Robust Motion Planning via Perception-Aware Multiobjective Search on GPUs

Brian Ichter, Benoit Landry, Edward Schmerling, and Marco Pavone

Abstract In this paper we approach the robust motion planning problem through the lens of perception-aware planning, whereby we seek a low-cost motion plan subject to a separate constraint on perception localization quality. To solve this problem we introduce the Multiobjective Perception-Aware Planning (MPAP) algorithm which explores the state space via a multiobjective search, considering both cost and a perception heuristic. This perception-heuristic formulation allows us to both capture the history dependence of localization drift and represent complex modern perception methods. The solution trajectory from this heuristic-based search is then certified via Monte Carlo methods to be robust. The additional computational burden of perception-aware planning is offset through massive parallelization on a GPU. Through numerical experiments the algorithm is shown to find robust solutions in about a second. Finally, we demonstrate MPAP on a quadrotor flying perception-aware and perception-agnostic plans using Google Tango for localization, finding the quadrotor safely executes the perception-aware plan every time, while crashing over 20% of the time on the perception-agnostic due to loss of localization.

1 Introduction

The ability of an autonomous robotic system to localize itself in an environment is fundamental to robust operation in the field. While recent advances in perception have enabled this capability in a wide range of settings, the associated uncertainty can still vary significantly during operation. Often this is due to variations in environmental knowledge, particularly in environments derived through simultaneous localization and mapping, or due to intrinsic environmental qualities, e.g., texture or lighting. Despite this, navigation in these environments is often posed as a motion

---

Brian Ichter, Benoit Landry, Marco Pavone
Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305
e-mail: {ichter,blandry,pavone}@stanford.edu

Edward Schmerling
Institute for Computational and Mathematical Engr., Stanford University, Stanford, CA 94305
e-mail: {schmrlng}@stanford.edu

planning problem agnostic to localization, operating under the assumption that the environment is known yet the ability of a robot to localize itself against and track a trajectory is independent of its context.

In this paper, we approach the robust motion planning problem through the lens of perception-aware motion planning, whereby we seek low-cost motion plans subject to a constraint on perception localization quality. Unfortunately, the complexity of modern perception techniques and the history dependency make it difficult to quantify localization uncertainty, both computationally and theoretically, and thus a key contribution of this work is to perform perception-aware motion planning with a multiobjective search on both cost and a *perception heuristic* to identify promising motion plans. By considering trajectory cost and perception independently, both the problem formulation and algorithm can admit a wide range of perception heuristics applicable to modern closed-loop SLAM methods. The robustness of these plans is then fully certified using Monte Carlo (MC) methods. We propose a sampling-based algorithm to approximate globally optimal plans in complex environments, the additional burden of considering perception and planning in tandem is offset through algorithmic design for massive parallelization on GPUs. We term this algorithm the Multiobjective Perception-Aware Planning (MPAP) algorithm.

1.1 Related Work

While research has traditionally focused on improving methods for planning and perception, without much thought towards their later coupling, several methods to fuse the two have been put forward. One possible approach is to formally combine them in a partially observable Markov Decision Process (POMDP) framework [1][2]. This approach, however, has been shown to quickly become intractable as the complexity or dimension of the problem increases. Many other works have sought to reduce this computational load by applying search techniques to graphical-representations of the belief space. For example, the Belief Roadmap (BRM) was introduced in [3], which applies ideas from probabilistic roadmap methods to planning in the belief space. The Rapidly-exploring Random Belief Trees algorithm [4] similarly adapts the rapidly-exploring random trees algorithm to planning in the belief space. Along with the use of sampling-based approaches, there exists a large body of work (including the above) that lessens the computational load of belief space planning through local approximations, receding horizon techniques, and simplifying assumptions like linearity, Gaussian belief spaces, stabilization phases, and maximum likelihood observations [3, 4, 5, 6, 7, 8, 9, 10]. While these works approach the problem of perception-aware planning in a principled manner, the required assumptions and computational load of tracking belief states throughout the search can limit their applicability on high-dimensional problems or problems with very complex, difficult-to-quantify perception, as is the case with many modern perception methods. Our approach simplifies the problem by avoiding the computation of the belief state during the graph search through instead only considering a perception heuristic and verifying the final result with MC. Monte Carlo verification is done using the best available perception system model, and as an MC method it is asymptotically exact up to the model fidelity.

Another rich area of literature on the subject comes from the field of active perception [11, 12, 13], where the problem is focused on computing motions that minimize the uncertainty of localization and mapping. In contrast to these approaches, we pose the problem as a motion planning problem in which we directly consider the tradeoff of perception and motion plan cost.

Some of the most similar works to our own have approached the problem as perception-aware motion planning. In [14] a perception heuristic to capture the feature-richness of an environment model is introduced and used within a rapidly-exploring random tree (RRT) framework as a bound on individual edge quality. Another work, [15], proposes an approach to incorporate photometric information into the planning process, which is used as part of a weighted cost metric with path cost in an RRT search. Lastly, [16] considers an uncertainty-constrained planning problem within a receding horizon framework to compute feasible plans with mixed-integer linear programming. In contrast to these works, we pose the problem as a bound on localization uncertainty (through a heuristic) of the *entire* trajectory. This shifts the approach more towards global motion plans and allows the use of more general heuristic formulations that can capture the history dependence of localization and trajectory deviation, though the proposed heuristics in these works too can be used in this framework. As a brief example, consider a robot executing a trajectory in a learned environment that relocalizes (i.e., performs loop closure) given enough learned features in view. A weighted cost and localization uncertainty metric is unable to capture the possibility of future decreases in localization uncertainty and a edge-wise or limited time horizon approach may reject trajectories that would soon relocalize or conversely allow trajectories created from semi-localized edges that is never capable of full relocalization and continues to drift.

Finally, we note that the concept of multiobjective search to balance uncertainty with plan cost has been considered previously in motion planning, notably in [17]. We utilize the same multiobjective search accelerated through massively parallel GPU implementation put forth by the Parallel Uncertainty-aware Multiobjective Planning (PUMP) algorithm [17]. However, while PUMP considers only uncertainties internal to the system, this work considers external uncertainties through perception. Additionally, in order to handle the added complexity of modern perception methods, such as the Google Tango we experimentally demonstrate on, we instead employ a perception-heuristic constraint, and consider it in the objective search.

1.2 Statement of Contributions

In this work, we present the Multiobjective Perception-Aware Planning (MPAP) algorithm, which computes low-cost, robust motion plans through perception-aware planning. Built on a sampling-based graphical representation of the state space, the algorithm performs a multiobjective search to identify a set of Pareto optimal motion plans (considering cost and a perception heuristic). By considering cost and perception separately, the perception heuristic can consider the history-dependence of the problem while being general enough to mirror modern perception techniques. The lowest-cost trajectory contained in the Pareto optimal set can then be certified to be robust through asymptotically exact Monte Carlo methods. The entire algorithm is designed specifically to be massively parallel and thus offset the burden of perception-awareness through the computational breadth of GPUs.

We demonstrate the efficacy of MPAP through both numerical and physical experiments in mapped environments. The numerical experiments demonstrate both that perception-aware plans can be generated on the order of seconds and that these plans are significantly more robust than their perception-agnostic counterparts. The physical experiments were performed by a quadrotor flying with a Google Tango smartphone performing localization in a previously mapped room. The resulting trials compare perception-agnostic and perception-aware trajectories, finding that the perception-agnostic flights crash several times due to loss of localization while the perception-aware flights complete all trajectories safely.

1.3 Organization

The remainder of this work is organized as follows. Section 2 details the problem formulations considered herein. Section 3 formally describes the MPAP algorithm. Section 4 demonstrates the algorithm's efficacy and timing over a series of simulations. Section 5 demonstrates the algorithm through physical experiments and further motivates the need for perception-aware planning. Lastly, Section 6 summarizes our findings and offers directions for future work.

2 Problem Statement

We consider a robot with dynamics $\dot{x} = f(x,u)$ attempting to track a planned (i.e., obstacle free and connecting a start and goal state) nominal trajectory $x^*(t)$ using a closed loop controller equipped with a state estimator. This given trajectory may or may not be dynamically feasible and the robot may be subject to noise in processing, measurement, from disturbances, or other sources, all of which result in a tracking deviation $||x(t) - x^*(t)||$. Given this, in this work we approach the robust motion planning problem by seeking a minimum cost trajectory (with cost function $c(x(\cdot))$) to be followed within a bound δx with high probability.

Robust Motion Planning (RMP):

$$\begin{aligned} \min_{u^*(\cdot)} \quad & c(x^*(\cdot)) \\ \text{s.t.} \quad & \mathbb{P}(\{||x(t) - x^*(t)|| \mid t \in [0,\tau]\} \geq \delta x) \leq \alpha. \end{aligned} \tag{1}$$

Herein we focus on a common source of this deviation for modern robotic systems, that of localization error due to perception. Unfortunately, perceptual localization error can be very difficult to quantify in modern systems, particularly for entire trajectories. Instead, we propose the use a perception heuristic $h(x,\mathscr{X})$ as an analog to the notion of perceptual localization error. We thus wish to solve the following **Perception-aware Motion Planning (PMP) problem:**

$$\begin{aligned} \min_{u^*(\cdot)} \quad & c(x^*(\cdot)) \\ \text{s.t.} \quad & h(x^*,\mathscr{X}) \leq \beta. \end{aligned} \tag{2}$$

By considering the PMP problem as a cost optimization with a separate perception-heuristic constraint, this formulation allows the heuristic to capture the history dependent nature of trajectory and localization drift and can be tailored to the available sensors and algorithmic approaches. This heuristic need not be linear or monotonic, and in general should be set to the system's perception method and empirical results. We provide one possibility heuristic in Section 4, but other possibilities include covariance of uncertainty, observability, number of features in view, map entropy, loop closure probability, or a learned function.

3 The Multiobjective Perception-Aware Planning Algorithm

In this section we introduce the Multiobjective Perception-Aware Planning (MPAP) algorithm. The algorithm assumes an a priori mapped environment, including both obstacles and some sense of perceptual capabilities (e.g., locations of features), though we do present a methodology for use within a SLAM framework at the end of this section. MPAP approaches the robust motion planning problem initially through the perception-aware motion planning problem via a multiobjective search of the state space to compute the Pareto optimal front of cost and a perception heuristic. This search removes any Pareto dominated plans, as well as any plans in violation of the perception-heuristic constraint. The lowest-cost feasible plan resulting from this search is then used to inform the robust motion planning problem through a Monte Carlo (MC) verification of the full system. The resulting verification or failure can be used to iteratively update the perception-heuristic bound (either loosening or tightening) until a final solution is found. The additional computational load of a multiobjective search and a MC verification are offset through algorithmic design for massive parallelization to leverage the computational power of GPUs. This approach is outlined in Alg. 1 and thoroughly detailed below. We note this algorithm uses the same multiobjective search approach as well as massive parallelization strategy introduced in [17]. In the interest of completeness, we detail the algorithm in its entirety here, while also providing new discussion relevant to perception-aware motion planning.

Algorithm 1 MPAP: Outline

1 Massively parallel sampling-based graph building (Alg. 2)
2 Massively parallel perception heuristic computation
3 Multiobjective search to find optimal plan subject to a perception-heuristic constraint (Alg. 3)
4 Monte Carlo verification of RMP constraint and adjustment of PMP bound

We now present formal notation and algorithmic primitives to aid in the elucidation of MPAP. We define sampled nodes (samples) in $\mathbb{R}^d$ where d is the dimension of the state space. We represent plans as structs $(\text{head}, \text{path}, \text{cost}, \text{h})$, where each respective field represents the terminal trajectory node from which we may extend other plans, the list of previous nodes, the cost, and the perception heuristic. In the following definitions let $u, v \in \mathscr{X}_{\text{free}}$ be samples and let $V \subset \mathscr{X}_{\text{free}}$ be a set of samples. Define the following standard algorithmic primitives: $\texttt{SampleFree}(n)$ returns a set of $n \in \mathbb{N}$ points sampled from $\mathscr{X}_{\text{free}}$, $\texttt{Cost}(u,v)$ returns the cost of the optimal trajectory from u to v, $\texttt{Near}(V,u,r)$ returns the set of samples $\{v \in V : \texttt{Cost}(u,v) < r\}$, and $\texttt{Collision}(u,v)$ returns a boolean value indicating whether the optimal trajectory from u to v intersects $\mathscr{X}_{\text{obs}}$.

Further define the following functions specific to the MPAP algorithm. Given a set of motion plans P and $P_{\text{open}} \subset P$, $\texttt{RemoveDominated}(P, P_{\text{open}})$ denotes a function that removes all $p \in P_{\text{open}}$ from P_{open} and P that are dominated by a motion plan in $\{p_{\text{dom}} \in P : p_{\text{dom}}.\text{head} = p.\text{head}\}$; we say that p_{dom} dominates p if $(p.\text{cost} > p_{\text{dom}}.\text{cost}) \wedge (p.\text{h} \geq p_{\text{dom}}.\text{h})$. Define $\texttt{PH}(v,p)$ as a function that returns the value of the perception heuristic for the motion plan p concatenated with the optimal connection p.head to v.

With these algorithmic primitives and notation in hand, we now present the full MPAP algorithm. Note that for brevity, as well as clarity, any necessary tuning parameters or variables not explicitly passed to functions are assumed to be global. Given a mapped environment, the algorithm begins with a graph building stage (Alg. 2) to construct a representation of available motions through the state space, similar to probabilistic roadmap methods [18]. The graph is initialized with x_{init} and n samples from $\mathscr{X}_{\text{free}}$, via $\texttt{SampleFree}(n)$, to form the set V. This set of samples must include at least one from $\mathscr{X}_{\text{goal}}$. An r_n-disc graph is then formed by computing the neighbors of each sample within an r_n cost radius and checking their connection is collision free. The value of r_n can be considered a tuning parameter in line with standard practices in sampling-based planning. For kinodynamic systems, these edges should consider the respective kinematics and dynamics. This first stage represents an embarrassingly parallel process following the discussion in [19].

Algorithm 2 $\texttt{BuildGraph}$

$V \leftarrow \{x_{\text{init}}\} \cup \texttt{SampleFree}(n)$
for all $v \in V$ **do** // massively parallel graph building
  $N(v) \leftarrow \texttt{Near}(V \setminus \{v\}, v, r_n)$
  for all $u \in N(v)$ **do**
    if $\texttt{Collision}(v,u)$ **then** $N(v) \leftarrow N(v) \setminus \{u\}$ **end if**
  end for
end for
return (V, N)

With the graph complete, the second phase performs any computation of the perception heuristic that does not require full trajectories. Key to the computational tractability of this algorithmic approach is the choice of a perception heuristic amenable to significant, massively parallel computation before the full trajectory is determined in the graph search. This may be a heuristic that is derived through the combination of nodes visited or edges checked, allowing the value of each individual section to be computed independently on the GPU. We present one such heuristic in Section 4, but note that the heuristic can be quite general and should be set to the given perception method.

Following the graph construction and any massively parallel computation for the perception heuristic, a massively parallel multiobjective search is performed over the state space, considering both the trajectory cost and perception heuristic, as detailed in Alg. 3. The exploration proceeds by iteratively expanding groups, $\mathscr{G}$, in parallel to their nearest neighbors. These groups are formed from the set of open motion plans, P_{open}, below an increasing cost threshold defined for each iteration i as $i\lambda r_n$, where $\lambda \in (0,1]$ denotes the group cost factor. The value of λ is a tuning parameter that represents a tradeoff between parallelism and the potential for discarding promising plans or early termination. For higher values of λ the number of samples in each group increases, and thus the computation is made more parallel, but this comes at the cost of spreading the set P_{open} (as plans are left behind the cost threshold), which potentially allows the search to terminate before the optimal motion plan has reached $\mathscr{X}_{\text{goal}}$. At the end of each iteration any dominated plans are removed, as are any plans found in violation of the perception-heuristic constraint ($p.\text{h} > \beta$). The exploration terminates when a feasible plan reaches $\mathscr{X}_{\text{goal}}$ or $P_{\text{open}} = \emptyset$. If multiple plans are found in the Pareto front that satisfy the constraint bound, the minimum cost plan is selected. We note that in addition to the parallelism introduced over individual plans, the use of a cost bound, $i\lambda r_n$, allows for plans to be stored in cost thresholded buckets with $\mathscr{O}(1)$ insertion and removal.

Algorithm 3 `Explore`

$P_{\text{open}} \leftarrow \{(x_{\text{init}}, \emptyset, 0, 0)\}$ // plans ready to be expanded
$P(x_{\text{init}}) \leftarrow P_{\text{open}}$ // plans with head at x_{init}
$\mathscr{G} \leftarrow P_{\text{open}}$ // plans considered for expansion
$i = 0$
while $P_{\text{open}} \neq \emptyset \wedge \{g \in \mathscr{G} : (g.\text{head} \in \mathscr{X}_{\text{goal}}) \wedge (g.\text{h} \leq \beta)\} = \emptyset$ **do**
for all $p \in \mathscr{G}$ **do**
for all $x \in N(p.\text{head})$ **do**
$q \leftarrow (x, p.\text{path} + \{p.\text{head}\}, p.\text{cost} + \texttt{Cost}(p.\text{head}, x), \texttt{PH}(x, p))$
if $q.\text{h} \leq \beta$ **then** // Perception-heuristic cutoff
$P(x) \leftarrow P(x) \cup \{q\}$
$P_{\text{open}} \leftarrow P_{\text{open}} \cup \{q\}$
end if
end for
end for
$(P, P_{\text{open}}) \leftarrow \texttt{RemoveDominated}(P, P_{\text{open}})$
$P_{\text{open}} \leftarrow P_{\text{open}} \setminus \mathscr{G}$
$i \leftarrow i + 1$
$\mathscr{G} \leftarrow \{p \in P_{\text{open}} : p.\text{cost} \leq i\lambda r_n\}$
end while
$P_{\text{candidates}} \leftarrow \{p \in P(v) : v \in \mathscr{X}_{\text{goal}}\}$
return $P_{\mathscr{X}_{\text{goal}}} \leftarrow \min_{p \in P_{\text{candidates}}} \{p.\text{cost}\}$

As discussed, we employ a multiobjective search to solve the PMP problem, but it may be natural to ask why incur the computational burden of a multiobjective search instead of a combined cost function. Since we are approaching a problem that considers an objective and constraint, we should consider both independently as we do not know how they will progress in the future. For example, consider the two potential plans expanding through a narrow passageway in Fig. 1. Plan p_A has a low cost, medium constraint value (but has not yet violated the constraint), and low combined cost function, while plan p_B has a high cost, low constraint value, and high combined cost function. With our combined cost function, p_A clearly dominates and is expanded through the narrow passageway, spawning several other plans as concatenations with p_A. If all of these new plans incur high additional penalties on the constraint value, none will be valid. Unfortunately, because the narrow passageway is already filled by p_A, the potentially feasible plans that begin with p_B are never realized. The multiobjective search has the further benefit of admitting a wide range of perception heuristic functions. In this work we consider a relatively simple additive heuristic intended to emulate the variance in a localization estimate (indeed, heuristics of this form could be addressed by state augmentation in a non-sampling-based context) but, e.g., more complicated particle-based heuristics have also been applied within a similar multiobjective search framework [17].

The final phase of MPAP computes an asymptotically exact probability of motion plan p satisfying the deviation bound in Eq. 1 through MC sampling [20]. Given the output of this search, the perception-heuristic bound may be updated (incremented or decremented) and the exploration rerun. Once a final feasible plan has been determined, this has the additional benefit of giving a certificate of constraint satisfaction. This final phase too can be augmented by a smoothing algorithm to adjust the pose along the trajectory, subject to feasibility, to maximize perception (one potential approach is outlined in [21]).

Briefly, we note that this final phase can be considered optional, dependent on the ability to accurately model the system's trajectories and perception as well as the quality of the perception heuristic used. Despite the flexibility of the MC approach, in some cases due to complex dynamics, modeling the system's trajectories may be computationally intractable or the onboard perception may not be well known, rendering a MC step inaccurate. We demonstrate this case in Section 5 of this work, where a Google Tango smartphone is used for localization, which, to the authors, may be considered a black box localization framework. Without the MC verification, significant gains are still observed by simply considering the perception heuristic.

Anytime implementation: Finally, we discuss here implementing MPAP for real-time planning in conjunction with SLAM. The computational load of the MPAP algorithm is primarily in the graph building and computation of the perception heuristic (for a wide range of choices), whereas the multiobjective search is relatively lightweight. Thankfully, both of these intensive algorithms can easily be iteratively updated as new information becomes available. For the graph building, if new obstacles are discovered or regions are updated, the number of new edge collision checks can be limited. For the perception heuristic, if updates are local (such as a new region or feature's discovery), again only small changes need occur. The exploration loop, being separate algorithmically from the other two, can then be rerun as needed.

4 Numerical Experiments

4.1 Simulation Setup

In this section we demonstrate the performance of the MPAP algorithm through numerical simulations with a quadrotor-inspired dynamical system operating in a known 3D workspace with visual-inertial odometry (VIO) localization. We model our system as a 6D double integrator ($\ddot{x} = u$) with an additional single integrator yaw state. The system's dynamics are propagated by discretized approximate dynamics and the nominal trajectory is tracked by an LQR trajectory tracking controller. For simplicity, the yaw state is considered to be tracked exactly.

The system is localized in MC through a learned-feature-based VIO method. The inertial estimate is obtained through a simulated accelerator with noise. The visual localization estimate is based on matching viewed features with known 3D locations (up to a level of noise), as to emulate a learned environment. At each timestep the system identifies the features in the field of view (unobstructed by an obstacle and within a field of view angle from the system's yaw direction) and is given a relative position with noise, as may be estimated from motion and successive images (for

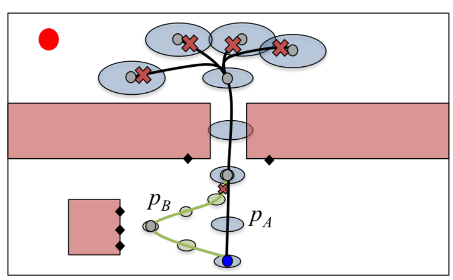


Fig. 1 An uncertainty-constrained planning problem in which a multiobjective formulation should be used over a weighted cost metric. Features to localize against are shown as black diamonds. While the partial plan p_A may outperform p_B in a combined cost metric, it eventually exceeds the constraint. A multiobjective search will consider continuations of both plans, while a combined formulation may extend only one.

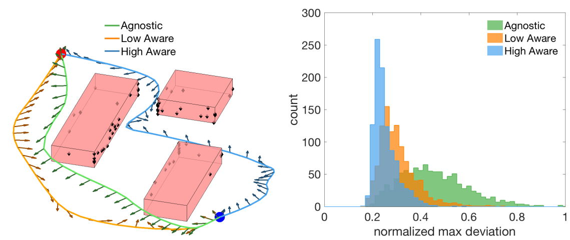
simplicity is assumed that features are identified perfectly if in the field of view). The position is then found from these features using a 3D-to-3D correspondence method described in [22]. Finally the estimates are fused via a Kalman filter.

While the goal of this work is not to introduce a new perception heuristic, and rather describe a methodology by which a simple perception heuristic can allow for perception-aware planning in difficult environments with complex systems, we introduce here a perception heuristic, based on repeated experimentation with the Google Tango smartphone, for use in the next two sections. Fig. 5 shows the localization uncertainty over several flights through an environment with varied perception and Section 5 describes the experimental setup used. We assumed a uniform density of features and a zero-feature white sheet to form the heuristic. Our proposed perception heuristic mimics the localization variance with a feature-based perception method in a learned environment. At each timestep dt along a given trajectory, an additive penalty of dt is incurred to represent increasing drift. This penalty is of when features are in view (i.e., in the field of view and unobstructed) by decrementing the heuristic by dt/n_f, where n_f represents the number of features required to offset any drift. If the number of features in view exceeds n_f, the heuristic will decrease in time, representing possible relocalization (loop closure). Based on the results in Fig. 5a we set $n_f = 12$ for the remainder of this work. If the heuristic becomes negative, it is set to zero to represent the concept that a robot can at most be fully localized. This heuristic is able to capture both some intuitional notions of our problem and handle the history-dependence of localization drift. It is one of many that could be used, and in practice the heuristic should combine theory, intuition, and empirical results. It does however demonstrate the capability of the MPAP framework even with a fairly basic perception heuristic.

4.2 Simulation Results

The simulations were implemented in CUDA C and run on an NVIDIA GeForce GTX 980 GPU on a Unix system with a 3.0 GHz CPU. Our implementation samples the state space using the deterministic, low-dispersion Halton sequence. As this set of samples is deterministic, no variance results are presented. Leveraging this determinism, we perform an offline precomputation phase to compute both nearest neighbors and edge discretizations. We also note that the obstacles used in our simulations represent axis-aligned bounding boxes, as often used in a broadphase collision checking phase [23]. This methodology can provide increasingly accurate representations of obstacles as needed. Finally, we set $\lambda = 0.5$, which we have found represents a good tradeoff between parallelism, ease of implementation, and potential early termination.

The first simulation environment, shown in Fig. 2, is rather simple, but demonstrates the benefits of perception-aware planning. The obstacle was generated to allow for low-cost trajectories around the outside of the workspace, but with a feature distribution favoring the inside of the workspace. Three plans were generated for varying levels of perception heuristic (from highly perception-aware to perception-agnostic). As can be seen, the perception-agnostic and low perception-aware plans utilize the same solution homotopy, varying primarily in yaw direction. The highly perception-aware plan however changes homotopy class. Fig. 2b and Table 1 show the MC simulation results, which demonstrate the robustness benefits both in terms



(a) (b)

Fig. 2: (2a) Motion plans resulting from MPAP with varying levels of perception heuristic bounds. Mapped features are shown in black, but only visible for localization if within the field of view and unobstructed. The low aware trajectory improves over the agnostic trajectory primarily with changes in yaw, whereas for high awareness the solution homotopy class must change. (2b) The histogram of maximum trajectory deviation from MC simulations demonstrates perception-awareness results in both lower mean trajectory deviation, but also a much lower variance.

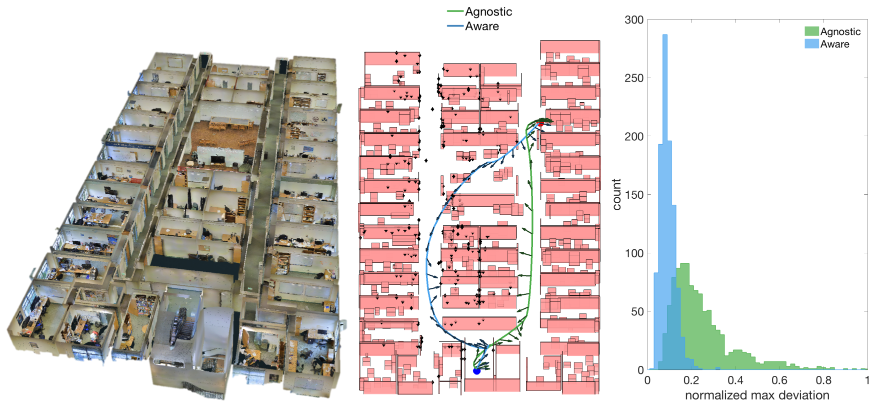
of mean, but also variance. The distributions vary most significantly in the length of the right tails, which is the regime in which the RMP problem is most impacted. The maximum deviation for the top 1% of plans is shown in Table 1.

The second simulation environment, shown in Fig. 3, represents a more complex, realistic environment generated from [24], where individual objects are bounded by axis-aligned boxes. The problem's initial state and goal region require the planner to compute trajectories for the full size of the building. The features were generated in this environment to favor the high-cost hallway, representing an incomplete mapping in the low-cost hallway. The resulting trajectories show the perception-agnostic formulation plans a more direct trajectory, while the perception-aware formulation plans a more robust trajectory at a higher cost. Corroborating the results in the other obstacle set, the mean and variance of the perception-aware trajectory is lower, but the largest difference occurs in the right tails of the distributions (Table 1).

Lastly, Table 2 details the computation time for each phase and the cost over a range of selected sample counts. On average over all runs, the computation time is allocated as 2% graph building, 61% perception heuristic computation, 8% exploration, and 29% MC. This further motivates the use of an anytime implementation discussed at the end of Section 3 to offset the computation cost of the perception heuristic computation. For other cases when the environment is known a priori (as is assumed in this work), the computation time reduces to the multiobjective search and potential MC verification. The MC trials too can be computationally intensive due to the repeated computations of field of view, though we note our current implementation is completely unoptimized and considering a trajectory assumed fully recomputing features at 50 Hz.

	Fig. 2			Fig. 3	
Perception-Awareness	High	Low	Agnostic	High	Agnostic
Max Deviation in Top 1% of Plans	0.21	0.59	0.84	0.21	0.79
Cost	288	211	201	263	193

Table 1: Simulation results over varying levels of perception-awareness. Note the max deviations are normalized by the max deviation observed over all 1000 trials.



(a) (b) (c)

Fig. 3: (3a) The point cloud representing the environment [24]. (3b) The obstacle representation of the environment as well as perception-aware and perception-agnostic trajectories. Mapped features are shown in black, but only visible to a plan if within the field of view and unobstructed. (3c) The histogram of the distribution of maximum deviation for each trajectory, generated via MC simulations. The results show both a lower mean deviation and a significantly lower variance for the perception-aware algorithm. The distributions vary especially in the right-tails.

	Fig. 2		Fig. 3	
Sample Count	2k	4k	2k	4k
Cost	296	288	302	263
Total Time (ms)	158	261	2556	3465
Graph Building (Alg. 2)	*1*	*4*	*20*	*59*
Perception Heuristic	*41*	*133*	*732*	*2405*
Exploration (Alg. 3)	**5**	**18**	**2**	**8**
Monte Carlo	*111*	*106*	*1802*	*993*

Table 2: Perception-aware plan costs and computation times over varying samples. In practice, since the environment can be mapped a priori, the full run time can be as little as just the exploration step. Note the trajectory in Fig. 3 must traverse the entire building and represents an upper bound on what we expect the planner to compute.

5 Physical Experiments

5.1 Experimental Testbed

The quadrotor used for the experiment was based on the DJI F330 frame. A Pixhawk autopilot running the PX4 software stack [25] was used for flight control. Additionally, the platform was augmented with a Linux computer, the Odroid XU4, responsible for communicating with the Pixhawk and bridging the autopilot with the rest of the network using the Robotic Operating System (ROS). Using ROS, the platform exposes an interface for a ground-station to send position waypoints to the quadrotor. Trajectories are flown using this waypoint interface. The quadrotor was also equipped with a Google Tango smartphone. Tango performs a feature-based visual-inertial odometry along with a feature-based loop closure (termed “area learning"); note the exact features and algorithmic methods are not known to the authors and thus the MC step was not performed. The localization computed by Tango was broadcasted over the network using ROS, transformed appropriately, and then used as a sensor measurement in the autopilot's Extended Kalman Filter.

The quadrotor was flown in an area measuring approximately 3m $\times$ 3m $\times$ 4m in a room equipped with motion tracking cameras to provide ground-truth measurements, i.e., not used by the quadrotor for navigation. Instead, they are only used to evaluate the performance of the Tango position estimate. The room was learned with Tango's area learning feature with a large obstacle made up of a white table cloth in the middle of the room along with two additional sheets on the walls on one side of the room (Fig. 4a). The presence of white uniform sheets reduces the number of visual features available to Tango in those regions. Moreover, the wind disturbances generated by the propellers tend to make the sheets move when the quadrotor flies past them resulting in non-stationary features. This makes it even more difficult for Tango to accurately track features that it can reliably use for pose estimation.

5.2 Experimental Results

Two trajectories with the same initial and goal states (across the room from each other) were generated with the mapped environment (Fig. 4b). One trajectory was computed using the full MPAP algorithm, with $\lambda = 0.5$, while the other was computed as a baseline perception-agnostic trajectory. Each trajectory was flown to the goal region and then back to the initial position.

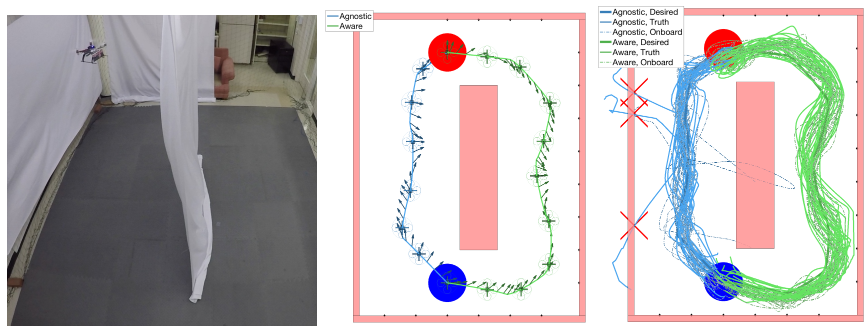
The motion plans were flown 13 times each with the resulting trajectories shown in Fig. 4c. The perception-aware trajectory was able to fly all successfully, while the perception-agnostic trajectory crashed three times, all due to loss of localization events. These events are comparable to the long right tails observed in the numerical simulations in Section 4, Figs. 2b and 3c. Outside of these localization losses, the two trajectories track the nominal approximately equally, despite the perception-aware being better localized. This is a result of the perception-aware plan incorporating more difficult turns for the controller to track. We feel incorporating this knowledge into the planning process is an interesting avenue for future research. Videos of the two different plans being flown are available on our group's Youtube channel [1].

6 Conclusions

In this work we presented the Multiobjective Perception-Aware Planning algorithm for the robust motion planning problem. The algorithm leverages massive parallelization on GPUs to perform a multiobjective search (considering cost and a perception heuristic) over a graph representation of the state space, ultimately identifying a motion plan to be certified robust via Monte Carlo methods. It is demonstrated through numerical experiments that MPAP identifies robust motion plans in under a few seconds and that the objectives of perception and cost must be considered simultaneously to obtain truly robust plans. Finally, we show through physical experiments that these perception-aware plans are indeed more robust and that the use of a simple perception heuristic can be effective in generating them.

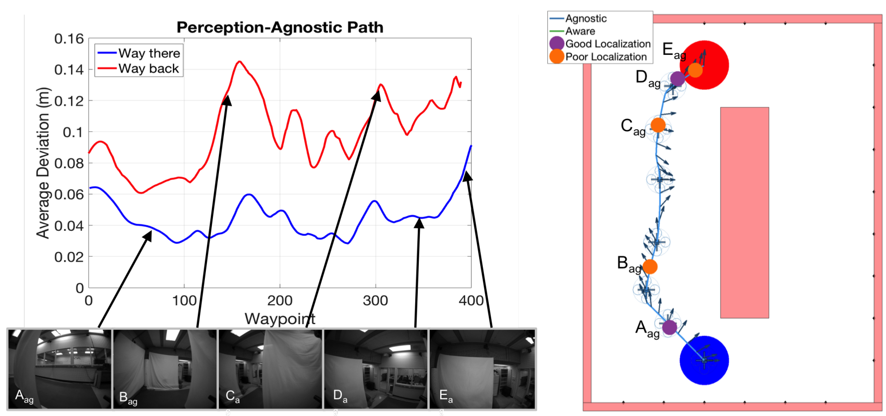
[1] https://www.youtube.com/watch?v=rMbTWamgibk

This work leaves many avenues for further investigation. First, we plan to implement the algorithm in a real-time framework with simultaneous localization and mapping. Second, we plan to investigate predictive and massively parallel perception heuristics to capture a wide variety of perception methods, particularly focusing on learned perception heuristics. Finally, while this work considered robustness through perception uncertainty, we plan to additionally consider environmental uncertainties and the ability of the controller to track trajectories while planning.



(a) (b) (c)

Fig. 4: (4a) Experimental environment setup, with white sheets on the left half of the room to reduce features. (4b) Planned trajectories, where known feature rich regions are represented with black diamonds. (4c) Desired, ground truth (Vicon), and onboard (Tango) estimate trajectories for the quadrotor over 26 flights, including 3 crashes for the perception-agnostic trajectory.



(a) (b)

Fig. 5: Localization deviation between Tango and ground truth plotted over a perception-agnostic plan. The data is averaged over 10 runs and smoothed using a moving average. The view from the Tango's fisheye lens is shown, where delocalization often corresponds to significantly higher regions of the image covered in white, feature-less sheets.

References

1. L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, “Planning and acting in partially observable stochastic domains,” *Artificial Intelligence*, 1998.
2. H. Kurniawati, D. Hsu, and W. S. Lee, “SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces,” in *Robotics: Science and Systems*, 2008.
3. S. Prentice and N. Roy, “The belief roadmap: Efficient planning in linear POMDPs by factoring the covariance,” *Int. Journal of Robotics Research*, 2009.
4. A. Bry and N. Roy, “Rapidly-exploring Random Belief Trees for motion planning under uncertainty,” in *Proc. IEEE Conf. on Robotics and Automation*, 2011.
5. J. van den Berg, S. Patil, and R. Alterovitz, “Motion planning under uncertainty using iterative local optimization in belief space,” *Int. Journal of Robotics Research*, 2012.
6. S. Patil, G. Kahn, M. Laskeym, J. Schulman, K. Goldberg, and P. Abbeel, “Scaling up gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation,” in *Workshop on Algorithmic Foundations of Robotics*, 2014.
7. V. Indelman, L. Carlone, and F. Dellaert, “Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments,” *Int. Journal of Robotics Research*, 2015.
8. J. van den Berg, P. Abbeel, and K. Goldberg, “LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information,” *Int. Journal of Robotics Research*, 2011.
9. A. Agha-mohammadi, S. Agarwal, S. Chakravorty, and N. M. Amato, “Simultaneous localization and planning for physical mobile robots via enabling dynamic replanning in belief space,” *IEEE Transactions on Robotics*, 2016, submitted, Available at https://arxiv.org/abs/1510.07380.
10. R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, “Belief space planning assuming maximum likelihood observations,” in *Robotics: Science and Systems*, 2010.
11. J. Aloimonos, I. Weiss, and A. Brandyopadhyay, “Active vision,” *Int. Journal of Computer Vision*, 1988.
12. S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
13. C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard, “Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age,” *IEEE Transactions on Robotics*, 2016.
14. S. A. Sadat, K. Chutskoff, D. Jungic, J. Wawerla, and R. Vaughan, “Feature-rich path planning for robust navigation of MAVs with Mono-SLAM,” in *Proc. IEEE Conf. on Robotics and Automation*, 2014.
15. G. Costante, C. Forster, J. Delmerico, P. Valigi, and D. Scaramuzza, “Perception-aware path planning,” *IEEE Transactions on Robotics*, 2017, submitted, Available at https://arxiv.org/abs/1605.04151.
16. L. Carlone and D. Lyons, “Uncertainty-constrained robot exploration: A mixed-integer linear programming approach,” in *Proc. IEEE Conf. on Robotics and Automation*, 2014.
17. B. Ichter, E. Schmerling, A. Agha-mohammadi, and M. Pavone, “Real-time stochastic kinodynamic motion planning via multiobjective search on GPUs,” in *Proc. IEEE Conf. on Robotics and Automation*, 2017, in press.
18. L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional spaces,” *IEEE Transactions on Robotics and Automation*, 1996.
19. N. M. Amato and L. K. Dale, “Probabilistic roadmap methods are embarrassingly parallel,” in *Proc. IEEE Conf. on Robotics and Automation*, 1999.
20. L. Janson, E. Schmerling, and M. Pavone, “Monte Carlo motion planning for robot trajectory optimization under uncertainty,” in *Int. Symp. on Robotics Research*, 2015.
21. A. J. Davison and D. W. Murray, “Simultaneous localization and map-building using active vision,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2002.
22. D. Scaramuzza and F. Fraundorfer, “Visual odometry part I: The first 30 years and fundamentals,” *IEEE Robotics and Automation Magazine*, 2011.
23. S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.
24. I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, “3D semantic parsing of large-scale indoor spaces,” in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.
25. PX4 Development Team, “PX4 autopilot,” Available at http://px4.io/.