *IEEE Transactions on Automatic Control*, 41(5):672–688, May 1996.
11. B. Tovar, S. LaValle, and R. Murrieta-Cid. Optimal navigation and object finding without geometric maps or localization. In *IEEE Int. Conf. on Robotics and Automation*, 2003.
12. B. Tovar, S. Yershova, J. O'Kane, and S. LaValle. Information spaces for mobile robots. In *Fifth International Workshop on Robot Motion and Control*, 2005.