International Journal of Robotics Research
():1–14
©The Author(s) 2017
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0278364918778338
www.sagepub.com/
SAGE

Provably Safe Robot Navigation with Obstacle Uncertainty

Brian Axelrod[1], Leslie Pack Kaelbling[2] and Tomás Lozano-Pérez[3]

Abstract

As drones and autonomous cars become more widespread it is becoming increasingly important that robots can operate safely under realistic conditions. The noisy information fed into real systems means that robots must use estimates of the environment to plan navigation. Efficiently guaranteeing that the resulting motion plans are safe under these circumstances has proved difficult. We examine how to guarantee that a trajectory or policy is has at most ϵ collision probability (ϵ-safe) with only imperfect observations of the environment. We examine the implications of various mathematical formalisms of safety and arrive at a mathematical notion of safety of a long-term execution, even when conditioned on observational information. We explore the idea of shadows which generalize the notion of a confidence set to estimated shapes and present a theorem which allows us to understand the relationship between shadows and their classical statistical equivalents like confidence and credible sets. We present efficient algorithms that use shadows to prove that trajectories or policies are safe with much tighter bounds than in previous work. Notably, the complexity of the environment does not affect our methods ability to evaluate if a trajectory or policy is safe. We then use these safety checking methods to design a safe variant of the RRT planning algorithm.

Keywords

Shadows, Shapes, Dual Cone, Polar Cone, Credible Sets, Confidence Sets, Random Geometry, Robot Safety

1 Introduction

1.1 Motivation

Safe and reliable operation of a robot in a cluttered environment can be difficult to achieve due to noisy and partial observations of the state of both the world and the robot. As autonomous systems leave the factory floor and become more pervasive in the form of drones and self-driving cars, it is becoming increasingly important to understand how to design systems that will not fail under these real-world conditions. While it is important that these systems be safe, it is also important that they not operate so conservatively as to be ineffective. They must have a strong understanding of the risks induced by their actions so they can avoid unnecessary risk and operate efficiently.

While most previous work focuses on robot state uncertainty, this paper focuses on safe navigation when the locations and geometries of these obstacles are uncertain. We focus on two aspects:

1. Quantifying uncertainty in obstacle estimates, and
2. Devising algorithms that find safety “certificates”—easily verifiable proofs that the trajectory or policy is safe.

Quantifying uncertainty in obstacle estimates leads to the notion of a shadow—a generalization of a confidence set to an estimated shape. In addition to proposing a formal construction of shadows, we explore some fundamental properties of shadows. This line of work culminates in a correspondence theorem that is instructive in understanding the construction, existence and optimality of shadows.

We examine two implications of the algorithms developed here. First, the computational complexity of reasoning about uncertainty can be quite low. Second, the mathematics surrounding robot safety can have surprising behavior. We demonstrate how these tools can be used to design a motion planner guaranteed to give only safe plans, and inform the design of more general systems that make decisions under uncertainty.

1.2 Problem Formulation

We consider two settings. In the *offline* setting we have a fixed set of information about the environment and are searching for an open-loop trajectory. In the *online* setting the robot has access to a stream of observations and can change its trajectory as a function of new information; the problem is to find a policy, a function from observations to actions, that allows the robot to adapt to changing circumstances. We show that different notions of safety are required for the two cases to ensure that the robot can guarantee a low probability of collision throughout its entire execution.

[1] Stanford University
[2] Massachusetts Institute of Technology
[3] Massachusetts Institute of Technology

Corresponding author:
Brian Axelrod, Stanford University Gates Computer Science Building, Serra Mall, Stanford, CA, USA.
Email: baxelrod@cs.stanford.edu

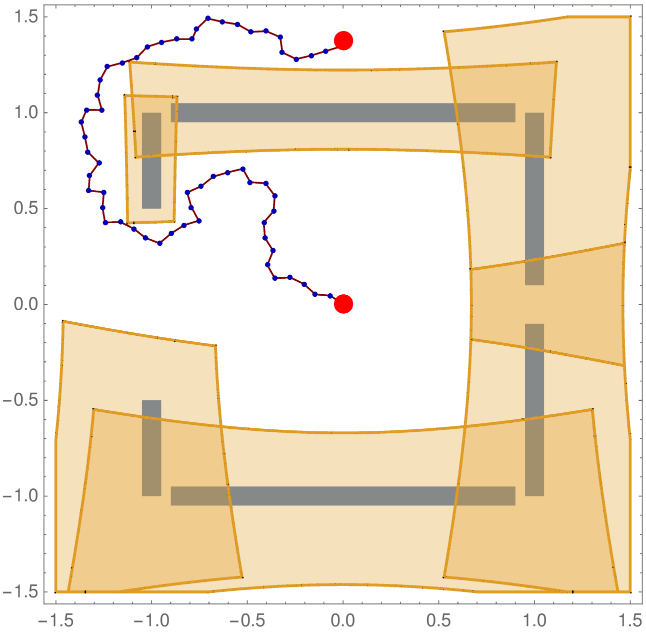


Figure 1. The desired trajectory found by the planner shown with its specialized shadows that certify the probability of collision as less than 0.26%.

Safety in the offline setting amounts to staying out of regions likely to be occupied by obstacles, and can be analyzed by computing geometric bounds on obstacles for which we have only partial information. Safety in the online setting builds on offline safety by requiring that the robot respect a contract with respect to the aggregate lifetime risk of operation while always having a guaranteed safe trajectory available to it.

We analyze safety through the use of "shadows"–a notion of a confidence set adapted to inference over geometry and provide an example of applying this framework to a specific model. We wish to emphasize that this framework can be applied to a wide variety of models beyond the example shown here.

We develop a framework for inference and computation over polytopes in $\mathbb{R}^n$. In this paper we focus on examples of typical robotic domains in $\mathbb{R}^2$ and $\mathbb{R}^3$, but we note that the math carries over directly to higher dimensional spaces that can incorporate information such as the velocity of obstacles. We say that a trajectory is safe if its swept volume is unlikely to collide with any obstacles. These swept volumes can be computed geometrically, or, for dynamical systems, via sum-of-squares programs (Majumdar et al. 2012).

We say that a trajectory, a map from time to robot configurations $\mathcal{Q}$, $\tau : [0, \infty) \to \mathcal{Q}$, is $\epsilon-$safe if the swept volume of the robot along trajectory τ intersects an obstacle with probability less than ϵ. Formally, if A is the event that the swept volume of τ intersects any obstacle, then τ is $\epsilon-$*safe* in the offline sense if $P(A_\tau) \leq \epsilon$.

We say that a policy, π, a map from observation history O, state history H and time to a trajectory τ,

$$\pi : O \times H \times [0, \infty) \to \tau$$

is $\epsilon-$safe if, under all sequences of observations O, $P(A_\pi, O) \leq \epsilon$. This notion of safety will be referred to as *policy safety*; it is a departure from previous models of robot safety, capturing the notion of a contract that the total risk over the lifetime of the system always be less than ϵ.

The requirement that the safety condition hold under all observations sequences is strictly more conservative than requiring safety on average (simply requiring the failure probability to be low on average). It is crucial to prevent undesirable behavior that can "cheat" the definition of safety; this is discussed in detail in Section 4.

1.3 Related Work

Planning under uncertainty has been studied extensively. Some approaches operate in generality and compute complete policies (Kaelbling et al. 1998) while others operate online, computing a plausible open-loop plan and updating it as more information is acquired (Platt et al. 2010; Erez and Smart 2010; Du Toit and Burdick 2010; Hadfield-Menell et al. 2015).

Generating plans that provide formal non-collision safety guarantees when the environment is uncertain has proven difficult. Many methods use heuristic strategies to try to ensure that the plans they generate are unlikely to result in a collision. One way of ensuring that a trajectory is safe is simply staying sufficiently far away from obstacles. If the robot's pose is uncertain this can be achieved by planning with a robot whose shape is grown by an uncertainty bound (Bry and Roy 2011). Alternatively, if the obstacle geometry is uncertain, the area around estimated obstacles can be expanded into a shadow whose size depends on the magnitude of the uncertainty (Kaelbling and Lozano-Pérez 2013; Lee et al. 2013).

Another approach focuses on evaluating the probability that a trajectory will collide. Monte-Carlo methods can evaluate the probability of collision by sampling, but can be computationally expensive when the likelihood of failure is very small (Janson et al. 2015). When the uncertainty is restricted to Gaussian uncertainty on the robot's pose, probabilistic collision checking can yield notable performance improvements (Sun et al. 2016; Park et al. 2016b,a).

Another perspective is finding a plan that is safe by construction. If the system is modeled as a Markov Decision Process, formal verification methods can be used to construct a plan that is guaranteed to be safe (Ding et al. 2013; Feyzabadi and Carpin 2016). Recent work on methods that are based on signal temporal logic (STL) model have also uncertainty in obstacle geometry. With PrSTL Sadigh and Kapoor (2016) explicitly model uncertainty in the environment to help generate safe plans but offer weaker guarantees than our work.

1.4 Contributions

This paper makes four main contributions. The first is a formal definition of online safety that provides risk bounds on the entire execution of a policy.

The second is the development of a theory of shadows. It formalizes a notion of confidence intervals for estimating shapes and relates them to the standard statistical concepts of confidence sets and credible sets. Finally it provides an impossibility result that helps us understand when interesting shadows *do not* exist. We also introduce a model of random shapes and how to compute shadows under said model.

The third contribution is an algorithm for efficiently verifying offline safety with respect to polytopes with Gaussian distributed faces (PGDFs) that is then generalized to the online case. In comparison with previous methods, the quality of the resulting bound is not dependent on the number of obstacles in the environment. The presented algorithms produce a certificate, which allows another system to efficiently verify that the actions about to be taken are safe. For a maximal collision probability of ϵ, the runtime of the algorithm grows as $\log \frac{1}{\epsilon}$ making it efficient even for very small ϵ's.

The fourth contribution is a modification to the RRT algorithm that generates safe plans. For any fixed ϵ, the resulting planner is guaranteed to only return trajectories for which the probability of failure is less than ϵ. We note that for n obstacles, the runtime of the RRT is increased only by a $\log \frac{n}{\epsilon}$ factor, which suggests that reasoning about uncertainty can come at a low computational cost. A result of running this algorithm is shown in Figure 1.

2 Shadows

In order to be able to provide safety guarantees for robot operation in domains with uncertainty about obstacles, we must develop a formal understanding of the relationship between randomness in the observations and resulting estimate of the volume occupied by obstacles. Consider the following scenario: the robot must avoid an obstacle but gets only a noisy observation of said obstacle. Given this noisy observation, it computes an estimate of the space occupied by the obstacle and avoids this region. Conditioned on the observation, however, the true space occupied by the obstacle is random. In other words the true space occupied by the obstacle is not guaranteed to be inside the estimated region. It is not sufficient for the robot to avoid the estimated region to ensure non-collision.

In order to provide theoretical guarantees about a robot's operation in such a domain we must develop mathematics regarding the *random shapes* that come from estimating geometry with noisy observations. The ultimate aim of this section is to develop *shadows*, a geometric equivalent of confidence intervals for uncertain obstacles. Shadows will prove useful in the development of provably correct algorithms for the safe robot navigation problem.

2.1 Inference over Geometry

To provide such safety guarantees we must formalize the setting in which we wish to prove them. This section assumes operation in the offline setting with a constant information set.

Throughout this work we assume that the robot's execution and environment follow the following model:

1. A set of obstacles is fixed in the environment. The obstacles remain static after this point.
2. The robot receives a set of noisy observations of the obstacles.
3. The robot computes and executes a trajectory as a function of its observations.

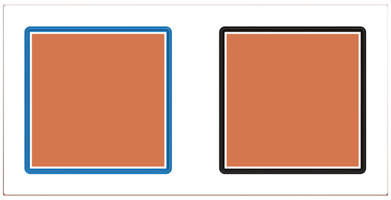


Figure 2. A fair coin is flipped. If the coin is heads the obstacle is placed in the left position, otherwise in the right. The orange region identifies the points with probability at least 0.5 of being in the obstacle. Both outlines represent a valid $0.5-$shadow. The shadow is sufficient to show that a trajectory that avoids the shadow has probability less than 0.5 of colliding with the obstacle. Merely knowing that the orange region is the set of points with at least 0.5 probability of being in the obstacle is not sufficient to provide any guarantees about the safety of a trajectory. We also note that shadows need not be unique—two 0.5-shadows are shown.

We say that a trajectory is safe if the probability of colliding in step 3 is smaller than some fixed ϵ.

In particular, we examine the case where step 3 begins with an attempt to recover the true geometry from the noisy estimates, using a geometric estimator.

Definition 1. Geometric Estimator. *For an observation X, a geometric estimator is a function $X \to \mathcal{P}(\mathbb{R}^n)$ that maps observations to a subset of $\mathbb{R}^n$ that corresponds to the estimated shape.*

Note that if we are using Bayesian decision theory, we can examine the conditional distribution of obstacles given the observations. This notion of a geometric estimator allows us to define shadows for a corresponding notion of confidence sets in frequentist statistics and credible sets in Bayesian statistics.

Definition 2. $\epsilon-$shadow. *A set $S \subseteq \mathbb{R}^n$ is an $\epsilon-$shadow of a potentially random obstacle O if $P(O \subseteq S) \geq 1 - \epsilon$.*

In other words a shadow is a region that contains the true obstacle with sufficiently high probability. The next section will demonstrate an algorithm that uses shadows to certify that trajectories are safe. It suffices to show that a trajectory avoids an ϵ-shadow to show that the total collision probability is less than ϵ.

It is important to understand the distinction between an ϵ-shadow and points with probability greater than ϵ of being inside the obstacle. This can most clearly be seen in the example illustrated in Figure 2. As we will show later in the paper, the probability calculation for a shadow and the likelihood that a point is inside the obstacle is quite different. This discrepancy allows us to provide stronger guarantees than works such as PrSTL that only consider the likelihood Sadigh and Kapoor (2016).

2.2 Geometry Preliminaries

The next section discusses the relationship between the geometry of probability distributions in the parameter space of shapes and the corresponding behavior of the random shapes. To understand the results presented there, it helps to be familiar with some concepts from convex geometry.

First is the notion of homogeneous coordinates. Take a coordinate $x \in \mathbb{R}^n$ from the robot's workspace. We can represent it as homogeneous coordinate as $x \to< x_1, x_2, ...x_n, 1>$, or more generally:

$$x \to< \lambda x_1, \lambda x_2, ...\lambda x_n, \lambda>$$

We can take a homogeneous coordinate $z \in \mathbb{R}^{n+1}$ and convert it to our standard workspace by normalizing by the last coordinate.

$$z \to< z_1/z_{n+1}, z_2/z_{n+1}, ..., z_n/z_{n+1}>$$

Note that these maps are not defined at $z_{n+1} = 0$ and $\lambda = 0$. The geometry of shadows will be much easier to work with in homogeneous coordinates because of their relationship to polar and dual cones.

First we define the notion of a cone.

Definition 3. Convex Cone. *We say that a set $C \subset \mathbb{R}^n$ is a convex cone if C is convex and closed under positive scaling. In other words, C is a cone if for any $x, y \in C, \lambda_1, \lambda_2 \geq 0$, $\lambda_1 x + \lambda_2 y \in C$.*

Polar and dual cones are defined relative to a set $X \subset \mathbb{R}^n$.

Definition 4. Dual Cone. *The dual cone $C^\star$ of X is defined as follows:*

$$C^\star = \{y \in \mathbb{R}^n \mid y^T x \geq 0 \ \forall x \in X\}.$$

In other words the dual cone of a subset of a vector space X is the set of linear functions that do not contain any point of X. If X is a set of linear functionals, its dual cone can be interpreted as contained in all the half-spaces defined by $\alpha^T x \leq 0$ with x in the set X.

Taking the dual cone of the dual cone of X yields the minimal convex cone containing X. If X was already a convex cone then taking the dual twice yields exactly the original set.

Definition 5. Polar Cone. *The polar cone C° of X is defined as follows:*

$$C^\circ = \{y \in \mathbb{R}^n \mid y^T x \leq 0 \ \forall x \in X\}.$$

An image of the polar and dual cone of a set can be seen in Figure 3. Note that the polar cone is a reflection of the dual cone about the origin.

In order to do concrete computations later in this paper we use the notion of a norm cone and a dual norm.

Definition 6. Norm Cone. *Let $||\cdot||$ be a norm (distance measure) on $\mathbb{R}^n$. The corresponding norm cone is*

$$C = \{(x_1, ...x_n, r) \mid ||x|| \leq r\}.$$

Definition 7. Dual Norm ($||\cdot||_\star$). *The dual norm of $||\cdot||$ is $||x||_\star = \sup_y \{y^T x \mid ||y|| \leq 1\}$.*

2.3 Useful Theorems

We will use several theorems without proof through the remainder of the text. They are stated here for convenience. The proofs may be found in the texbook by Boyd and Vandenberghe (2004).

Theorem 1. *Let $C^\star$ be the dual cone of X, and C° the polar cone of X. Then $C^\star = -C^\circ$.*

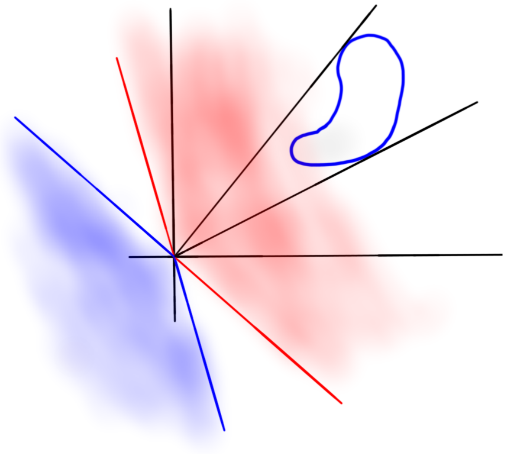


Figure 3. The red cone is the dual cone of the blob outlined in blue. The blue cone is the polar cone of the same region. Note how the polar cone is a reflection of the dual cone. The red and blue cones are in the dual space while the blob outlined in blue is not.

Theorem 2. *Let C be the norm cone corresponding to $||\cdot||$. Then the dual cone, $C^\star$ is the norm cone with norm $||\cdot||_\star$, the dual norm of $||\cdot||$.*

Theorem 3. *Let $||x|| = \sqrt{x^T \Sigma x}$ for $\Sigma \succ 0$. Then $||y||_\star = \sqrt{y^T \Sigma^{-1} y}$.*

2.4 Characterization of Affine Shadows

First we attempt to characterize shadows of half-spaces under a general distribution. This section attempts to answer the following questions:

1. How can one construct a shadow?
2. Do shadows always exist? If so, are they unique?
3. How can one compute the exact probability of collision given a probability distribution over obstacles?

The choice of half-space shadows ends up being crucial. Not only does it correspond to the algorithm we present later in the paper, but there is a natural correspondence between the convex geometry of the homogeneous coordinate system and shadows that allows us to give a complete characterization of shadows in this case.

For the remainder of this section we assume that our random obstacle $\alpha^T x \leq 0$, is parametrized by $\alpha \in \mathbb{R}^n$. Note that α and $k\alpha$ correspond to the same half-space if k is positive. We assume that α is drawn from some known distribution with probability measure μ. For a set X we use the notation $\mu(X)$ to denote $Pr[\alpha \in X]$ under the distribution defined by μ.

We will use different scripts to distinguish between the space of half-space parameterizations, $\mathcal{R}^n$, and the workspace of the robot, $\mathbb{R}^n$.

2.4.1 Probability Computation

First we define a procedure for computing the exact probability that a random half-space $\alpha^T x \leq 0$, parametrized by α, will intersect an arbitrary set $\boldsymbol{A} \subseteq \mathbb{R}^n$. We relate this value to the measure of a dual cone, a notion from convex geometry. More formally, we examine how to find:

$$P(\exists x \in \boldsymbol{A} \text{ such that } \alpha^T x \leq 0).$$

Theorem 4. *$Pr(\exists\, x \in \boldsymbol{A} : \alpha^T x \leq 0) = 1 - \mu(\mathcal{C}^\star)$ where $\mathcal{C}^\star$ is the dual cone of $\boldsymbol{A}$.*

Proof. Recall that the dual cone $\mathcal{C}^\star$ of $\boldsymbol{A}$ is the set $\{\alpha \mid \alpha^T x \geq 0,\ \forall x \in \boldsymbol{A}\}$. First we examine the set of parameters α such that there exists an $x \in \boldsymbol{A}$ such that $\alpha^T x \leq 0$. The complement of this set is the set $\{\alpha \mid \alpha^T x > 0,\ \forall x \in \boldsymbol{A}\}$. Thus

$$Pr(\exists\, x \in \boldsymbol{A} : \alpha^T x \leq 0) = 1 - Pr(\alpha^T x > 0,\ \forall x \in \boldsymbol{A}).$$

Up to the boundary, which is measure zero, $\{\alpha \mid \alpha^T x > 0,\ \forall x \in \boldsymbol{A}\}$ is exactly the dual cone of $\boldsymbol{A}$.

$$\begin{aligned} Pr(\exists\, x \in \boldsymbol{A} : \alpha^T x \leq 0) &= 1 - Pr(\alpha^T x > 0,\ \forall x \in \boldsymbol{A}) \\ &= 1 - \mu(\mathcal{C}^\star) \quad \square \end{aligned}$$

2.4.2 Construction via Polar Cones A related notion in convex geometry, the polar cone, allows us to relate the construction of shadows with a distribution in the halfspace parameter space. Please see Section 2.2 for the related definitions. A correspondence theorem defines a bijection between a elements of set we understand (polar cones) and elements of a set we are trying to characterize (shadows). This will allow us to use our knowledge of the existence and uniqueness of polar cones to understand the existence and uniqueness of shadows. Following the theorems we will provide a statistical interpretation in terms of frequentist confidence set and bayesian credible sets.

The proof of the correspondence theorem (theorem 6) will highlight an important nuance of our current definitions; an ϵ shadow contains the shape with probability *at least* $1-\epsilon$ instead of *exactly* probability $1-\epsilon$. For example, a $0.25-$shadow is also a $0.5-$shadow. This ambiguity will make it difficult to construct a one-to-one map between all shadows and another object which we understand.

We resolve this ambiguity by restricting our attention to maximal shadows.

Definition 8. Maximal ϵ-shadow. *An $\epsilon-$shadow $\boldsymbol{A} \subseteq \mathbb{R}^n$ of a random shape O is maximal if $P(O \subseteq \boldsymbol{A}) = 1-\epsilon$.*

This leads to our first, and most general, method of constructing shadows.

Theorem 5. *For every set $\mathcal{Y} \subseteq \mathcal{R}^n$ of measure $1-\epsilon$, the polar cone C° of $\mathcal{Y}$ is a $\epsilon-$shadow for the random shape defined by the same measure.*

Proof. Recall that a point x is in collision with the halfspace defined by α if $\alpha^T x \leq 0$.

Consider a set $\mathcal{Y}$ of measure $1-\epsilon$. Since sets with empty interiors have zero measure, we can assume without loss of generality that $\mathcal{Y}$ is open.

First we identify the set of points not inside any halfspace defined by a point in $\mathcal{Y}$:

$$\boldsymbol{A} = \{x \mid \alpha^T x > 0,\ \forall \alpha \in \mathcal{Y}\} \subseteq \mathbb{R}^n$$

Up to the boundary, which is a set of measure zero, this is exactly the polar cone, $\boldsymbol{C}^\circ$, of $\mathcal{Y}$.

With probability $1-\epsilon$, a draw of α will be in $\mathcal{Y}$ and thus not correspond to a halfspace that intersects $\boldsymbol{C}^\circ$. This implies that $\boldsymbol{C}^\circ$ is an $\epsilon-$shadow. □

Theorems 4 and 5 can be combined to give a correspondence theorem that will allow us to better understand shadows.

Theorem 6. *There is a one-to-one correspondence between convex cones in parameter space of measure (cumulative probability mass) $1-\epsilon$ and maximal ϵ-shadows.*

Proof. Our proof will show that applying the constructions in theorems 4 and 5 yields the identity.

First we start with a convex cone in the space of half-space parameters, $\mathcal{Y}$, of measure $1-\epsilon$. Theorem 5 tells us that the polar cone of $\mathcal{Y}$ is an $\epsilon-$shadow. The construction in theorem 4 tells us that the probability of the shadow not containing the random halfspace is the measure of the negative of its dual cone.

Since $\mathcal{Y}$ is a convex cone, the negative of the dual cone of the polar cone the original set $\mathcal{Y}$ itself.

The same procedure works for the reverse direction. □

Theorem 6 gives us guidelines about how to construct shadows. It shows if we wish for our $\epsilon-$shadows to be tight, we should construct them by taking the polar cone of convex cones of measure $1-\epsilon$.

It also gives insight as to when $\epsilon-$shadows are not unique. Any set of measure $1-\epsilon$ with a distinct polar cone can be used to create a distinct shadow.

The non-uniqueness gives insight into why not all shadows are equivalent when bounding the probability of intersection. One $\epsilon-$shadow may be sufficient to certify non-collision, but another might not, similar to the situation in Figures 7 and 8. This non-uniqueness has the following important practical implication: ensuring safe, but not conservative, behavior may require searching for the “right” shadow to prove safety.

Finally it helps us answer the question of whether nontrivial shadows always exist. In $\mathcal{R}^n = \mathbb{R}^n$, if the distribution over parameters is such that any halfspace through the origin has measure greater than ϵ then the only shadow is the entire space (the trivial shadow). This comes from the fact that the minimal convex cone with sufficient measure then becomes the entire space. For example, consider constructing an $\epsilon = 0.25$-shadow with $\alpha \sim \mathcal{N}(0, I)$. The distribution is symmetric and all halfspaces through the origin have measure 0.5. Thus we cannot construct any non-trivial $0.25-$shadows for this distribution.

Note that in practice, this means that for a given set of observations, it is not always possible to find a trajectory with probability less than ϵ of collision. One can have insufficient information to certify any trajectory as safe, no matter how conservative.

This suggests a procedure by which we can find the maximal ϵ such that no ϵ-shadow smaller than the full space cannot exist.

Theorem 7. *Let*

$$\epsilon^\star = \inf_{\alpha \in R^n} \mu(\alpha^T x \leq 0)$$

*Then for all $\epsilon' < \epsilon^\star$, there do not exist **any** $\epsilon'-$shadows.*

Since any convex cone whose interior contains a halfspace through the origin must be the entire vector space, and no halfspace has measure more than $\epsilon^\star$, for any $\epsilon' < \epsilon^\star$ there cannot exist an $\epsilon'-$shadow. In other words there is no shadow that contains the set with probability more than $1-\epsilon$. Any distribution which does not contain all of its measure on one side of the origin must then exhibit this threshold behavior.

2.4.3 Statistical Interpretation Finally theorem 6 has a more statistical interpretation. The polar cone operator maps a confidence set or credible set to a shadow. In other words, our standard machinery to construct confidence sets and credible sets can be used to construct shadows (by simply taking the polar cone of the confidence/credible set). Furthermore, when the confidence set or credible set is a convex cone, nothing is lost in the transition from the classical statistical setting to our setting.

2.5 PGDF Shadows

In the previous section we constructed shadows for halfspace obstacles in full generality. Unfortunately, even the integrals required to determine the probability that a shadow contains an obstacle can be difficult to compute.

In this section we will restrict our attention to a particular model and propose an efficient algorithm to construct shadows. Furthermore, instead of trying to compute a shadow that is tight with respect to its probability, we will construct shadows for which it is easy to compute a lower bound on the probability that a shadow contains a random shape.

2.5.1 Model In order to compute shadows we must specify the distribution from which the obstacle is drawn. One way to arrive at a distribution on the shape and position of obstacles is to imagine that sensor data is obtained in the form of point clouds in which the points are segmented according to the face of the obstacle to which they belong. Then, the points belonging to a particular obstacle face can be used to perform a Bayesian linear regression on the parameters of the plane containing the face; given a Gaussian prior on the face-plane parameters and under reasonable assumptions about the noise-generation process, the posterior on the face-plane parameters will also be Gaussian (Bishop 2006; Rasmussen and Williams 2006).

Note that we use homogeneous coordinates to allow us express faces in the form $\alpha^T x \leq 0$ instead of $\alpha^T x \leq b$. This allows us to give formal definitions:

Definition 9. *A face is said to be Gaussian distributed with parameters μ, Σ if $\alpha^T x \leq 0$ defines the face and $\alpha \sim \mathcal{N}(\mu, \Sigma)$.*

Definition 10. *A random polytope is said to be a Polytope with Gaussian Distributed Faces (PGDF) if it a known number of faces, each of which is Gaussian distributed with known parameters; that is, if it is of the following form:*

$$\bigcap_i \alpha_i^T x \leq 0$$

$$\alpha_i \sim \mathcal{N}(\mu_i, \Sigma_i)$$

2.5.2 Construction In this section we construct a shadow for a given distribution of random shapes as follows. We identify a sufficiently large scaled covariance ellipse around the mean parameter vector such that its measure is $1-\epsilon$. We then take the polar cone of this ellipse and use it as our shadow. A pictorial outline of the derivation is presented in Figure 4.

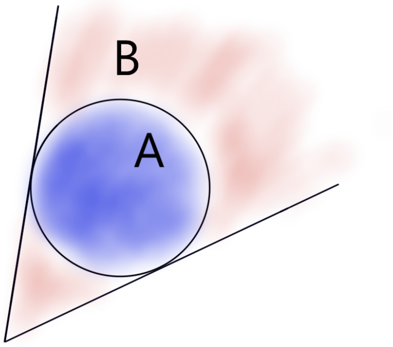


Figure 4. Set B corresponds to parameters of half-spaces that will be occupied our shadow. We provide a lower bound on the measure of set B using the measure of set A, an ellipsoid of measure $1-\epsilon$.

We begin by identifying required size of the covariance ellipse.

Lemma 1. *Let $\alpha \sim \mathcal{N}(\mu, \Sigma)$, ϕ be the CDF of the Chi-Squared distribution with n degrees of freedom and:*

$$\mathcal{X} = \{\beta \mid (\beta-\mu)^T \Sigma^{-1} (\beta-\mu) \leq \phi^{-1}(1-\epsilon))\}.$$

Then $P(\alpha \in \mathcal{X}) = 1-\epsilon$.

Proof. First we note that α is equal in distribution to the following random variable:

$$\Sigma\alpha' + \mu, \alpha' \sim \mathcal{N}(0, I).$$

$\alpha'^T\alpha'$ is then a Chi-Squared random variable with n degrees of freedom. It follows that:

$$P(\alpha'^T\alpha' \leq \phi^{-1}(1-\epsilon)) = 1-\epsilon$$

Define Z as follows:

$$Z = \{\alpha \mid \alpha'^T\alpha' \leq \phi^{-1/2}(1-\epsilon)\}.$$

Then $P(\alpha' \in Z) = 1-\epsilon$. Let Y be the image of Z under the map we used to generate a random variable identical in distribution to α.

Written out explicitly:

$$Y = \{\beta \mid (\beta-\mu)^T \Sigma^{-1} (\beta-\mu) \leq \phi^{-1}(1-\epsilon)\}$$

The measure of Y under the distribution over α must be the same as the measure of Z under the distribution over α'. Thus $P(\alpha \in Y) = 1-\epsilon$. □

Now, given this ellipse of sufficient measure, we can compute its polar cone and resulting $\epsilon-$shadow. We note that if the ellipse given in lemma 1 contains the origin in its interior, the resulting polar cone will be empty since the minimal enclosing convex cone is the complete space (recall that operating homogeneous coordinates means that containing a neighborhood about the origin implies containing the whole space). We compute the polar cone by first computing the minimal cone $\mathcal{C}$ which contains the ellipse, and then computing the polar cone of $\mathcal{C}$.

For the remainder of the section we assume that the space has been rotated and scaled such that $\mu = (0..., 0, 1)$ and the generated ellipses do not contain the origin.

Theorem 8. *For nondegenerate PGDF halfspaces, there exists Σ' such that*

$$\boldsymbol{X} = \{< x_1....x_{n-1}, z >| \; x^T\Sigma' x \geq z\}$$

is an $\epsilon-$shadow.

Our proof is constructive and tells us how to compute Σ'.

Proof. Let $\mathcal{X} = \{\beta \mid (\beta - \mu)^T\Sigma(\beta - \mu) \leq r^2\}$ be the ellipse identified in lemma 1. For the remainder of the proof we work with the surface and not the volume (switching to an equality).

We expand the equation of the above surface for convenience:

$$\begin{aligned} r^2 &= (\beta - \mu)^T\Sigma(\beta - \mu) \\ &= \beta^T\Sigma\beta - 2\beta^T\Sigma\mu + \mu^T\Sigma\mu. \end{aligned}$$

Now we compute the equation for the normals to the surface at point x

$$2\Sigma x - 2\Sigma\mu.$$

Then we identify the set where the normal vectors are orthogonal to the vector to the point x:

$$\begin{aligned} x^T(\Sigma x - \Sigma\mu) &= 0 \\ x^T\Sigma x - x^T\Sigma\mu &= 0 \\ x^T\Sigma x &= x^T\Sigma\mu. \end{aligned}$$

Substituting this into the equation of the original ellipse gives us the equation for the plane that contains the set where the ellipse is tangent to the minimal containing cone:

$$\begin{aligned} \beta^T\Sigma\mu - 2\beta^T\Sigma\mu + \mu^T\Sigma\mu &= r^2 \\ -\beta^T\Sigma\mu + \mu^T\Sigma\mu &= r^2 \\ \mu^T\Sigma\beta &= \mu^T\Sigma\mu - r^2. \end{aligned}$$

This is a linear equation in β which we can use to solve for β_n in terms of the remaining indices of β. Substituting it into the original equation yields the equation of a new ellipse. Let Σ^E, x_0, r' be the parameters of this new ellipse of the following form:

$$(\beta^T_{1...n-1} - x_0)^T\Sigma^E(\beta_{1...n-1} - x_0) \leq r'^2$$

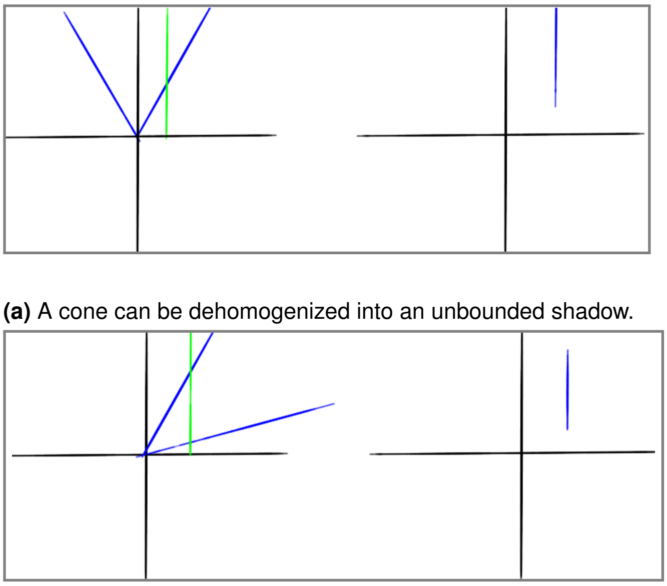
Now we can directly identify the equation of the cone as:

$$\mathcal{C} = \beta^T\begin{pmatrix}\Sigma^E & 0 \\ 0 & -r^2\end{pmatrix}\beta \leq 0;\ \beta_n \geq 0.$$

Alternatively we can describe the set $\mathcal{C}$ as a second order cone:

$$\mathcal{C} = \left\{\beta \mid \sqrt{\beta^T_{1..n-1}\Sigma'\beta_{1..n-1}} \leq \beta_n\right\}$$

$$\Sigma' = \frac{\Sigma^E}{r^2}$$


(a) A cone can be dehomogenized into an unbounded shadow.

(b) Other cones can be dehomogenized into bounded shadows. Note that in order to be safe in this scenario the robot may be forced to stay *within* a bounded, safe "bubble" as opposed to *avoid* a bounded "bad" region.

Figure 5. Taking the dehomogenized version of a set is exactly taking a linear slice (corresponding to $x_n = 1$). When the set we are dehomogenizing is a convex cone this means that the result is a conic section. Depending on the cone we may get different results after dehomogenization. Even in 2 dimensions (illustrated above), these results may differ quite drastically. They can be unbounded (a), bounded (b), and empty.

This identifies our cone as a standard-norm cone with the norm induced by Σ', and makes finding the dual cone a standard problem. The dual cone is just $||\beta_{1...n}||_\star \leq \beta_n$ where $||\cdot||_\star$ denotes the dual norm (see Boyd and Vandenberghe 2004, example 2.25, page 52). The dual norm is the natural one induced by Σ'^{-1}. Thus the dual cone is

$$\beta^T\begin{pmatrix}\Sigma'^{-1} & 0 \\ 0 & 1\end{pmatrix}\beta \leq 0;\ \beta_n \geq 0.$$

Finally since $C^\circ = -C^\star$ we can write down the equation of the polar cone which defines the shadow.

$$\beta^T\begin{pmatrix}\Sigma'^{-1} & 0 \\ 0 & 1\end{pmatrix}\beta \leq 0;\ \beta_n \leq 0 \quad \square$$

When we dehomogenize the coordinate system we get a conic section as a shadow. We refer to Handlin (2013) for a classification of different sections and a discussion of conic sections in high dimensions, but we note that this step implies that the procedure does not always produce non-degenerate shadows, as shown in Figure 5.

Now that we can construct shadows for an individual face we can combine several to create a shadow for a PGDF obstacle.

Theorem 9. *Let $X = \bigcap x^T\alpha_i \leq 0$ with $\alpha_i \sim (\mu_i, \Sigma_i)$. Let S_i be an ϵ_i-shadow for $x^T\alpha_i \leq 0$. Then $S = \bigcap S_i$ is a $\sum \epsilon_i-$shadow for X.*

The proof follows by application of a union bound. Before we continue on to algorithms that use shadows, we further examine the limiting behavior of the PGDF model.

2.5.3 Threshold Behavior Recall theorem (7) that states that unless there exists a halfspace through the origin of measure 1, there exists some $\epsilon^\star$ such that no nontrival $\epsilon-$shadows exist for $\epsilon < \epsilon^\star$.

Normal distributions clearly satisfy this property meaning that PGDF shadows exhibit this threshold behavior. This suggests that future work may wish to examine other noise models. Note that the current noise model suggests that it is possible observe points "behind" the sensor—an unreasonable assumption for most depth sensors currently on the market.

Theorem (7) suggests that the failure to generate a shadow during the dehomogenization step is not simply an artifact of the current approach. While the threshold at which this behavior manifests can be increased, it is fundamentally unavoidable under this noise model since there will always be a threshold below which non-trivial shadows cannot be constructed.

3 Certifying Safety

We will verify that trajectories are safe by finding a set of shadows that proves the swept volume of the trajectory is unlikely to collide with an obstacle. In order to minimize the number of scenarios in which a trajectory is actually safe, but our system fails to certify it as such, we will search for the optimal set of shadows for the given trajectory, allowing shadows for distant obstacles to be larger than those for obstacles near the trajectory. In order to understand the search for shadows of multiple obstacles we first examine the case of a single obstacle and space X potentially visited by the robot (for example compuated as the swept volume of a trajectory or SOS funnel).

3.1 Single Obstacle

For a single obstacle with index i, we want to find the smallest ϵ_i risk bound, or equivalently, largest shadow that contains the estimate but not the volume of space that the robot may visit. That is, we wish to solve the following optimization problem:

$$\begin{aligned} &\underset{\epsilon\in(0,1)}{\text{minimize}} && \epsilon \\ &\text{subject to} && \text{shadow}(\epsilon) \cap X = \varnothing \end{aligned}$$

If we restrict ourselves to the shadows obtained by theorem 8, a shadow with a larger ϵ is strictly contained in a shadow with a smaller ϵ. This implies that the intersection is monotone in ϵ, allowing us to solve the above problem with a line search as shown in Figure 6. While we restrict our attention to the general case, in certain cases, such as where X is a collection of points, this optimization can be solved analytically.

Essentially we are growing the size of the shadow until it almost touches the space that the robot can visit, X.

We define FIND_MAXIMAL_SHADOW$(\epsilon_p, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, V)$, which takes the precision ϵ_p, PGDF parameters $\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$, and swept volume V, and uses a standard bisection search to find and return the largest ϵ for which the shadows are non-intersecting with V. This requires $O(\log 1/\epsilon_p)$ calls of intersection—proportional to the number of digits of precision required. The runtime grows very slowly as the acceptable probability of collision goes to zero.

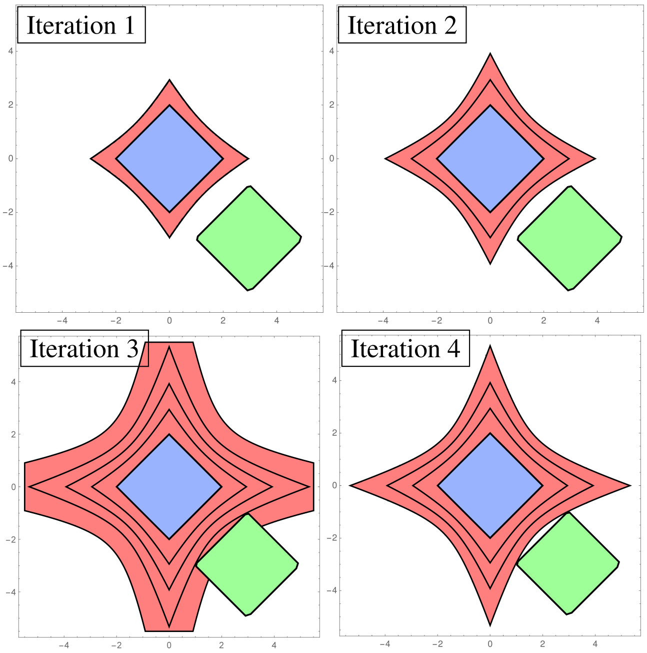


Figure 6. A line search for the maximal shadow. The shadow "grows" and "shrinks" until it contacts the green space visited by the robot.

3.2 Multiple Obstacles

In order to extend the algorithm to multiple obstacles we search for multiple shadows and use a union bound to guarantee safety. Before we present the algorithm we present the lemma underlying its correctness.

Lemma 2. *Consider a robot operating in an environment with n random (uncertain) obstacles. If its trajectory avoids the ϵ_i-shadow of obstacle i, then the robot avoids collision with any obstacle with at least probability* $\sum_{i=0}^{n} \epsilon_i$.

The proof follows by application of the union bound on the probability of colliding with each obstacle.

To compute this probability, we run a line search to determine the largest allowable ϵ for every obstacle, and sum the resulting ϵ's to get the ultimate bound on the risk. The psuedocode is presented in algorithm 1.

Algorithm 1 FIND_MAXIMAL_SHADOW_SET

```
Input: $\epsilon_p, \{\boldsymbol{\mu}_i\}, \{\boldsymbol{\Sigma}_i\}, V$
Output: $\epsilon$, s.t. the path generating volume $V$ is at least $\epsilon$ safe
    and each shadow is less than $\epsilon_p$ away from the minimal
    $\epsilon$ for which this class of bound may be obtained.
for i = 1...n do
    $\epsilon_i$ = FIND_MAXIMAL_SHADOW($\epsilon_p, \{\boldsymbol{\mu}_i\}, \{\boldsymbol{\Sigma}_i\}, V$)
end for
return $\sum \epsilon_i$
```

This algorithm is embarrassingly parallel because every ϵ_i can be computed independently without increasing the total amount of required computational operations. The resulting risks can be added with an adder tree of $O(\log n)$ depth.

To obtain a total accumulated numerical error less than δ we only need to set $\epsilon_p = \delta/n$. If ω is the complexity of a single evaluation of intersection, our serial algorithm runs in $O(\omega n \log n/\delta)$ time. However, since the search for shadows can be done in parallel in a work-efficient manner, the algorithm can be parallelized to run in $O(\omega \log n \log n/\delta)$ time on $\Theta(n)$ processors. Note that there is only a polylogarithmic n computation time penalty for obstacles given sufficient parallelism.

If the intersection check is implemented with a tesselation of the shadow collision checker then finding a safety certificate is only log factors slower than running a collision check–suggesting that systems robust to uncertainty do not necessarily have to have significantly more computational power.

Furthermore, since the algorithm computes a separate ϵ for every obstacle, obstacles with little relevance to the robot's actions do not significantly affect the resulting risk bound. This allows for a much tighter bound than algorithms which allocate the same risk for every obstacle.

3.3 Experiments

We can illustrate the advantages of a geometric approach by certifying a trajectory with a probability of failure very close to zero. For an allowable chance of failure of ϵ, the runtime of sample-based, Monte-Carlo methods tends to depend on $1/\epsilon$ as opposed to $\log 1/\epsilon$. Monte-Carlo based techniques rely on counting failed samples requiring them to run enough simulations to observe many failed samples. This means that they have trouble scaling to situations where ϵ approaches zero and failed samples are very rare. For example, Janson et al.'s method takes seconds to evaluate a simple trajectory with $\epsilon = 0.01$, even with variance reduction techniques (Janson et al. 2015).

We demonstrate our algorithm on a simple domain with $\epsilon = 2.2 \times 10^{-5}$. Our algorithm required just 6 calls to a collision checker for each obstacle. We also demonstrate that our algorithm can certify trajectories which cannot be certified as safe with shadows of equal sizes. Figures 7 and 8 show the problem domain. Figure 7 shows that the trajectory cannot be certified as safe with a uniform risk assigned to each obstacle. Figure 8 shows the shadows found by our algorithm that prove the trajectory is safe.

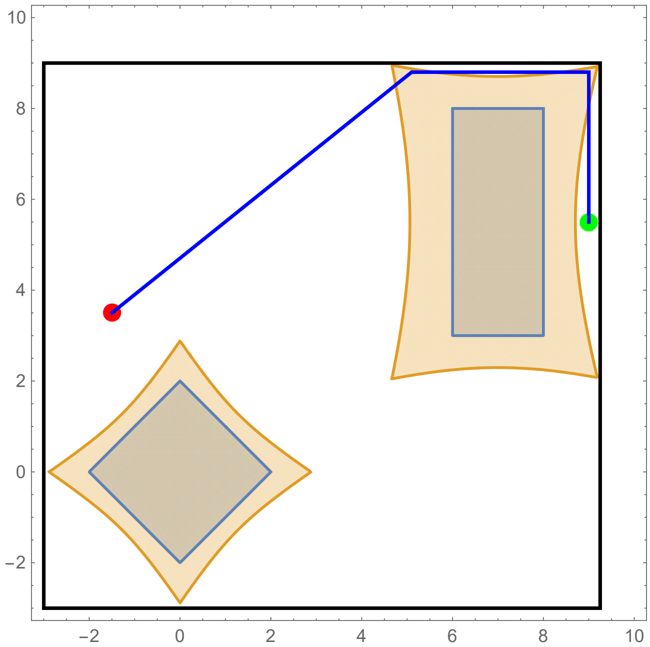


Figure 7. Computing the optimal equal allocation of probabilities fails to certify the safety of the path.

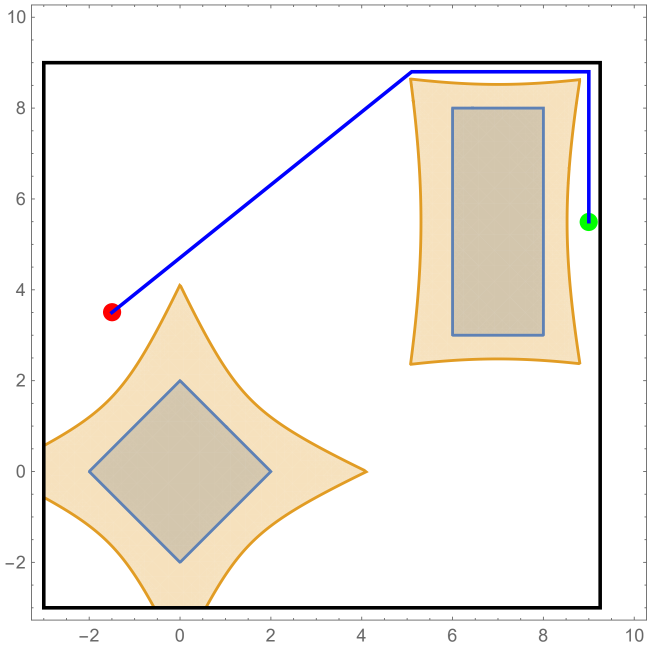


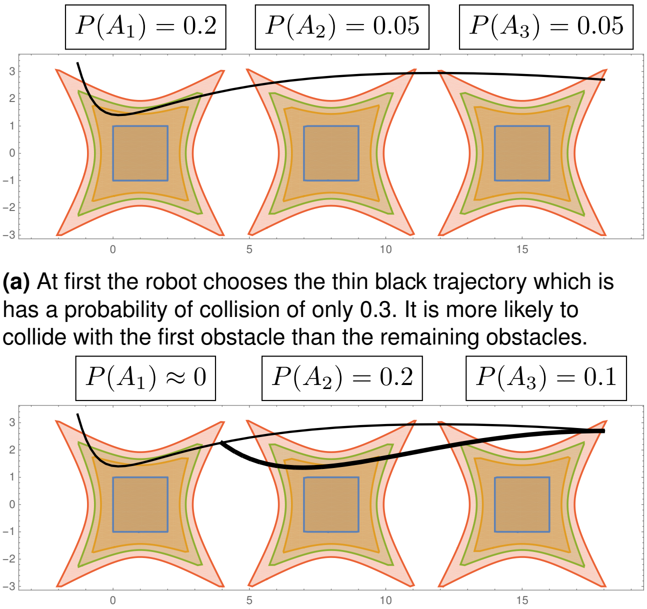
Figure 8. Computing the optimal probability for each shadow allows us to successfully verify that the trajectory is safe.

4 Online Safety

The bounds in the previous section do not immediately generalize to a setting where the robot acquires more information over time and can be allowed to change its desired trajectory. Additional care must be taken to ensure that the system cannot "trick" the notion of safety used, and not honor the desired contract on aggregate lifetime risk of the execution instance. We do not wish to ever allow the system to take actions that are known to be unsafe. Consider the case where, if a fair coin turns up as heads the robot takes a path with a 1.5ϵ probability of failure and it takes a trajectory with a 0.5ϵ probability of failure otherwise. This policy may take an action that is known to be unsafe, but randomization is exploited to ensure that the probability of failure is less than ϵ. We note that while the above example requires a randomized policy, there is no fundamental reason why the policy would need direct access to random bits to achieve this effect. Random bits may instead be extracted from noise in the observations. This allows the policy to take unsafe actions in certain scenarios, even without access to internal randomization.

Furthermore, the history of actions is also important in ensuring aggregate lifetime safety. In Figure 9 we illustrate a example of how the robot can always be committed to some trajectory that is ϵ-safe but have more than an ϵ probability of collision over the lifetime of the execution.

Figure 9 highlights the need to ensure low probability of failure under all sets of observations. If this scenario is run multiple times the failure rate will be much greater than



(a) At first the robot chooses the thin black trajectory which is has a probability of collision of only 0.3. It is more likely to collide with the first obstacle than the remaining obstacles.

(b) After observing that it had not collided with the first obstacle, it readjusts the plan to follow the bold trajectory so that the probability of collision is still less than 0.3. However, a system that follows this policy will collide with probability greater than 0.65 even though at every point it was following a trajectory with probability of collision less than 0.3. In other words, for this set of observations O, $P(A|\pi, O) > 0.3$. A system that is allowed to change its action after seeing additional information (in this case the fact that it did not collide) must properly account for the risk already taken.

Figure 9. Tricking "safety" by changing paths once more information is acquired. Note that even knowing that the system did not collide can serve as information.

acceptable. In order to propose an algorithm that allows the robot to change the desired trajectory as a function of a stream of information, we develop an alternative criterion for safety that accounts for risks as they are about to be incurred. We let p_t denote the probability of collision at time t, given the information available at time t and conditioned on no collision before time t, provided that we follow the trajectory currently predicted by the policy π. We note that since the information itself is random, p_t is a random variable for future times. We say that a policy π is *absolutely safe* if for all times t, equation (1) is satisfied. The expectation in the integral is with respect to the information available at the current time t.

$$\int_0^\infty E[p_t \mid \pi]dt = \int_0^t p_t dt + \int_t^\infty E[p_t \mid \pi]\, dt \le \epsilon \qquad (1)$$

We note that the $\int_0^t p_t dt$ can be evaluated as an accumulation with standard numerical techniques for evaluating integrals.

The second term, $\int_t^\infty E[p_t \mid \pi]\, dt$, is exactly the probability that the remaining part of the trajectory will collide and can be evaluated with the method for solving the offline safety problem.

4.1 Absolute Safety vs Policy Safety

Algorithm 2 provides a method for performing safe online planning in the case that the PGDF parameters are updated during execution. While it shows that absolute safety can be verified efficiently, it is not clear how to efficiently verify policy safety. However, unlike absolute safety, policy safety is a very direct condition on aggregate lifetime probability of collision and can be easier to interpret. In this section we compare policy safety to absolute safety in order to identify when they are equivalent.

First we show that absolute safety is a strictly stronger condition than policy safety in theorem 10. This comes by integrating the probability of failure over time to get the total probability of failure. Since the absolute safety condition in equation (1) must always be satisfied, regardless of the observation set, the probability of failure for that information set will always be sufficiently small.

Theorem 10. *If a policy is absolutely safe, then it is also safe in the policy safety sense.*

Proof. Assume for the sake of contradiction that a policy is absolutely safe, but not policy safe. That means there exists a set of observations O and time t for which policy safety does not hold, but absolute safety does.

The probability of collision conditioned on these observations is:

$$\int_0^t p_t|O dt + \int_t^\infty E[p_t|O]dt$$

Note that once the set of observations is fixed, p_t becomes constant so the above expression is *not* a stochastic integral.

Since the integral evaluates to less than ϵ, so must the probability of collision. However, this implies that the system is policy safe for this set of observations O and time t, yielding a contradiction.

Thus if a policy is absolutely safe it must also be policy safe. □

Absolute safety, however, is not always equivalent to policy safety. The key difference lies in how the two conditions allow future information to be used. Absolute safety requires that the system always designate a safe trajectory under the current information while policy safety allows the robot to postpone specifying a complete, safe trajectory if it is certain it will acquire critical information in the future.

In order to formalize when policy safety and absolute safety are equivalent, we introduce the notion of an information adversary. An information adversary is allowed to (1) see the observations at the same time as the agent, (2) access the policy used by the agent, and (3) terminate the agent's information stream at any point. Policy safety under an information adversary is guaranteed by the policy safety conditions if the information stream can stop naturally at any point. Theorem 11 shows that policy safety with an information adversary is equivalent to absolute safety.

Theorem 11. *A policy that is safe at all times under an information adversary is also absolutely safe.*

Algorithm 2 FIND_MAXIMAL_SHADOWS_ONLINE

```
Input: $\epsilon_p$, $t$, $\{p_{t'} | \forall t' \in [0, t]\}$, $V$
Output: $\epsilon$, s.t. $\epsilon$ is greater than the sum cumulative risk
    taken before the current time $t$ and the future risk. $\epsilon$ is
    at most $\epsilon_p$ away from the minimal $\epsilon$ for which a bound
    of this class may be obtained.
$\epsilon_1 = \int_0^t p_t dt$
$\epsilon_2 =$ FIND_MAXIMAL_SHADOW_SET($\epsilon_p, \boldsymbol{\mu}_{1...n}, \boldsymbol{\Sigma}_{1...n}, V$)
return $\epsilon_1 + \epsilon_2$
```

Proof. Assume for the sake of contradiction that there exists set of observations O and a time t for which absolute safety does not hold. That is to say that conditioned on these observations

$$\int_0^t p_t dt + \int_t^\infty E[p_t|\pi]dt > \epsilon.$$

Let the information adversary stop the flow of information to the robot at time t. Let A_1 denote the event that the system fails during times $(0, t]$ and A_2 denote the event that the system fails during times (t, ∞). We note that a system can fail at most once, so A_1, A_2 are exclusive.

$E[p_t \mid O]$ is martingale so we can use the oracle developed in our algorithm:

$$\int_t^\infty E[p_t|O]dt = P(A_2|O).$$

Since $\int_0^t p_t|Odt = P(A_1|O)$, we get that the probability of failure exceeds ϵ:

$$P(A_1 \cup A_2) = P(A_1) + P(A_2) \geq \epsilon.$$

The above violates our assumption that the system is policy safe under the set of observations O.

Thus if a system is policy safe under an information adversary, it is also absolutely safe. □

4.2 Experiments

We demonstrate a simple replanning example based on the domain presented in Figure 8. During execution the robot will receive a new observation that helps it refine its estimate of the larger, second obstacle. This allows it to shrink the volume of the shadow corresponding to the same probability and certify a new, shorter path as safe. This new path is shown in Figure 10. It takes this new trajectory without exceeding the lifetime risk threshold.

5 Computing Safe Plans

The safety certification algorithms we presented above can be used for more than just checking safety. They can enable safe planning as well. We present a modification to the RRT algorithm that restricts output to only safe plans (Lavalle 1998). Every time the tree is about to be expanded, the risk of the trajectory to the node is computed. The tree is only grown if the risk of the resulting trajectory is acceptable.

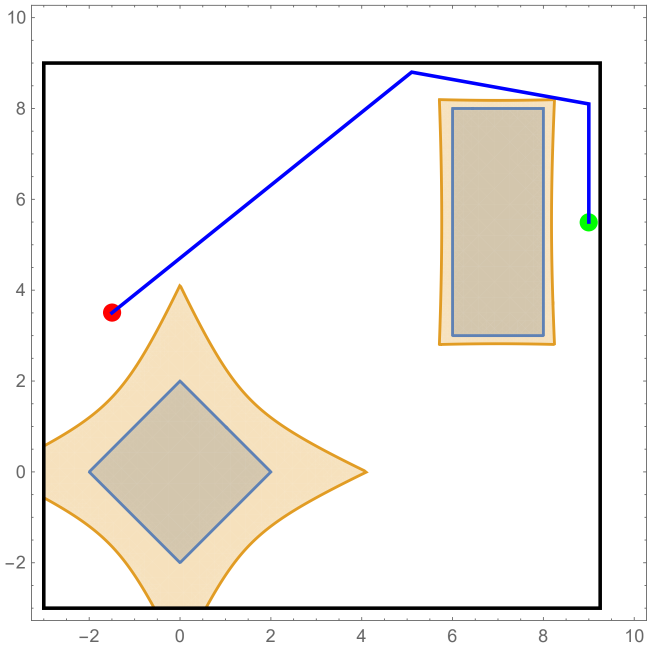


Figure 10. In the continuation of the scenario described in Figure 8 the robot makes a new observation of the obstacle after reaching the top of the old trajectory. This allows it to shrink the shadow around the second obstacle and take the less conservative path that is shown above.

Algorithm 3 SAFE_RRT

```
Input: $\epsilon_{safe}, \epsilon_p, q_s, \boldsymbol{\mu}_{1...n}, \boldsymbol{\Sigma}_{1...n}$
Output: A tree of trajectories with collision probability
    less than $\epsilon_{safe}$ of collision, computed with numerical
    precision $\epsilon_p$.
$tree$ = new TREE($q_s$)
for iteration $= 1...num\_iters$ do
    $x_{rand}$ = RANDOM_STATE
    $x_{near}$ = NEAREST_NEIGHBOR(TREE, $x_{rand}$)
    $x_{new}$ = EXTEND($x_{near}, x_{rand}$)
    $X$ = GET_TRAJECTORY_SWEPT_VOL($tree, x_{near}, x_{new}$)
    $risk$ =
      FIND_MAXIMAL_SHADOW_SET($\epsilon_p, \boldsymbol{\mu}_{1...n}, \boldsymbol{\Sigma}_{1...n}, X$)
    if $risk \leq \epsilon_{safe}$ then
        ADD_CHILD($tree, x_{near}, x_{new}$)
    end if
end for
return $tree$
```

We note that it is not necessary to check the safety of the entire trajectory every time the tree is extended. Since the bounds for each obstacle are determined by a single point in the trajectory (where the trajectory contacts the shadow), we can reuse information without running a new collision check on the whole trajectory. It is sufficient to compute the collision check for the trajectory from x_{near} to x_{new}, and use the contact points that define the highest risk shadows.

Finally we note that, unlike soundness, analyzing probabilistic completeness is quite different in the risk constrained case from the original case. We do not believe this method is probabilistically complete. Unlike the deterministic planning problem, the trajectory taken to reach a point affects the ability to reach future states—breaking down a crucial assumption required for RRTs to be probabilistically complete.

We demonstrate the safe-RRT algorithm on a point robot trying to escape a box. The box has two exits. While the robot can safely pass through the larger exit, it cannot safely pass through the smaller exit. The planner is run to only return paths with a probability of failure less than 0.5%. Figure 11a shows a safe tree from the red dot inside the box to the red dot above the box. Figure 11b shows just the ultimate trajectory with its corresponding shadows that certify the probability of failure as less than 0.26%. Note that some shadows did not extend all the way to the trajectory as their risk was already below the numerical threshold.

The experiment shown in Figure 11a demonstrates offline-safety. If the robot were given additional information during execution, we could use the equations for online-safety to re-run the RRT with the new estimates of obstacles while preserving the safety guarantee.

6 Conclusion

We presented a framework to compute shadows, the geometric equivalent of a confidence interval, around observed geometric objects. Our bounds are tighter than those of previous methods and, crucially, the tightness of the bounds does not depend on the number of obstacles. In order to achieve this tightness we rely on computing a bound specific to a trajectory instead of trying to identify a generic "safe" set.

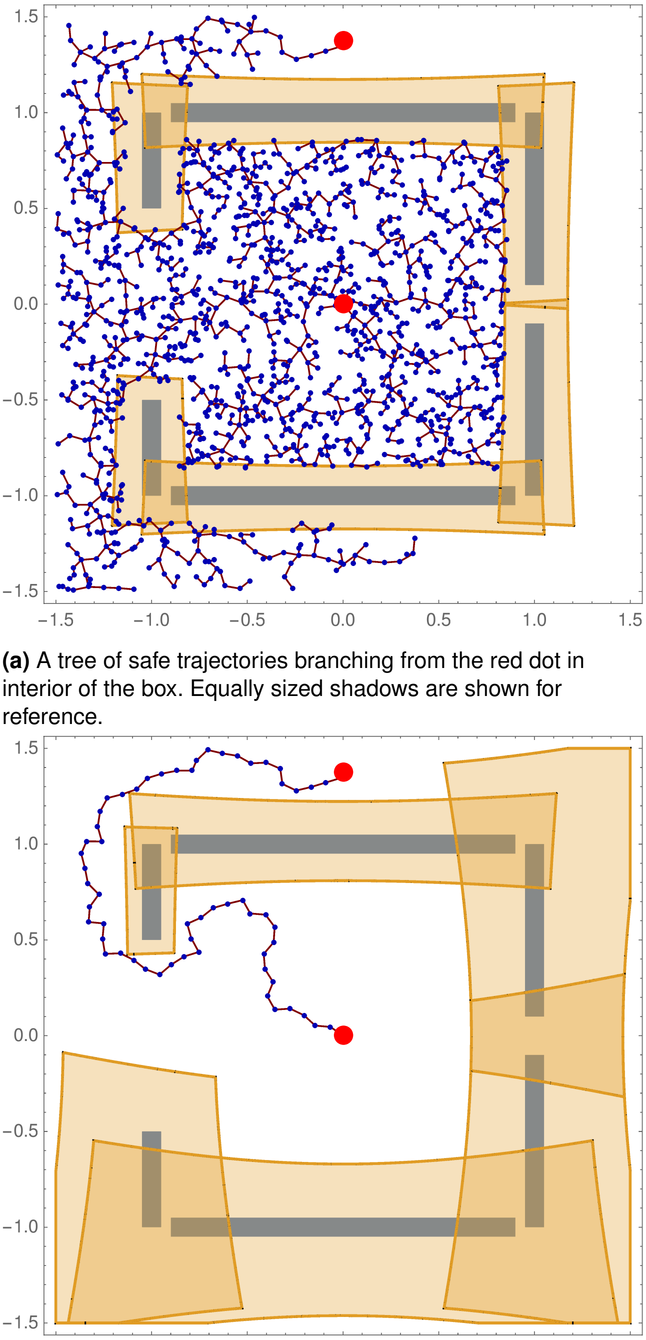
We present offline and online variants of algorithms that can verify safety with respect to the shadows identified above for both trajectories and policies. The online method highlights nuances and potential issues with a mathematical definition of safety, and presents a strong, but still computationally verifiable notion of safety. These algorithms do not have a computational complexity much larger than a collision check, and are only a $O\left(\log \frac{n}{\epsilon}\right)$ factor slower than a collision check for n obstacles and an $\epsilon-$safety guarantee. Finally the output of these algorithms is easy to verify, allowing the output to serve as a safety certificate.

These safety certification algorithms are important not only in ensuring that a given action is safe, but also in enabling the search for safe plans. We demonstrate an extension to the RRT algorithm that, while no longer probabilistically complete, only outputs safe plans.

6.1 Future work

Many of the exciting lines of work building upon the ideas presented here can be split into two categories.

6.1.1 Statistics of Estimating Geometry While we present a method of computing shadows as a way of quantifying the quality of an estimated shape we do not explore different noise models and geometric estimators. The threshold behavior in the PGDF model suggests that we may be able to develop better models for this setting. Furthermore, there is little evidence that the averaging estimator used to justify the PGDF model is optimal in practice. Considering the relationship between the shape parameter space and the robot's workspace may inform the design of estimators other than the maximum likelihood estimator that have smaller shadows similar to how the bias-variance trade-off allows one to reduce the variance of an estimator in a classical statistical setting. Furthermore, additional assumptions about the environment may allow one to construct more efficient estimators. Even in the case of PGDF obstacles, one may examine generalizations that use other approximations of the measure of the cone to create tighter shadows, mixture models that allow one to discard the assumption that the number of faces is known



(a) A tree of safe trajectories branching from the red dot in interior of the box. Equally sized shadows are shown for reference.

(b) The trajectory found by the above RRT and the corresponding shadow safety certificate

Figure 11. A graph generated by Safe-RRT to find a plan between the two red dots.

apriori, methods that compute different level shadows for different faces, etc.

6.1.2 Algorithmic Aspects of Safety with Respect to Uncertain Geometry The planning algorithm presented in this paper is not probabilistically complete. Regaining probabilistic completeness with a polynomial time increase in complexity is non-trivial, even when considering modifications to more complicated algorithms such as RRG, PRM and PRM* (Karaman and Frazzoli 2011; Kavraki et al. 1996). Minimizing risk in the setting presented in this paper on a graph can be reduced to a path-finding problem with a very structured submodular cost. While constrained submodular minimization is NP-hard in general, it remains to be seen if there are useful heuristic and approximation algorithms for this setting (Goel et al. 2009; Goemans et al. 2009; Svitkina and Fleischer 2011).

Efficient algorithms for computing shadows under different noise models and tighter bounds also remains open.

6.1.3 Other Applications of Shadows and Geometric Estimators While we propose using shadows for ensuring safe robot operation, we believe there are many other applications for geometric estimators and shadows. One could also consider modifying the definition of shadow to be a volume inside the shape with probability at least $1 - \epsilon$. The construction would remain similar. This inverted notion of a shadow may be useful in applications such as grasping where high probability of contact with the object is desirable.

Acknowledgements

BA would like to thank Gireeja Ranade, Debadeepta Dey and Ashish Kapoor for introducing him to the safe robot navigation problem during his time at Microsoft Research

Funding

We gratefully acknowledge support from the Thomas and Stacey Siebel Foundation, NSF Fellowship grant DGE-1656518, NSF grants 1420316, 1523767 and 1723381, from AFOSR FA9550-17-1-0165, from ONR grant N00014-14-1-0486, and from ARO grant W911NF1410433. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of our sponsors.

References

Bishop CM (2006) Pattern recognition. *Machine Learning* 128.

Boyd S and Vandenberghe L (2004) *Convex optimization*. Cambridge University Press.

Bry A and Roy N (2011) Rapidly-exploring random belief trees for motion planning under uncertainty. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, pp. 723–730.

Ding XC, Pinto A and Surana A (2013) Strategic planning under uncertainties via constrained markov decision processes. In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, pp. 4568–4575.

Du Toit NE and Burdick JW (2010) Robotic motion planning in dynamic, cluttered, uncertain environments. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 966–973.

Erez T and Smart WD (2010) A scalable method for solving high-dimensional continuous pomdps using local approximation. In: *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI.

Feyzabadi S and Carpin S (2016) Multi-objective planning with multiple high level task specifications. In: *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, pp. 5483–5490.

Goel G, Karande C, Tripathi P and Wang L (2009) Approximability of combinatorial problems with multi-agent submodular cost functions. In: *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*. IEEE, pp. 755–764.

Goemans MX, Harvey NJ, Iwata S and Mirrokni V (2009) Approximating submodular functions everywhere. In: *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, pp. 535–544.

Hadfield-Menell D, Groshev E, Chitnis R and Abbeel P (2015) Modular task and motion planning in belief space. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4991–4998.

Handlin M (2013) Conic sections beyond $\mathbb{R}^2$. https://www.whitman.edu/Documents/Academics/Mathematics/Handlin.pdf. Accessed: 2017-2-16.

Janson L, Schmerling E and Pavone M (2015) Monte carlo motion planning for robot trajectory optimization under uncertainty. In: *International Symposium on Robotics Research*. Sestri Levante, Italy.

Kaelbling LP, Littman ML and Cassandra AR (1998) Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101(1): 99–134.

Kaelbling LP and Lozano-Pérez T (2013) Integrated task and motion planning in belief space. *The International Journal of Robotics Research* : 0278364913484072.

Karaman S and Frazzoli E (2011) Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research* 30(7): 846–894.

Kavraki LE, Svestka P, Latombe JC and Overmars MH (1996) Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation* 12(4): 566–580.

Lavalle SM (1998) Rapidly-exploring random trees: A new tool for path planning. Technical report.

Lee A, Duan Y, Patil S, Schulman J, McCarthy Z, van den Berg J, Goldberg K and Abbeel P (2013) Sigma hulls for gaussian belief space planning for imprecise articulated robots amid obstacles. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 5660–5667.

Majumdar A, Tobenkin M and Tedrake R (2012) Algebraic verification for parameterized motion planning libraries. In: *American Control Conference (ACC), 2012*. IEEE, pp. 250–257.

Park C, Park JS and Manocha D (2016a) Fast and bounded probabilistic collision detection in dynamic environments for high-dof trajectory planning. *CoRR* abs/1607.04788.

Park JS, Park C and Manocha D (2016b) Efficient probabilistic collision detection for non-convex shapes. *CoRR* abs/1610.03651.

Platt R, Tedrake R, Kaelbling L and Lozano-Perez T (2010) Belief space planning assuming maximum likelihood observations.

In: *Proceedings of Robotics: Science and Systems*. Zaragoza, Spain.

Rasmussen CE and Williams CK (2006) Gaussian processes for machine learning. 2006. *The MIT Press, Cambridge, MA, USA* 38: 715–719.

Sadigh D and Kapoor A (2016) Safe control under uncertainty with probabilistic signal temporal logic. In: *Proceedings of Robotics: Science and Systems*. AnnArbor, Michigan.

Sun W, Torres LG, Van Den Berg J and Alterovitz R (2016) Safe motion planning for imprecise robotic manipulators by minimizing probability of collision. In: *Robotics Research*. Springer, pp. 685–701.

Svitkina Z and Fleischer L (2011) Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM Journal on Computing* 40(6): 1715–1737.