Realtime Informed Path Sampling for Motion Planning Search

Ross A. Knepper and Matthew T. Mason

Abstract Robot motions typically originate from an uninformed path sampling process such as random or low-dispersion sampling. We demonstrate an alternative approach to path sampling that closes the loop on the expensive collision-testing process. Although all necessary information for collision-testing a path is known to the planner, that information is typically stored in a relatively unavailable form in a costmap. By summarizing the most salient data in a more accessible form, our process delivers a denser sampling of the free space per unit time than open-loop sampling techniques. We obtain this result by probabilistically modeling—in real time and with minimal information—the locations of obstacles, based on collision test results. We demonstrate up to a 780% increase in paths surviving collision test.

1 Introduction

The motion planning problem is to find a path or trajectory that guides the robot from a given start state to a given goal state while obeying constraints and avoiding obstacles. The solution space is high dimensional, so motion planning algorithms typically decompose the problem by searching for a sequence of shorter, local paths, which solve the original motion planning problem when concatenated.

Each local path comprising this concatenated solution must obey motion constraints and avoid obstacles and hazards in the environment. Many alternate local paths may be considered for each component, so planners select a combination of paths that optimizes some objective function. In order to generate such a set of candidate paths, the planner must generate many more candidate paths, each of which must be verified against motion constraints and collision-tested prior to consider-

Ross A. Knepper and Matthew T. Mason
Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA, USA,
e-mail: {rak,mason}@ri.cmu.edu

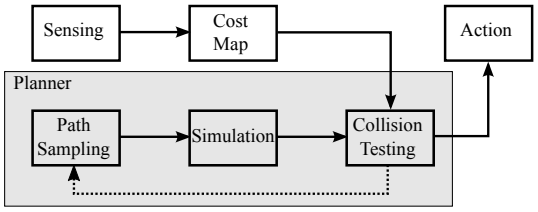


Fig. 1 Typical data flow within a robot closes the loop around the sense-plan-act cycle, but the planner itself runs open-loop. We close the planning loop, informing path sampling with the results of collision-testing earlier paths.

ation. Motion planners generate this large collection of paths by sampling—most often at random or from a low-dispersion sequence.

All the information needed to find collision-free path samples exists within the costmap, but the expensive collision test process prevents that information from being readily available to the planner. A negative collision-test result (i.e. no collision) is retained for future consideration, but a positive collision-test result is typically thrown away because the path is not viable for execution. Such planners may later waste time sampling and testing similar paths that collide with the same obstacle.

This policy of discarding information about colliding paths highlights a major inefficiency, which especially impacts realtime planning. Every detected collision provides a known obstacle location. This observation may not seem significant at first, as all detected collisions represent known obstacles in the costmap. However, not all such obstacles are equally relevant to a given local planning problem, and so we can benefit by storing relevant costmap obstacles in a form more immediately available to the planner. We argue that the planner may derive increased performance by feeding back the set of collision points, known from prior collision tests, to the path sampling process, as in Fig. 1.

1.1 Path Sampling and Path Parametrization

The general path sampling problem is to supply a sequence of distinct paths $\{p_1, p_2, \ldots\} = \mathscr{P} \subset \mathscr{X}$, the continuum space of paths. Often, these paths are not parametrized directly by their geometry but instead are described by their means of generation. For instance, some planners consider only straight-line paths. Given a current robot state $x_0 \in X$, the configuration space, a straight-line path is uniquely specified by connecting x_0 to an arbitrary sampled state $x_f \in X$. In such planners, it is expected that the robot is able to execute arbitrary paths, and so the boundary value problem is easy to solve because it is under-constrained.

Definition 1. Given start and end states, the **boundary value problem** (BVP) is to find any feasible path from the start to the goal (i.e. the local planning problem). A variant of this problem is to find the shortest such path. □

Some classes of robotic systems possess velocity constraints that limit the direction in which they may move instantaneously. The most well known example of

these nonholonomic constraints is the difficulty of parallel parking a car. In such highly constrained, underactuated systems, the set of feasible paths $\mathscr{F}$ is much smaller than the space of all paths, $\mathscr{X}$. Thus, an arbitrary path sample drawn from $\mathscr{X}$ is unlikely to be in the feasible set $\mathscr{F}$. In such cases, the BVP is difficult to solve.

For constrained systems, we may avoid solving the BVP by instead sampling in U, the space of actions. Suppose we have a "black box" model of the robot's response to a control input, which is a mapping $M : U \rightarrow \mathscr{F}$. Sampling in the control space offers several advantages: 1) all sampled paths trivially obey motion constraints; and 2) we may precompute a diverse set of such paths. For a mobile robot, these paths are independent of initial position and heading, depending only on their derivatives (we ignore external interference such as wind or wheel slip). Therefore, a relatively small lookup table suffices to describe an expressive set of robot motions.

1.2 Prior Work

The motion planning community has invested considerable effort in the topic of non-uniform and adaptive sampling. The literature on this topic largely concerns probabilistic roadmaps (PRMs), which sample states rather than paths. Hsu, et al. [11] provide a survey of recent work in non-uniform sampling for PRMs. We touch on a few of the broad approaches here.

One approach employs a fixed strategy to bias configuration sampling towards narrow corridors—parts of the C-space that are less likely to be sampled on their own due to their small measure [1, 9, 17]. These works all restrict the sampling consideration to points, whereas we sample directly in the space of paths, using a distribution that varies in reaction to new collision test results.

The non-uniform sampling field has largely moved towards such adaptive strategies. For instance, Jaillet, et al. [12] restrict sampling to size-varying balls around nodes in an RRT to avoid testing paths that would go through obstacles. Yu and Gupta [19] perform sensor-based planning in which a PRM is incrementally constructed based on the robot's partial observations of obstacles. Exploratory motions are selected by maximizing information gain. Another recent adaptive approach is to construct a meta-planner with several tools at its disposal; such planners employ multiple sampling strategies [10] or multiple randomized roadmap planners [16], based on a prediction of which approach is most effective in a given setting.

One important feature of our work is the use of information from all collision tests, including positive results, to minimize entropy in a model approximating obstacle locations. The work of Burns and Brock [3–6] bears considerable resemblance to ours in this regard. They describe an adaptive model of obstacle locations in C-space based on previous collision test results. Their model utilizes locally weighted learning to select state [4] or path [5] samples that reduce model uncertainty. We likewise develop a model of obstacle location, although ours inhabits the workspace, and its simplicity is better suited to realtime applications. Burns and Brock subsequently observe, as we do, that model refinement is not an end in itself, but merely a

means to the end of finding collision-free paths [6]. We proceed from this observation to consider what level of refinement is appropriate, in the context of constrained paths, based on the maximum width of corridor we are willing to miss discovering.

Separately, Burns and Brock describe an entropy-guided approach to the selection of configuration samples likely to unify two connected components of the PRM graph [3]. In later work [6], they augment this approach with the notion of utility, which combines information gain regarding obstacle locations with the value of a resulting sample for solving the planning problem. Utility-guided sampling selects the configuration expected to solve the eventual planning problem most efficiently. We take a similar approach, in that we sample a combination of paths intended to navigate the space and to refine our obstacle model.

2 Informed Path Sampling Approach

In closing the loop on path sampling, we must feed back knowledge of obstacles reachable by the robot (in the form of collision-test results) into the sample space of paths, be it X or U, so that we can suppress from the sampled path sequence future paths intersecting those regions of the workspace.

A collision can be described as an ordered pair $c = (p,s)$, with $p \in \mathscr{F}$ and $s \in I = [0,1]$, a time/distance parameter describing where on the path the collision occurred. A path is a mapping $p : I \to X$. Thus, c maps directly into a state $x \in X$, identifying the location of an obstacle. However, this collision state is special because it is known to be reachable by an action $u \in U$. In fact, x is probably reachable by a continuum of other actions, of which we can easily precompute a sampled subset for each possible collision point.

Our approach to path sampling feedback in highly constrained systems is to keep the feedback entirely within the action space, by which such paths are parametrized. We first collision test some paths drawn from a low-dispersion sequence [7]. After finding the first collision point, its location biases future action samples. We may construct, a priori, a list of correspondences between possible action samples to accelerate this process.

Often, collision-test routines are able to identify the precise location where a collision occurred. Knowing that workspace point w is part of an obstacle, we may eliminate from our sampled sequence all paths passing through the set $cs(w)$, the set of robot configurations $x_i \in X$ in which the robot intersects the workspace point w. To eliminate these paths, we must store the list of actions by which they are parametrized.

In order to identify the set of paths passing unacceptably close to an obstacle point w, we precompute a proximity look-up table (PLUT), as shown in Fig. 2. Suppose our precomputed path set $\mathscr{P}$ contains N paths, each discretized into M points. The PLUT stores, for every ordered pair of paths (p_i, p_j) in $\mathscr{P}$, the shortest distance to the kth discretized point on p_j:

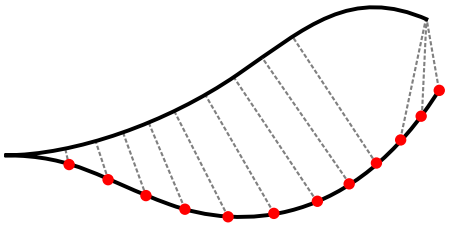


Fig. 2 Given a path set of N paths, each discretized into M points, the proximity look-up table (PLUT) stores for each ordered pair of paths a list of shortest distances to each discrete point on the second path. Thus, there are a total of MN^2 unique PLUT entries.

$$\mathrm{PLUT}(p_i, p_j, k) = \min_{w \in p_i} d\left(w, p_j\left(\frac{k}{M}\right)\right), \tag{1}$$

where $d(w_1, w_2)$ gives the Euclidean distance between two points.

Now, given a collision $c = (p_j, s)$, we would like to find out if another path p_i would collide with the same obstacle. We simply query the PLUT as

$$\mathrm{PLUT}(p_i, p_j, sM) < r_r, \tag{2}$$

where r_r is the radius of the robot (or an inscribed circle of the robot). When this condition holds, the collision test would fail. Knowing this, we may eliminate the path without a test, and spend the CPU time considering other paths.

However, we may go beyond short-circuiting the collision-test to estimating the probability distribution on obstacle locations using the *principle of locality*, which states that points inside an obstacle tend to occur near other points inside an obstacle. We propose a series of models of locality and two path sampling problems, which we address using our models. The key to success of this approach is that the final evaluation be less costly than the collision tests it replaces.

3 Probabilistic Foundations

In this paper, we develop a series of probabilistic models that enable us to rapidly select paths for collision test that maximize one of two properties. First, in order to find valid robot motions, we must sample a selection of collision-free paths for execution. Second, we wish to sample broadly within the free space of paths, including in proximity to obstacles. The precision with which we know the obstacle/free-space boundaries directly relates to the size of narrow corridor we expect to find.

The workspace comprises a set of points divided into two categories: obstacle and free. The function

$$\mathrm{obs}\colon W \to \beta, \tag{3}$$

where $\beta = \{\mathrm{true}, \mathrm{false}\}$, reveals the outcome that a particular workspace point w is either inside (true) or outside of an obstacle. Building on such outcomes, we then describe an event of interest. A path collision-test takes the form

$$\text{pct}\colon \mathscr{P} \to \beta, \tag{4}$$

which returns the disjunction of obs(w) for all w within the swept volume (or *swath*) of the path. A result of true indicates that this path intersects an obstacle.

Using these concepts as a basis, we pose two problems:

1. **Exploitation**. We are given a set of workspace points inside obstacles, $C = \{w_1, \ldots w_m\}$ and a set of untested paths $\mathscr{P}_{unknown} = \{p_1, \ldots p_n\}$. Knowing only a finite subset of the continuum of obstacle-points, find the path that minimizes the probability of collision:

$$p_{next} = \underset{p_i \in \mathscr{P}}{\text{argmin}} \Pr(\text{pct}(p_i, C)). \tag{5}$$

2. **Exploration**. Suppose we have a model of uncertainty $U(\mathscr{P}_{safe}, C)$ over the collision status of a set of untested paths, $\mathscr{P}_{unknown}$, in terms of a set of tested paths and known obstacle-points. Find the path $p_{next} \in \mathscr{P}_{unknown}$ giving the greatest reduction in expected uncertainty:

$$\begin{aligned} U_{exp}(p_i) = {} & U\left(\mathscr{P}_{safe} \cup \{p_i\}, C\right) \Pr\left(\neg \text{pct}(p_i, C)\right) \\ & + U\left(\mathscr{P}_{safe}, C \cup \{c_i\}\right) \Pr\left(\text{pct}(p_i, C)\right) \end{aligned} \tag{6}$$

$$p_{next} = \underset{p_i \in \mathscr{P}_{unknown}}{\text{argmax}} \; U(\mathscr{P}_{safe}, C) - U_{exp}(p_i). \tag{7}$$

These two considerations are essentially the same as those encapsulated in the utility function of Burns and Brock [6] (see Section 1.2).

4 Locality

Thus far, we have demonstrated how a single failed collision test may serve to eliminate an entire set of untested paths from consideration because they pass through the same obstacle point. We may extend this approach one step further using the principle of locality.

Definition 2. The **principle of locality** states that if a robot state is in collision with an obstacle, then that state is contained in a neighborhood ball of obstacle points. □

Two contributing factors combine to produce the locality effect. First, the nonzero volume of the robot means that even a point obstacle results in a set of robot states $cs(t)$ in collision with that point. The second factor is that real-world obstacles occupy some volume in space.

Given a known collision point, we employ the principle of locality to define a function expressing the probability that a new path under test is in collision with the same obstacle. A locality model takes the following general form:

$$\text{loc}(p_i \mid C) = \Pr(\text{pct}(p_i) \mid C) \tag{8}$$

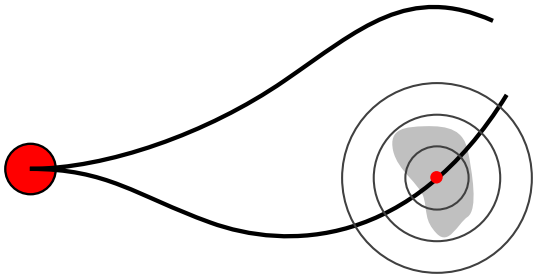


Fig. 3 The robot (red disc at left) considers two paths. The bottom path fails its collision test. The locality model does not know the full extent of the obstacle (gray), but it can approximate the obstacle using a probability distribution (concentric circles) and can estimate the likelihood of the top path colliding.

Here, C may be a single point collision outcome or a set of collisions. If omitted, it is assumed to be the set of all known collisions.

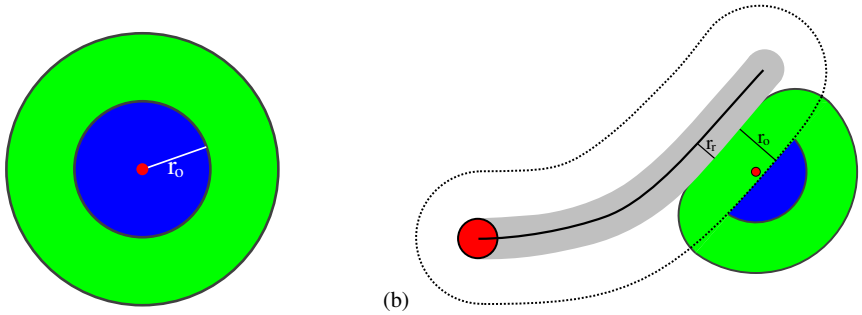
This function depends on many factors, including the size and shape of the robot as well as the distribution on size and shape of obstacles in the environment. The most important parameter, however, is the distance between the new path and the known collision site. Thus, we may establish a rapidly computable first-order locality model in which we abstract away the size and shape of obstacles using a single distribution on radius, as in Fig. 3. We discsuss several intermediate locality model formulations before coming to the final form.

4.1 General Locality Model

By explicitly modeling locality, we may reason about which paths are more or less likely to be in collision with any known obstacle, even with only partial information about its location. A path sampling algorithm, when informed by a locality model, provides a path sequence ordered by likelihood of collision, given currently known collision sites. We propose here a general model of locality that can be expected to produce collision-free path samples with high probability.

In constructing a general locality model, we abstract away many parameters; we consider both the robot and obstacles to be balls (in $\mathbb{R}^2$ or $\mathbb{R}^3$), and the obstacles are assumed uniform in radius. We relax some of these assumptions later, in Section 4.4. For now, these restrictions permit us to simplify the model by removing bearing from consideration. Thus, the general model's prediction of future collisions is purely a function of *range* from the known collision site to the path. The fixed radii of both the obstacles (r_o) and the robot (r_r) result in the intuitive notion of locality: that its influence is over a limited range only.

The precise formulation of the general locality model, as depicted in Fig. 4, is based on maintaining a probability distribution on possible locations of the obstacle centroid, given a known collision site. In this naive model, the location of the centroid is described by a uniform distribution over $B(r_o)$, a ball of radius r_o centered at the colliding position of the robot. A path p_i sweeps out a swath $S(p_i)$ of width



(a) (b)

Fig. 4 The general locality model: **(a)** Given a point known to be in collision with an obstacle (center red dot), the blue inner circle of radius r_o represents possible locations of the centroid of the obstacle. The green outer circle comprises points possibly occupied by some part of the obstacle. **(b)** The probability that a new candidate robot path is collision-free equals the fraction of possible obstacle centroids outside a swath of width $2r_o + 2r_r$. The blue region represents the set of possible centroids, while the green region depicts possible obstacle extents.

$2r_o + 2r_r$. Any non-empty set $B(r_o) \cap S(p_i)$ represents some probability of collision. This general model then predicts that the probability of collision is

$$\mathrm{loc}_{general}(p_i \mid c) = \frac{|B(r_o) \cap S(p_i)|}{|B(r_o)|}. \tag{9}$$

If we regard p_i as a straight line, then in 2D the probability of collision is the ratio of the area of a circular segment to the area of the whole circle, which is [2]:

$$f_{segment}(r) = \begin{cases} \frac{1}{\pi r_e^2}\left(r_e^2 \cos^{-1}\frac{r-r_e}{r_e} - (r-r_e)\sqrt{2r_e r - r^2}\right) & \text{if } 0 \le r \le 2r_e \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where r is the range between the path and the collision point. We call r_e the range of effect, which we set equal to r_o here.

Definition 3. The **range of effect** of a known collision point describes the radius around that point at which paths are regarded to be at risk of collision with the known obstacle. □

4.2 Simple Locality Model

We now propose an even simpler locality model, which closely approximates (10) but makes use of the existing PLUT. Instead of path area, we consider only the point on the new path most closely approaching the known collision point. This new locality model employs the raised cosine distribution:

$$f_{rcd}(r) = \begin{cases} \frac{1}{2r_e}\left[1+\cos\left(\pi\frac{r}{2r_e}\right)\right] & \text{if } 0 \le r \le 2r_e \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

For a stright or gently curving path, this approximation is very good. Then, the probability that a new path p_i will collide with the same obstacle represented by the previous collision $c = (p_j, s)$ is simply

$$\text{loc}_{simple}(p_i \mid c) = f_{rcd}(\text{PLUT}(p_i, p_j, sM) - r_r). \tag{12}$$

Note that here we are no longer maintaining an explicit probability distribution on the location of an obstacle but instead a heuristic estimate of the risk of a single path relative to a single collision site.

4.3 Handling Multiple Collision Sites

Given a known collision site, both (9) and (12) provide a tool for selecting a candidate path to minimize the probability of collision. However, we have not yet addressed the issue of multiple known collision sites. The likelihood that two workspace points have the same obstacle outcome correlates strongly with the distance between them, by virtue of describing the same obstacle. The estimate of collision likelihood for an untested path depends on what statistical independence assumptions we make among known collision points.

Fig. 5 depicts a situation in which two collision sites appear to be correlated. However, many possible policies for estimating statistical independence among a set of collision points, such as clustering, are complex to compute and implement.

In contrast, we may conservatively assume that all collisions are independent, in which case basic probability theory states that

$$\text{loc}(p) = \text{loc}(p \mid \{c_1, \ldots, c_n\}) = 1 - \prod_{i \in \{1,\ldots,n\}} (1 - \text{loc}(p \mid c_i)). \tag{13}$$

If some collision sites are actually part of the same obstacle, then we are overestimating the likelihood of collision for p. In the absence of any knowledge regarding correlation, however, the most conservative policy is the safest. Thus, we now have the means to address Problem 1, Exploitation:

$$p_{next} = \underset{p_i \in \mathscr{P}}{\text{argmin}}\ \text{loc}(p_i). \tag{14}$$

In the next section, we explore an information theoretic approach to safely adjusting this pessimistic model.

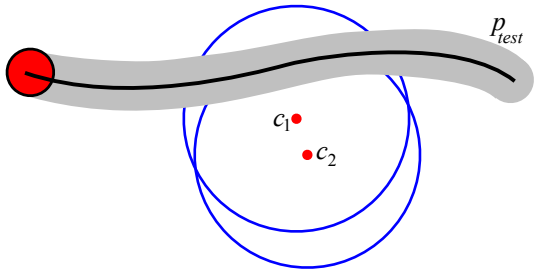


Fig. 5 Two collision sites c_1 and c_2 are located in close proximity. Intuition suggests that c_2 should be ignored when computing the risk of collision of path p_{test}. Either the two sites belong to the same obstacle, or else the obstacle at c_2 is "blocked" by the obstacle at c_1.

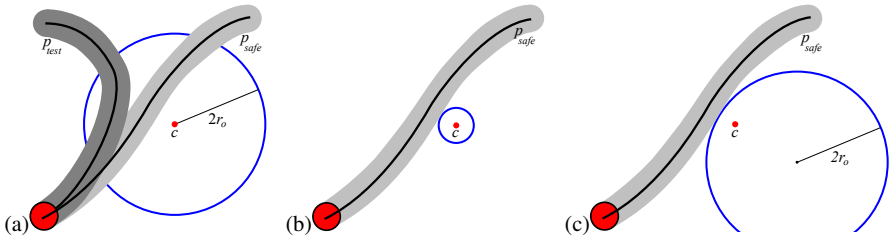


Fig. 6 **(a)** Given a collision point c and neighboring collision-free path p_{safe}, the blue circle represents a distribution on obstacle locations, some of which are invalidated by p_{safe}. The more distant candidate path p_{test} is not at risk of collision with the obstacle represented by c.
(b) and **(c)** Two simple hypotheses on obstacle scale and position explain these two results. The distribution shown in Fig. 4(b) is simpler to represent during online path sampling.

4.4 Adaptive Locality Model

The locality models presented in Sections 4.1–4.2 incorporate only positive collision test outcomes. Those static models conservatively estimate an obstacle distribution spread over a large but finite range of effect. We now construct an *adaptive* locality model capable of incorporating both positive and negative collision-test outcomes.

If we should happen to discover a safe path p_{safe} passing within collision site c's range of effect, then we may use this new information to refine the obstacle model of c. In effect, we adjust the locality function to act over a smaller range in the direction of p_{safe}. As Fig. 6a shows, no path p_{test} that is separated from c by p_{safe} can possibly be at risk of collision with this obstacle. This adaptive model effectively relaxes the earlier, rigid assumptions on obstacle size and independence of collision sites. In modeling such geometric relations, we depart from prior work addressing locality.

Following an update to the model, all future probability estimates involving c incorporate this new information. Although the independence assumption may initially make nearby paths like p_{test} appear riskier than they should (Fig. 5), the adaptive model rapidly cancels out this effect after finding a safe path to shrink each collision point's range of effect.

In addressing the problem of how to adaptively adjust obstacle distributions in reaction to a collision-free path, a variety of approaches present themselves. One

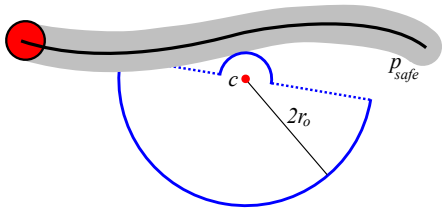


Fig. 7 The range of effect on each side of collision site c is maintained separately. The left range began at $2r_o$, but it was reduced after successfully collision-testing path p_{safe}.

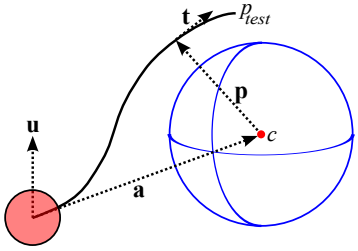


Fig. 8 In three dimensions, the adaptive locality model's range of effect is split into four sides. The robot's up vector, **u**, and the vector pointing toward the collision point, **a**, are used to define which of four sides the path p_{test} is on. As illustrated, the path is on the top-left side.

possible approach is to shrink the range of effect for the obstacle at c, as in Fig 6b, which supposes that the obstacle is smaller than initially thought. Another approach, to shift the entire distribution away from the safe path as in Fig. 6c, assumes that the obstacle size was correctly estimated, but its position was off.

We adopt a compromise position. We prefer that the collision site remains the center of a distribution in order to keep range checks efficient via look-up table. However, we also prefer to avoid altering the range of effect of the opposing side, about which we have no new data. We therefore split the range of effect into several regions of influence ("sides") centered around each collision site. In 2D, we have left and right sides of the obstacle, as in Fig. 7. In 3D, the division is topologically more arbitrary, although we split the obstacle into four sides.

In splitting the locality model into several directions, we require a rule to consistently associate each path with a particular side of the collision point. The sides are defined relative to the pose of the robot before executing the path. The sides meet at the line **a**, an axis running through the start pose and the collision point. We assign names to the sides describing their position relative to the robot's frame of reference. Sides are determined by

$$left = \text{sgn}(\mathbf{t} \times \mathbf{p} \cdot \mathbf{u}) \tag{15}$$

$$top = \text{sgn}(\mathbf{t} \times \mathbf{p} \cdot \mathbf{a} \times \mathbf{u}), \tag{16}$$

where **u** denotes the robot's up vector, **p** the projection of c onto the path, and **t** the tangent vector of the path at this point, as in Fig. 8. These sides may be precomputed for each path. In 2D, it is particularly convenient to augment the PLUT with a sign indicating on which side of the path each possible collision point lies.

Fig. 9 shows a family of paths on the left side of an obstacle. We deem each path equally likely to collide with the obstacle because they each approach equally near

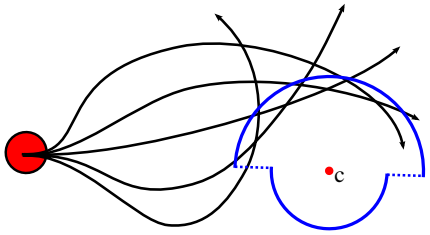


Fig. 9 A family of paths, all of which pass to the left of the collision site, c. Despite the variety of shapes, each path intrudes equally into the left range of effect of c, and thus they would each reduce its left range of effect equally.

to the collision point, c. This assignment of paths to a single side of an obstacle places assumptions on the path's shape. We assume here that curvature is bounded and that paths are reasonably short. Previous work by Knepper, et al. [15] thoroughly discusses these assumptions.

5 Path Entropy

Having established an adaptive locality model, we now consider a means to reap maximal advantage from its predictive capabilities in order to solve Problem 2, Exploration. It is important to select paths for collision test that cause the model to rapidly converge to an accurate description of obstacles, while simultaneously minimizing failed collision tests. Given a set of collision sites, the best path to collision test is that path with maximum entropy according to the current model parameters.

Definition 4. An untested path's **entropy** (sometimes called Shannon entropy) refers to the expected amount of information about the safety of other untested paths that would be gained from collision-testing it. A path's entropy is

$$H(\text{pct}(p)) = -\Pr(\text{pct}(p))\log\Pr(\text{pct}(p)) - \Pr(\neg\text{pct}(p))\log\Pr(\neg\text{pct}(p)).\square \quad (17)$$

In order to maximize our understanding of the true distribution of obstacles with the fewest possible samples, we choose to sample the maximum entropy path:

$$p_{test} = \underset{p_i \in \mathscr{P}}{\text{argmax}}\, H(\text{pct}(p_i)). \quad (18)$$

Based on current information, the maximum entropy path has maximal uncertainty with regard to its collision with obstacles; its probability of collision is nearest to 50%. Testing this particular path will therefore increase total knowledge more than any other. The result will be either a path that significantly reduces the range of effect for some known collision point(s) or a new collision point that is far from known collisions. In either case, model accuracy increases with maximal utility.

The policy of maximizing entropy was proposed by Jaynes [13] for the purpose of estimating an unknown distribution. Maximum entropy has been specifically applied to decision theory [8], as we employ it here. The decision that maximizes

entropy is the one that minimizes the worst case possible outcome. In our case, the worst outcome is rediscovering a known result because it wastes computation time for no gain. This outcome takes two forms: testing a path passing through a known collision site, or retesting a known-safe path. This Γ-minimax approach [18] is capable of reasoning simultaneously about an entire family of probability distributions, called Γ—in our case, a range of theories about obstacle extent.

If the maximum entropy policy is pursued repeatedly, path selection proceeds to discover a sequence of safe paths and collision sites that are progressively nearer to each other, thus establishing precisely the boundaries separating the obstacles from free space. Knowing these boundaries may accelerate the process of sampling and testing paths more densely within the free space. In prior work [15], we provide one possible approach to this process.

Refining these boundaries is a process of diminishing returns, however. As we discover safe paths progressively closer to obstacles, the margin of uncertainty becomes so low that additional maximum entropy path samples provide negligible advantage for a variety of reasons: 1) the cost of collision testing a path often increases with greater obstacle proximity; 2) there is a computational cost associated with path sampling that is proportional to the number of known collision sites; and 3) paths close to obstacles are poor choices for execution. These factors could be incorporated into a utility function [6], but this remains as future work. Thus, we should not exclusively pursue the maximum entropy sampling policy, but also select path samples far from obstacles to maximize safety and path diversity.

We utilize several strategies to combine exploration and exploitation in a hybrid approach. In the absence of any uncertainty from our locality model (such as before the first collision site has been discovered), we sample from a low-dispersion sequence. In the presence of uncertainty, we compute the fraction f of the total allowed plan time that has already elapsed (recall that we assume realtime planning). With probability f we pursue an exploitation (obstacle avoidance) sampling strategy, whereas with probability $1-f$ we instead pursue an exploration (boundary finding) strategy to refine our locality model. Sampled paths very near to known obstacles are set aside without testing for later use, since they make poor candidates for traversal. This threshold distance reflects a willingness to overlook very narrow corridors while other options avail themselves. We search these risky paths if there is time at the end of the plan cycle, after exhausting other paths for test.

6 Experimental Results

We conducted a set of experiments in simulation in order to obtain a sufficient quantity of trials to recognize statistically meaningful trends. Experimental setup is as in our earlier work [14], in which trials comprise sets of one-hundred planning problems; in each one, the curvature-constrained robot attempts to navigate through randomly generated 2D point-obstacle environments, each based on a query comprising start and goal poses. The robot moves continuously while replanning at a fixed rate.

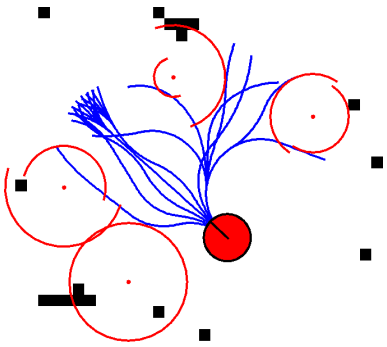


Fig. 10 Simulator depiction of locality model, showing: collision-free paths (blue), known collision sites (red dots), and their ranges of effect (concentric semicircles). Note that the dots correspond to the nearest edge of each C-space obstacle—the point most relevant from the robot's current pose. Not all obstacles (black) are relevant to the current local plan.

A heuristic function selects goal-directed local paths from a set of 2,401 paths. A low-fideltiy global planner helps guide the robot to the goal. In order to assess the effect of various path sampling strategies, we varied replan cycle time. Fig. 10 shows a screen capture of our simulator, indicating known collision sites (in C-space) and ranges of effect. We consider four path sampling strategies:

- Low dispersion—sequence generated by the Green-Kelly algorithm [7]
- Avoid obstacles—pure exploitation; sample as far as possible from obstacles
- Find boundaries—pure exploration; selects maximum entropy path
- Hybrid approach—combines "avoid obstacles" and "find boundaries" strategies.

We present results on three of the strategies in Fig. 11. We see an increase in safe paths produced per unit time for both pure obstacle avoidance (up to 7.8x) and the hybrid approach (up to 3x), compared to a fixed low dispersion sequence. However, pure obstacle avoidance sampling suffers a drop in performance at solving planning queries, which the hybrid approach overcomes. Note that the low-dispersion sampler actually declines slightly in performance as more samples are allowed, which is consistent with earlier results [14]. The hybrid planner shows a trend of increasing performance as it improves its locality model.

7 Discussion

In real time planning, performance is sensitive to the computational cost associated with collision testing. Alternatives that alleviate some of that computation can be beneficial, provided that such alternatives are computationally efficient themselves. In this paper, we present a strategy for informed path sampling that guides the search away from obstacles and towards safe or unexplored parts of the workspace. Although obstacle information is already available to the planner in costmap form, we obtain a significant increase in performance by representing the most salient subset of those obstacles in a more immediately accessible form.

We utilize a proximity look-up table to accelerate this process. Even so, our statistical model describing nearby obstacles and their relationship is necessarily sim-

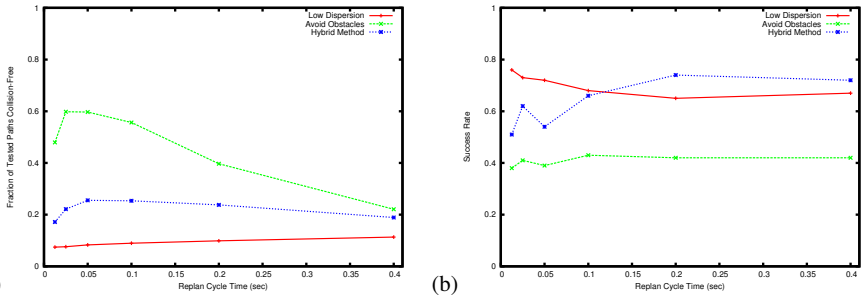


Fig. 11 All tests in these plots were run at an obstacle density of 2%. **(a)** The locality model provides up to a 7.8x increase in the fraction of paths collision-free per replan cycle. As replan cycle time increases, we see regression toward the mean, as a larger total fraction of available paths are collision tested. **(b)** In success rate at solving planning queries, we see that the "avoid obstacles" strategy suffers in performance, whereas the hybrid approach, which strikes a balance between exploration and exploitation of the locality model, performs increasingly well as it has more time to gather information.

ple. This model makes use of the principle of locality to search appropriately far from obstacle locations already discovered in prior collision tests to maximally reduce uncertainty. Using our probabilistic locality model, we trade off between exploration and exploitation in order to discover a variety of safe paths while largely avoiding searching colliding paths.

Acknowledgment

This work is sponsored by the Defense Advanced Research Projects Agency. This work does not necessarily reflect the position or the policy of the Government. No official endorsement should be inferred.

References

[1] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Workshop on the Algorithmic Foundations of Robotics*, pages 155–168, Houston, TX, USA, March 1998.

[2] W. H. Beyer, editor. *CRC Standard Mathematical Tables and Formulae*. CRC Press, Boca Raton, FL, 1991.

[3] B. Burns and O. Brock. Information theoretic construction of probabilistic roadmaps. In *Intl. Conf. on Intelligent Robots and Systems*, Las Vegas, NV, USA, October 2003.

[4] B. Burns and O. Brock. Sampling-based motion planning using predictive models. In *Intl. Conf. on Robotics and Automation*, Barcelona, Spain, April 2005.

[5] B. Burns and O. Brock. Single-query entropy-guided path planning. In *Intl. Conf. on Robotics and Automation*, Barcelona, Spain, April 2005.

[6] B. Burns and O. Brock. Toward optimal configuration space sampling. In *Robotics: Science and Systems*, Cambridge, MA, USA, June 2005.

[7] C. Green and A. Kelly. Toward optimal sampling in the space of paths. In *Intl. Symposium of Robotics Research*, Hiroshima, Japan, November 2007.

[8] P. D. Grünwald and A. P. Dawid. Game theory, maximum entropy, minimum discrepancy and robust bayesian decision theory. *The Annals of Statistics*, 32 (4), 2004.

[9] D. Hsu, T. Jiang, J. Reif, and Zheng Sun. The bridge test for sampling narrow passages with probabilistic roadmap planners. In *Intematiooal Conf. on Robotics and Automation*, Taipei, Taiwnn, September 2003.

[10] D. Hsu, G. Sánchez-Ante, and Z. Sun. Hybrid PRM sampling with a cost-sensitive adaptive strategy. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 3885–3891, 2005.

[11] D. Hsu, J.C. Latombe, and H. Kurniawati. On the probabilistic foundations of probabilistic roadmap planning. *Intl. Journal of Robotics Research*, 25(7): 627–643, 2006.

[12] L. Jaillet, A. Yershova, S. M. LaValle, and T. Simeon. Adaptive tuning of the sampling domain for dynamic-domain RRTs. In *Intl. Conf. on Intelligent Robots and Systems*, 2005.

[13] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, May 1957.

[14] R. A. Knepper and M. T. Mason. Empirical sampling of path sets for local area motion planning. In *Intl. Symposium of Experimental Robotics*, Athens, Greece, July 2008.

[15] R. A. Knepper, S. S. Srinivasa, and M. T. Mason. An equivalence relation for local path sets. In *Workshop on the Algorithmic Foundations of Robotics*, Singapore, December 2010.

[16] M. Morales, L. Tapia, R. Pearce, S. Rodriguez, and N. M. Amato. A machine learning approach for feature-sensitive motion planning. In *Workshop on the Algorithmic Foundations of Robotics*, pages 361–376, Utrecht/Zeist, The Netherlands, July 2004.

[17] A. F. van der Stappen V. Boor, M. H. Overmars. The Gaussian sampling strategy for probabilistic roadmap planners. In *Intl. Conf. on Robotics and Automation*, pages 1018–1023, 1999.

[18] B. Vidakovic. *Γ-minimax: A paradigm for conservative robust Bayesians*, volume 152 of *Lecture Notes in Statistics*, pages 241–259. Springer-Verlag, New York, 2000.

[19] Y. Yu and K Gupta. An information theoretic approach to view point planning for motion planning of eye-in-hand systems. In *Intl. Symposium on Robotics*, pages 306–311, 2000.