Asymptotically Optimal Motion Planning for Tasks Using Learned Virtual Landmarks

Chris Bowen and Ron Alterovitz

***Abstract*—Utilizing appropriate landmarks in the environment is often critical to planning a robot's motion for a given task. We propose a method to automatically learn task-relevant landmarks, and incorporate the method into an asymptotically optimal motion planner that is informed by a set of human-guided demonstrations. Our method learns from kinesthetic demonstrations a task model that is parameterized by the poses of virtual landmarks. The approach models a task using multivariate Gaussian distributions in a feature space that includes the robot's configurations and the relative positions of landmarks in the environment. The method automatically learns virtual landmarks that are based on linear combinations or projections of sensed landmarks whose pose is identified using the robot's kinematic model and vision sensors. To compute motion plans for the task in new environments, we parameterize the learned task model using the virtual landmark poses and compute paths that maximally adhere to the learned task model while avoiding obstacles. We experimentally evaluate our approach on two manipulation tasks using the Baxter robot in an environment with obstacles.**

***Index Terms*—Motion and Path Planning; Probability and Statistical Methods**

I. Introduction

MANY robot manipulation tasks require planning motions relative to specific landmarks in a scene. For example, consider the task of moving a pitcher across a table and pouring liquid into a bowl as shown in Fig. 1. Successfully performing this task requires properly positioning and orienting the pitcher relative the bowl, and this relative position and orientation changes over time during the the task (i.e., the pitcher's orientation is initially level and then changes so that the liquid pours out). Implicit in performing this example task is that the robot must be aware of certain task-relevant landmarks, including (1) awareness that the bowl (rather than other landmarks in the scene, e.g., the paper towel roll) is important to the task, and (2) awareness that the position of the spout of the pitcher (as apposed to other landmarks on the pitcher) is most important to successfully pouring the liquid into the bowl. Utilizing appropriate task-relevant landmarks is often critical to successfully performing a task.

Manuscript received: August 31, 2015; Revised December 7, 2015; Accepted January 18, 2016.

This paper was recommended for publication by Editor Nancy Amato upon evaluation of the Associate Editor and Reviewers' comments. This research was supported in part by the National Science Foundation (NSF) under awards IIS-1117127 and IIS-1149965. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

Chris Bowen and Ron Alterovitz are with the Department of Computer Science, The University of North Carolina at Chapel Hill, NC, USA {cbbowen,ron}@cs.unc.edu

Digital Object Identifier (DOI): see top of this page.

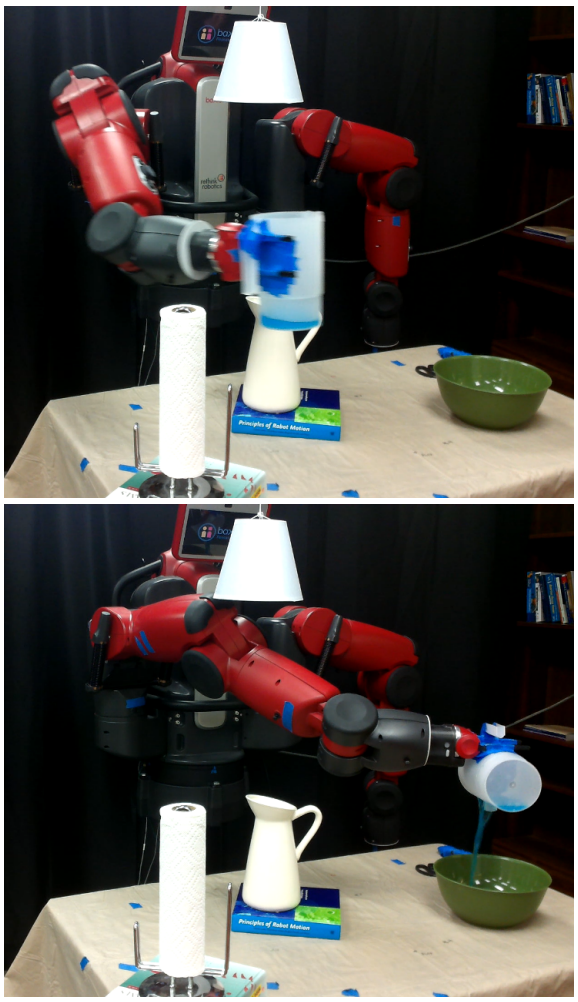

Fig. 1. The Baxter robot executes the liquid pouring task while avoiding obstacles, including the paper towel roll, vase, lamp shade, and books. Our method automatically learns task-relevant virtual landmarks, such as the relevance of the center of the bowl and the spout of the pitcher. The method also learns a time-dependent task model parameterized by the poses of the virtual landmarks, which is used by a sampling-based motion planner to compute trajectories that accomplish the task while avoiding obstacles.

In this paper we propose a method to automatically learn task-relevant landmarks, and incorporate the method into an existing approach for motion planning for a learned task [1], [2]. Prior approaches for learning manipulation tasks often assume that task-relevant landmarks are manually specified by a user (e.g., [1], [2], [3], [4], [5]). Our new method for automatically learning task-relevant landmarks (e.g., selecting the bowl and pitcher's spout in the scenario in Fig. 1) reduces the manual human effort required for a robot to learn and perform a task.

Our method aims to enable a robot to perform certain tasks in environments with obstacles by using a sampling-based roadmap informed by a set of human-guided demonstrations. Following the general approach of learning from demonstrations, during a learning phase a user provides a set of kinesthetic demonstrations of a task. In our implementation, we learn a task model that can be used in conjunction with a sampling-based motion planner [1]. To execute the learned task in a new environment where task-relevant objects may have moved, we execute an asymptotically optimal sampling-based motion planner that plans a trajectory that maximally adheres to the learned task model while avoiding obstacles.

The time-variant parameters of the learned task model encode, at discrete time steps, the means and covariances of features, including the relative positions of landmarks. Low variance features indicate aspects of the task that are likely important for successful performance due to their consistency across demonstrations. Proper selection of landmarks in the environment and on grasped objects is critical to properly capturing the relevant means and covariances, but there are many potential landmarks in a cluttered environment, of which likely only a few are relevant to the task.

In this paper we propose a method to automatically learn time-invariant parameters which specify which landmarks on the robot and in the environment should be considered during execution. We assume the robot, using its kinematic model and vision system, can sense landmarks on the robot and in the environment, such as the poses of the robot's end-effector and significant objects in the scene, both during the demonstrations and during motion planning in a new environment. During the learning phase, our method learns virtual landmarks that are based on linear combinations or projections of sensed landmarks.

To perform the task in a new environment with obstacles, we build a sampling-based roadmap and use a learned task model parameterized by the virtual landmarks to compute costs for edges in the roadmap. The costs are computed such that a path that minimizes cost best adheres to the learned task model while avoiding obstacles, including ones not present in the demonstrations. We execute the motion planner in a closed-loop manner, enabling the robot to execute the learned task in new environments while quickly reacting to the movement of sensed landmarks found to be relevant to the task.

We demonstrate the efficacy of our approach on two tasks with the Baxter robot [6] in an environment with obstacles, a powder transfer task and a liquid pouring task. Our method improved the success rate compared to arbitrary or hand-selected landmarks by automatically selecting virtual landmarks.

II. Related Work

The problem of learning a task from human demonstrations has been studied extensively, and many different approaches have been considered [7], [8].

In [1], we presented an approach based on a user-specified feature space which incorporated the position of a landmark on the robot's end-effector (or grasped object) relative to landmarks in the environment. We extend that approach with the addition of learned time-invariant parameters on which the feature space is parameterized to reduce the amount of human-provided information required. Specifically, these parameters encode the definitions of landmark which were manually-specified previously. The feature space then incorporates the relative positions of these virtual landmarks.

One class of approaches for learning from demonstration are regression methods which directly learn a reference trajectory. This is the approach taken in [9], [10], and [3] using Gaussian Mixture Regression in a feature space. The feature space used in these works inspired the one used in [1], [2], and this paper. The learning approach we present in this paper could be adapted to learn virtual landmarks to establish coordinate systems in the context of Gaussian Mixture Regression methods as well. In [11], this was extended with a feature space selection step from a finite pool of predefined features. Instead, we effective incorporate feature space selection in the learning as part of the optimization problem permitting more general feature spaces to be learned. Additionally, these approaches generally assume that the robot can always traverse the reference trajectory and are thus not directly applicable when new obstacles not present in the demonstrations are introduced.

A second class of approaches learn a control policy, mapping robot states to control inputs. Because of the dimensionality of this space, the class of policies must be restricted. In [12], [13], and [4] for example, the policies are restricted to nonlinear equations of a specific form the authors call Dynamic Movement Primitives, which are adapted to avoid obstacles locally, but not globally (i.e., by considering multiple homotopic classes of paths).

Finally, our approach can be contextualized in the class of approaches which learn a mapping from robot state to a cost (or reward). This can be thought of as a refinement of the second class of approaches, wherein the control policy learned is defined by the cost it optimizes. This is the approach taken in inverse reinforcement learning (e.g., [14], [15], [16], [17]), wherein this cost is assumed to be a parameterized function of some features of the robot state. In [5], a time-dependent multivariate Gaussian distribution is learned, and the Mahalanobis distance is taken as the cost. This approach was extended in [18] to incorporate feature space selection from a finite pool by using all the possible features and enforcing sparsity during the learning phase.

In this paper, we learn a probabilistic model based on a Hidden Markov Model. HMMs have previously been applied to motion recognition (e.g., [19], [20], [21]) and generation (e.g., [9], [21]). By framing the learning method as an optimization problem, we can simultaneously learn time-variant and time-invariant parameters of the task, including what virtual landmarks are most relevant to the task and produce the most consistent model. We then derive a cost which, when minimized, maximizes the probability that the path was generated by the learned model.

Cost-oriented approaches, like ours, are more amenable to global obstacle avoidance because asymptotically-optimal sampling-based planners like PRM*, RRG, and RRT* [22] are readily available, including variants designed to effectively

explore low-cost regions (e.g., [23], [24]). Asymptotically optimal sampling-based planners avoid the local minima inherent in potential field methods [25], and can avoid the suboptimal plans resulting from sampling-based planners which are merely probabilistically *complete* (e.g., RRT [22]). Because we rapidly replan [26], [27] when landmarks move, it is also useful to be able to access the best known plan at any given time [28]. For these reasons, we use a variation on a probabilistic roadmap (PRM), but do not allow it to grow arbitrarily large. Nevertheless, we could still guarantee near-optimality in finite time [29], [30].

III. Problem

A. Inputs and Outputs

We consider a robot with a d-dimensional configuration space $\mathcal{Q} \subseteq \mathbb{R}^d$. In order to teach the robot a task, we provide M demonstrations of the task in which the pose of task-relevant objects vary across the demonstrations. Each demonstration $m \in \{1 \dots M\}$ is a sequence of S_m observed configurations of the robot at fixed time intervals. Let $\mathbf{q}_m^s \in \mathcal{Q}$ denote the s'th observed configuration in demonstration m. Additionally, we assume we have for each demonstration m, a description of the environment $\mathbf{a}_m$ which lists the poses in $SE(3)$ of Z sensed landmarks in the demonstration environment. We assume that the landmarks may be distinguished from each other (e.g., via visual feature matching). The resulting task model will ultimately specify which subset of these landmarks is relevant to the task and must be present in the execution environment.

Although we assume that obstacles in the execution environments do not move during execution, we do not require that they be present in the demonstrations or known during the learning phase. To enable the robot to successfully perform the task in environments with never-before-seen obstacles, the problem we consider is that of estimating the parameters in a parametric probabilistic task model amenable to use by a sampling-based motion planner given these demonstrations.

We consider the problem of estimating parameters defining the task model of two distinct types:

- ζ-parameters, which will encode the position of a virtual landmark on a grasped object (e.g., a tool) relative to the robot's end-effector and what linear combination of the Z sensed landmarks define a virtual landmark in the environment. Such parameters are time-invariant; they do not vary with time or between demonstrations.
- η-parameters, which represent the dependence on task progress, like the tendency for a landmark on a grasped object to be at a specific position relative to a landmark in the environment at some time point in the task. Specifically, these parameters consist of means and covariances in a ζ-parameterized feature space incorporating the positions of points on a grasped object relative to landmarks (discussed in detail in Section IV-C). Such parameters are time-variant; they vary during the task (although not between demonstrations).

We note that the challenge of learning the η-parameters was addressed in prior work [1], but that work assumed the ζ-parameters were manually provided by a human user. To extend that work to consider time-invariant ζ-parameters, which results in non-local interactions between time steps, we explicitly re-frame the problem as an optimization.

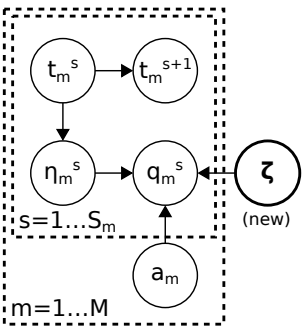

Fig. 2. Bayesian network describing independence assumptions for the learned task model. Note that for a specific demonstration, this generalizes a Hidden Markov Model with the addition of ζ-parameters.

Once the task has been learned, we then consider the problem of executing the task in new environments with new obstacles. This requires computing an obstacle-free path in the robot's configuration space from its start configuration $\mathbf{q}_{\text{start}} \in \mathcal{Q}$ to a goal configuration $\mathbf{q}_{\text{goal}} \in \mathcal{Q}$ which incorporates learned information from the task model and considers the sensed locations of the landmarks that were found to be relevant to the task during learning. We compute paths by constructing a notion of cost for which minimum cost paths correspond to maximum probability paths given the model, and applying this cost metric to an asymptotically optimal motion planner.

B. Probabilistic Task Model

The ζ- and η-parameters form the basis for a task model similar to a Hidden Markov Model (Fig. 2) with discrete states, which we call *time steps*, $\{1 \dots T\}$, where the probability of observing configuration $\mathbf{q}$ at time step t during demonstration m is given by $p(\mathbf{q} \,|\, \boldsymbol{\zeta}, \boldsymbol{\eta}_t)$ and the probability of transitioning to time step t' from t is given by $p(t' \,|\, t)$. We also consider priors on each latent parameter, denoted $p(\boldsymbol{\zeta})$ and $p(\boldsymbol{\eta})$.

Let $\mathbf{Q} = \{\mathbf{q}_{1\dots M}^{1\dots S_m}\}$ denote the observed configurations from each demonstration, and let $\mathbf{A} = \{\mathbf{a}_{1\dots M}\}$ denote the environment descriptions from each demonstration that list the sensed landmark poses. Let $\mathbf{H} = \{\boldsymbol{\eta}_{1\dots T}\}$ denote the set of η-parameters from each time step, and let $p(\mathbf{H})$ be given by $p(\boldsymbol{\eta}_1, \dots, \boldsymbol{\eta}_T) = \prod_{t=1}^{T} p(\boldsymbol{\eta}_t)$ by independence. For convenience of notation, we will let $\boldsymbol{\eta}_m^s = \boldsymbol{\eta}_{t_m^s}$ denote the η-parameter corresponding to the time step associated with the s'th observation in demonstration m. There are still only T such parameters; this notation merely serves as a convenient view of them. By the conditional independence properties of the given model, we have the following:

$$p(\mathbf{Q}, \boldsymbol{\zeta}, \mathbf{H} \,|\, \mathbf{A}) = \tag{1}$$
$$p(\boldsymbol{\zeta}) \cdot \prod_{t=1}^{T} p(\boldsymbol{\eta}_t) \cdot \prod_{m=1}^{M} \prod_{s=1}^{S_m} (p(t_m^{s+1} \,|\, t_m^s) p(\mathbf{q}_m^s \,|\, \boldsymbol{\zeta}, \boldsymbol{\eta}_m^s, \mathbf{a}_m)).$$

Putting the model in this form separates the prior, transition, and observation distributions to facilitate parameter estimation in the following section.

IV. Learning

We first define virtual landmarks as time-invariant parameters, and then describe our approach for simultaneously estimating the time-variant and time-invariant parameters of the task model. Although much of the problem is similar to our prior work [1], the introduction of time-invariant parameters necessitates a different, global, learning approach due to their non-local nature.

A. Feature Space using Virtual Landmarks

We begin by defining the virtual landmarks that will be learned as the ζ-parameters. These virtual landmarks are based on linear combinations or projections of sensed landmarks $\mathbf{a}$ whose pose is identified using the robot's kinematic model and vision sensors.

We first consider an *environmental virtual landmark*, which is based on a linear combination of sensed landmarks in the environment. ζ includes the coefficients of this linear combination. We denote this portion of ζ as ζ^{env} and require the sum of these coefficients be 1 by locally parameterizing the tangent space of this constraint. In addition to being an intuitive way to combine sensed landmarks, constraining the L^1-norm encourages sparsity. After learning, to further enforce sparsity, we discard sensed landmarks with coefficients less than 5% to form the set of sensed landmarks required during task execution. Specifically, we compute the pose of the environmental virtual landmark as follows:

$$K_{\text{virt}}(\boldsymbol{\zeta}, \mathbf{a})^{-1} = \sum_{i=1}^{Z} \boldsymbol{\zeta}_i^{\text{env}} K_{\text{sens}}^i(\mathbf{a})^{-1}$$

where $K_{\text{sens}}^i(\mathbf{a}) \in SE(3)$ denotes the pose of the i'th sensed landmark in the environment described by the input $\mathbf{a}$ from III-A.

The ζ-parameters may also encode the position of a point relative to the robot's end effector. We will denote this portion of ζ as ζ^{tool}. This point defines a *tool virtual landmark*, i.e., a point on a grasped object (e.g., a tool).

Together, these virtual landmarks, combined with the robot's configuration $\mathbf{q}$, serve to define a *feature space* that augments that used in [2] by including the position of a tool virtual landmark relative to an environmental virtual landmark. This feature space incorporates both configuration and task spaces, enabling the method to learn tasks which require both. Specifically, we define a function f to lift configurations into the feature space for learning (see Section IV-C):

$$f(\mathbf{q}, \boldsymbol{\zeta}, \mathbf{a}) = \begin{bmatrix} \mathbf{q} \\ K_{\text{virt}}(\boldsymbol{\zeta}, \mathbf{a})^{-1} K_{\text{end}}(\mathbf{q}) \boldsymbol{\zeta}^{\text{tool}} \end{bmatrix}, \qquad (2)$$

where $K_{\text{end}}(\mathbf{q})$ denotes the pose of the end effector when the robot is in configuration $\mathbf{q}$, and $K_{\text{virt}}(\mathbf{a})$ denotes the pose of the environmental virtual landmark.

B. Maximum a Posteriori Estimation

Given a set of demonstrations, our goal is to find the maximum *a posteriori* probability (MAP) estimates for $\boldsymbol{\zeta}$ and $\mathbf{H}$. To accomplish this, we combine dynamic time warping (DTW) [31] and local optimization using an expectation-maximization (EM) approach.

For numerical convenience, we first take the negative logarithm of Eq. (1), a common loss function in the machine learning literature. By monotonicity, minimizing this quantity is equivalent to maximizing the original function. To facilitate this transformation, let $L(\cdot)$ denote $-\log p(\cdot)$, yielding the following:

$$L(\mathbf{Q}, \boldsymbol{\zeta}, \mathbf{H} \mid \mathbf{A}) = L(\boldsymbol{\zeta}) + \sum_{t=1}^{T} L(\boldsymbol{\eta}_t) + \sum_{m=1}^{M} \sum_{s=1}^{S_m} \big(L(t_m^{s+1} \mid t_m^s) + L(\mathbf{q}_m^s \mid \boldsymbol{\zeta}, \boldsymbol{\eta}_m^s, \mathbf{a}_m) \big).$$

In this form, it is perhaps apparent that if we fix the time step t_m^s corresponding to each observation $\mathbf{q}_m^s$, the problem of finding the most likely ζ- and η-parameters becomes one of performing a simple, if high-dimensional, nonlinear optimization. In our implementation, this is accomplished using the Ceres nonlinear least-squares solver [32]. The transformation from this form to a nonlinear least squares problem is discussed in Section IV-C.

Dynamic time warping is used to find the most likely sequence of time steps $t_m^{1 \cdots S_m}$ corresponding to the observations $\mathbf{q}_m^{1 \cdots S_m}$ in each demonstration m using the current best estimates for the latent parameters. This is analogous to the time-alignment steps used in prior methods [5], [33]. The value update equations for the dynamic time warping which maximize likelihood are as follows:

$$l_m^0[t'] = 0$$
$$l_m^s[t'] = \min_{t \in [1,T]} \big(l_m^{s-1}[t] + L(t' \mid t) \big) + L(\mathbf{q}_m^s \mid \boldsymbol{\zeta}, \boldsymbol{\eta}_{t'}, \mathbf{a}_m),$$

where $l_m^s[u]$ denotes the value of the best assignment of time steps $t_m^{1 \cdots s}$ to observed configurations $\mathbf{q}_m^{1 \cdots s}$ from demonstration m with $t_m^s = u$.

Following the EM approach, this most likely sequence of time steps is then fixed and used to estimate new ζ- and η-parameters as discussed above. This process is repeated until convergence. Because EM approaches may become caught in local optima, we employ random restarts with randomly-chosen initial alignments.

The transition probabilities $p(t' \mid t)$ may either be fixed (the approach taken in [1]), estimated from the alignments, or a combination of the two approaches with a fixed set of permitted transitions with estimated probabilities (the approach taken in [2]). In the experiments, we use the last of these approaches.

C. Estimation via Minimization

We consider a specific class of distribution based on the assumption that observations have multivariate Gaussian distributions in a feature space at each time step. We cannot estimate time steps independently as in prior work because we simultaneously estimate time-invariant parameters.

Given the differentiable function $f(\mathbf{q}, \boldsymbol{\zeta}, \mathbf{a})$ defined in Section IV-A which lifts a configuration into feature space $\mathbb{R}^F$,

we consider $\boldsymbol{\eta}$ which captures the mean $\boldsymbol{\mu} \in \mathbb{R}^F$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{F \times F}$ in feature space. Specifically, $\boldsymbol{\eta} = [\boldsymbol{\mu}^\intercal, \mathrm{vec}[\boldsymbol{\sigma}^{-1}]^\intercal]^\intercal$ where $\mathrm{vec}[\boldsymbol{\sigma}^{-1}] \in \mathbb{R}^{F(F+1)/2}$ denotes a vector containing the components of the inverse of the lower-triangular Cholesky factorization of $\boldsymbol{\Sigma} = \boldsymbol{\sigma}\boldsymbol{\sigma}^\intercal$. In this feature space model, we consider $p(\mathbf{q} \,|\, \boldsymbol{\zeta}, \boldsymbol{\eta}, \mathbf{a})$ defined by $f(\mathbf{q}, \boldsymbol{\zeta}, \mathbf{a}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We then have the following negative log conditional probability:

$$\begin{aligned} &L(\mathbf{q} \,|\, \boldsymbol{\zeta}, \boldsymbol{\eta}, \mathbf{a}) = \\ &L(f(\mathbf{q}, \boldsymbol{\zeta}, \mathbf{a}) \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \\ &\log \|\boldsymbol{\Sigma}\| + (f(\mathbf{q}, \boldsymbol{\zeta}, \mathbf{a}) - \boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1} (f(\mathbf{q}, \boldsymbol{\zeta}, \mathbf{a}) - \boldsymbol{\mu}) = \\ &\left| \sqrt{\log \|\boldsymbol{\Sigma}\|} \right|^2 + \left| \boldsymbol{\sigma}^{-1} (f(\mathbf{q}, \boldsymbol{\zeta}, \mathbf{a}) - \boldsymbol{\mu}) \right|^2 . \end{aligned} \tag{3}$$

Minimizing this is explicitly a nonlinear least-squares problem.

The choice of representation in terms of $\boldsymbol{\sigma}^{-1}$ is particularly convenient because it admits fast computation of the residual of the transformed problem. Specifically, to compute the first term, we use the fact that $\log \|\boldsymbol{\Sigma}\| = -2 \sum_{i=1}^{F} \log |\boldsymbol{\sigma}_{i,i}^{-1}|$ by the multiplicative property of the determinant and triangularity of $\boldsymbol{\sigma}$ (and thus of $\boldsymbol{\sigma}^{-1}$). In the second term, the residual is linear in both $\boldsymbol{\sigma}^{-1}$ and $\boldsymbol{\mu}$ and thus multilinear in $\boldsymbol{\eta}$. This not only decreases the computation time needed to evaluate the residual, but greatly improves the convergence rate and reduces local minima.

We note that the first term depends only on $\boldsymbol{\sigma}^{-1}$ and so need not be recomputed for each observation. For computational efficiency, it can even be incorporated into the prior (effectively treating the conditional as an unnormalized distribution). In fact, this term is proportional to the logarithms of multiple classical priors [34], so such a prior can be incorporated simply by changing the scale on this term.

The only restriction we place on the priors is that they be log differentiable and note that they may be transformed similarly to the way the normalization term was above, although they may also admit more elegant forms (e.g., exponential distributions). In the experiments conducted in Section VI, we used (unnormalized) uniform priors for all parameters.

V. Motion Planning

After the task model has been learned, we use it to plan motions for the robot in new environments with obstacles. We assume we can sample the obstacle-free configuration-space $\mathcal{Q}_{\text{free}} \subseteq \mathcal{Q}$ (e.g., via rejection sampling). We additionally assume we are given a new vector $\hat{\mathbf{a}}$ encoding the sensed poses, in the new environment, of those landmarks found to be task-relevant during the learning phase. The planning method we employ is a simple adaptation of [2] to incorporate the learned ζ-parameters, which we briefly summarize below before deriving the adaptation.

A. Roadmap

To accommodate obstacles not present in the demonstrations, we use a sampling-based motion planner with path costs based on the learned task model. Specifically, we employ an asymptotically-optimal variant of a probabilistic roadmap (PRM) [22] with a biased sampling distribution [2]. A PRM consists of randomly sampled configurations in $\mathcal{Q}_{\text{free}}$ which are treated as vertices in a graph. Nearby configurations are connected by edges if an obstacle-free local plan can be found between them. This effectively discretizes configuration space as a graph similar to the way in which the task model discretizes time as a graph.

For the purposes of motion planning, the state depends not only on the configuration of the robot, but on the current time step in the task model. For this reason, we need to perform planning in space-time, which we discretize with the graph Cartesian product of the probabilistic roadmap and the task model as in [2]. In the Cartesian product graph, each vertex is defined by a pair of vertices, one from each constituent graph. Edges are defined by a vertex v from one graph and an edge $u \to w$ from the other, forming an edge $(v, u) \to (v, w)$. We call this graph a *spatiotemporal roadmap*.

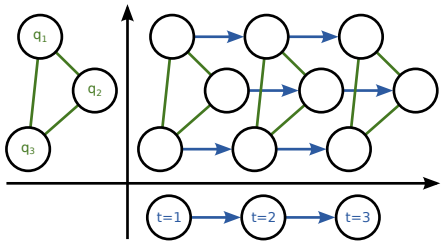

Fig. 3. **Left.** A very small probabilistic roadmap with edges shown in green. **Bottom.** A very simple task model with edges shown in blue. **Center.** Cartesian product of these graphs where edge color indicates from which constituent graph the edge was derived.

B. Search

Motion planning can be thought of as the reverse of the learning process. We wish to find the maximum probability sequence of configurations given the learned ζ- and η-parameters given $\hat{\mathbf{a}}$ while satisfying additional constraints imposed by obstacles.

To find the most probable path in the spatiotemporal roadmap, we again consider the negative logarithm of the probability. By independence (notably the Markov property), we have:

$$\begin{aligned} &\underset{t_1, \mathbf{q}_1, \ldots, t_n, \mathbf{q}_n}{\arg\max} \; p(t_1, \mathbf{q}_1, \ldots, t_n, \mathbf{q}_n \,|\, t_0, \mathbf{q}_0, \boldsymbol{\zeta}, \mathbf{H}, \hat{\mathbf{a}}) = \\ &\underset{t_1, \mathbf{q}_1, \ldots, t_n, \mathbf{q}_n}{\arg\max} \prod_{i=1}^{n} p(t_i \,|\, t_{i-1}) p(\mathbf{q}_i \,|\, \boldsymbol{\zeta}, \boldsymbol{\eta}_{t_i}, \hat{\mathbf{a}}) = \\ &\underset{t_1, \mathbf{q}_1, \ldots, t_n, \mathbf{q}_n}{\arg\min} \sum_{i=1}^{n} \left(L(t_i \,|\, t_{i-1}) + L(\mathbf{q}_i \,|\, \boldsymbol{\zeta}, \boldsymbol{\eta}_{t_i}, \hat{\mathbf{a}}) \right) \end{aligned}$$

which yields the following cost for a single edge from configuration $\mathbf{q}$ at time step t to configuration $\mathbf{q}'$ at time step t':

$$\mathrm{cost}(t, \mathbf{q} \to t', \mathbf{q}' \,|\, \hat{\mathbf{a}}) = L(t' \,|\, t) + L(\mathbf{q}' \,|\, \boldsymbol{\zeta}, \boldsymbol{\eta}_{t'}, \hat{\mathbf{a}}).$$

Finally, we note that edges in the roadmap may take varying times to traverse based on the limitations of the robot, whereas the observations in the demonstrations were taken at fixed time

Fig. 4. The Baxter robot performing the powder transfer task with the spoon being used as a tool in blue, the source container in yellow, and the destination container in green. Both the lamp and paper towel obstacles are white.

intervals. To account for this, we use the line integral of this cost across time as the actual edge cost.

We then employ a classical bidirectional shortest path search algorithm [35] in parallel on the spatiotemporal roadmap to find the minimum cost path, which by the derivation above corresponds to the most probable path given the learned task model. When new sensed landmark poses become available, we invalidate the roadmap edge costs and then perform a search, lazily recomputing new edge costs. This allows the robot to react in a closed-loop manner to the motion of the relevant landmarks during task execution at interactive rates.

VI. Results

We evaluated our method on two simplified food preparation tasks on the Baxter research robot [6]. All computation was performed on two 2.0 GHz 6-core Intel Xeon E5-2620 processors. Because we do not solve the full vision problem in this paper, sensed landmarks were continually tracked using a Kinect sensor by estimating the centroids of blobs with similar pre-defined colors.

A. Powder Transfer Task

We first tested our method on the same task as in [2], wherein the Baxter robot learned to transfer powder from one container to another (see Fig. 4) while the bowl moved. However, unlike prior work, a human did not specify a landmark on the spoon, which previously was manually specified as the tip. Instead, we automatically learned a tool virtual landmark via ζ^{tool} specifying a position relative to the pose of the robot's gripper. Additionally, we synthetically introduced 4 sensed landmarks into each demonstration in addition to the bowl, randomly sampled in the reachable space of the robot, bringing Z to 5 and the dimensionality of the ζ-parameter to 8. The feature space for learning included the robot's configuration specified by its 7 joint angles as well as the position of the tool virtual landmark relative to a learned environmental virtual landmark based on Eq. 2.

As mentioned in Section IV-C, the η-parameters used were means and covariance matrices in this feature space. For this task, we used $T = 24$ time steps. The learning algorithm was then used to estimate these parameters from the same 11 demonstrations used to evaluate the previous method [2], but with the 4 synthetic new sensed landmarks added to the set $\mathbf{A} = \{\mathbf{a}_{1...M}\}$ for each demonstration.

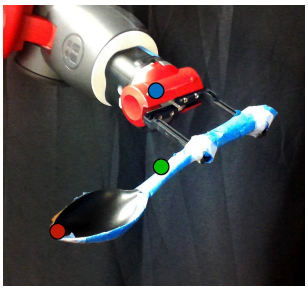

Fig. 5. A spoon being used as a tool in the powder transfer task with an arbitrary landmark at the robot's gripper marked in blue, the hand-picked landmark at the tip marked in red, and the tool virtual landmark learned by our method marked in green.

Tool Landmark (on spoon)	Environment Landmark (bowl)	Success Rate	Learning Time	Planning Latency
Arbitrary	Manual	60%	26s	201ms
Manual	Manual	80%	26s	226ms
Learned	Manual	**100%**	**1,658s**	**182ms**
Learned	**Learned**	**100%**	**1,686s**	**182ms**
(No Obstacle Avoidance)		10%	1,686s	108ms

TABLE I
Powder transfer task results averaged across 10 scenarios.

The actual virtual tool landmark learned for the spoon differed notably from the hand-picked point at the tip used previously (see Fig. 5). The point estimated by the learning method corresponded roughly to the average point of rotation on the spoon during the dumping motion. The method also correctly learned a virtual environmental landmark which consisted only of the sensed landmark corresponding to the bowl, with effectively no contribution from the other sensed landmarks. This was because the standard deviation of the tool landmark relative to the bowl in the facing direction of the robot in one time step was only 7 cm, while for the other landmarks, it was as high as 78 cm.

We tested the new learned model using the same planner on the same 10 scenarios as in the previous method's evaluation [2]. The scenarios included uniformly random paper towel obstacle and bowl locations as well as a hanging lamp obstacle. The bowl was moved to another random location midway through the task, requiring the closed-loop motion planner to react quickly. An execution was considered successful if the robot avoided obstacles in the environment and transferred the

Fig. 6. Baxter robot performing the liquid pouring task with the pitcher (being used as a tool) with blue liquid and the green bowl. The lamp, vase, and paper towel obstacles are white.

Tool Landmark (on pitcher)	Environment Landmark (bowl)	Success Rate	Learning Time	Planning Latency
Arbitrary	Manual	40%	216s	225ms
Learned	**Learned**	**90%**	**71,462s**	**200ms**
(No Obstacle Avoidance)		30%	71,462s	132ms

TABLE II
LIQUID POURING TASK RESULTS AVERAGED ACROSS 10 SCENARIOS.

powder without spilling. A video of the robot executing this task is attached and quantitative results are provided in Table I.

Surprisingly, the new learned model resulted in a higher success rate than the previous method even though less information was manually provided, likely because the relevant covariances were better captured with the new virtual landmarks in ζ. This also slightly improved planning latency because lower variances imply narrower low-cost regions, which allow the path search to explore and consequently lazily evaluate fewer edges.

To further demonstrate the importance of appropriately choosing the tool virtual landmark, we also ran the learner arbitrarily fixing $\zeta^{\text{tool}} = \mathbf{0}$ corresponding to the robot's gripper. This was only successful near the center of the table and not at the extremes where the robot completely missed the bowl.

Finally, we evaluated planning using only local optimization of the learned task model without any obstacle avoidance. As expected, this resulted in many failures due to collisions with obstacles. The results of each approach are shown in Table I.

B. Liquid Pouring Task

We next tested our method on the task of pouring liquid from a grasped pitcher into a bowl. The pose of the bowl and 4 other sensed landmarks (a pair of scissors, a spoon, a vase, and a roll of paper towels) varied between executions, bringing Z to 5. The feature space used to construct the probabilistic model was the same as for the powder transfer task, again with $T = 24$ time steps and the dimensionality of the ζ-parameter was again 8.

We performed 11 demonstrations of the task which we then supplied to the learning method with the pose of each sensed landmark, including the bowl, selected uniformly at random from a 30-inch square on the surface of the table. Additionally, the extra sensed landmarks were lifted up to 10 inches off the table surface. The learning phase took significantly longer than for the transfer task (see Table II) because the demonstrations were sampled at 50Hz rather than 10Hz, resulting in significantly more observations per demonstration. This could have been mitigated simply by subsampling during learning. However, this does show that the learning method is robust to different sampling rates. We note that learning only has to be done once after the demonstrations are provided, and does not need to be performed again when the robot executes the task in a new environment using the motion planner.

The method found a tool virtual landmark corresponding to a point just below the spout of the pitcher (distinct from that learned in Section VI-A). The only sensed landmark with a learned contribution to the environmental virtual landmark was the bowl. So the objects corresponding to the other sensed landmarks served only as obstacles during execution.

We tested the motion planner on 10 randomly selected scenarios which included the same objects as the demonstrations as well as a hanging lamp obstacle. An execution was considered successful if the robot avoided obstacles in the environment and poured the liquid into the bowl without spilling. During execution, the large size of the pitcher resulted in more constrained motion planning problems in some of the scenarios than in the powder transfer task, which caused one trial to result in failure to pour the liquid. A video of the robot successfully executing this task is attached and quantitative results are provided in Table II.

VII. CONCLUSION

We presented a method for performing tasks relative to initially unknown landmarks using a task model which encodes both the task motion and the landmarks required by it. We showed that such a model can be learned from human-guided demonstrations by simultaneously estimating time-variant and invariant parameters. Furthermore, this model is amenable to fast, global sampling-based motion planning.

Our method requires less user-provided information compared to prior work by enabling the robot to learn, from demonstrations, relevant virtual landmarks both on tools and in the environment. These virtual landmarks help define the feature space of the learned task model. We demonstrated

the efficacy of our approach by learning and executing two manipulation tasks on the Baxter robot, including a powder transfer task and a liquid pouring task.

In future work, we plan to consider per-demonstration latent parameters and how this impacts the planning phase. We would also like to use SIFT (or similar) features to automatically identify visual landmarks. Additionally, we would like to consider more general task models for which execution would incorporate aspects of task planning. To scale to more complex problems or problems with more landmarks, we will investigate customizing the optimization method in the learning phase to achieve better performance.

REFERENCES

[1] C. Bowen, G. Ye, and R. Alterovitz, "Asymptotically-optimal motion planning for learned tasks using time-dependent cost maps," *IEEE Trans. Automation Science and Engineering*, vol. 12, no. 1, pp. 171–182, Jan. 2015.

[2] C. Bowen and R. Alterovitz, "Closed-loop global motion planning for reactive execution of learned tasks," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Sep. 2014, pp. 1754–1760.

[3] S. Calinon, D. Bruno, and D. Caldwell, "A task-parameterized probabilistic model with minimal intervention control," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2014, pp. 3339–3344.

[4] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, Feb. 2013.

[5] C. Eppner, J. Sturm, M. Bennewitz, C. Stachniss, and W. Burgard, "Imitation learning with generalized task descriptions," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2009, pp. 3968–3974.

[6] Rethink Robotics, "Baxter Research Robot," www.rethinkrobotics.com/baxter-research-robot/, 2013.

[7] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, ch. 59, pp. 1371–1394.

[8] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, pp. 469–483, 2009.

[9] S. Calinon, F. D'halluin, D. G. Caldwell, and A. G. Billard, "Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Dec. 2009, pp. 582–588.

[10] S. Calinon, *Robot Programming by Demonstration*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 2009.

[11] M. Mühlig, M. Gienger, and J. J. Steil, "Interactive imitation learning of object movement skills," *Autonomous Robots*, vol. 32, pp. 97–114, 2012.

[12] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2008, pp. 91–98.

[13] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Autonomous Robots*, vol. 32, no. 4, pp. 433–454, May 2012.

[14] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proc. Int. Conf. Machine Learning (ICML)*. New York, New York, USA: ACM Press, 2004.

[15] J. van den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2010, pp. 2074–2081.

[16] A. Boularias, J. Kober, and J. Peters, "Relative entropy inverse reinforcement learning," in *Int. Conf. Artificial Intelligence and Statistics*, vol. 15, 2011, pp. 182–189.

[17] N. Aghasadeghi and T. Bretl, "Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, no. 3, 2011, pp. 1561–1566.

[18] N. Jetchev and M. Toussaint, "Task space retrieval using inverse feedback control," in *Proc. Int. Conf. Machine Learning (ICML)*, 2011.

[19] K. Bernardin, K. Ogawara, K. Ikeuchi, and R. Dillmann, "A sensor fusion approach for recognizing continuous human grasping sequences," *IEEE Trans. Robotics*, vol. 21, no. 1, pp. 47–57, 2005.

[20] N. T. Nguyen, D. Q. Phung, and S. Venkatesh, "Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 955–960.

[21] D. Kulic, W. Takano, and Y. Nakamura, "Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden Markov chains," *Int. J. Robotics Research*, vol. 27, no. 7, pp. 761–784, Jul. 2008.

[22] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robotics Research*, vol. 30, no. 7, pp. 846–894, Jun. 2011.

[23] D. Berenson, T. Simeon, and S. S. Srinivasa, "Addressing cost-space chasms in manipulation planning," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2011, pp. 4561–4568.

[24] M. Stollenga, L. Pape, M. Frank, J. Leitner, A. Forster, and J. Schmidhuber, "Task-relevant roadmaps: A framework for humanoid motion planning," in *IEEE Int. Conf. Intelligent Robots and Systems (IROS)*, 2013, pp. 5772–5778.

[25] H. Choset, K. M. Lynch, S. A. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.

[26] D. Ferguson, N. Kalra, and A. Stentz, "Replanning with RRTs," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2006, pp. 1243–1248.

[27] M. Zucker, J. Kuffner, and M. Branicky, "Multipartite RRTs for rapid replanning in dynamic environments," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Apr. 2007, pp. 1603–1609.

[28] R. Luna, I. A. Şucan, M. Moll, and L. E. Kavraki, "Anytime solution optimization for sampling-based motion planning," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2013, pp. 5068 – 5074.

[29] J. D. Marble and K. E. Bekris, "Asymptotically near optimal planning with probabilistic roadmap spanners," *IEEE Trans. Robotics*, vol. 29, no. 2, pp. 432–444, Apr. 2013.

[30] A. Dobson and K. E. Bekris, "A study on the finite-time near-optimality properties of sampling-based motion planners," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Nov. 2013, pp. 1236 – 1241.

[31] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, Feb. 1978.

[32] S. Agarwal, K. Mierle, and Others, "Ceres Solver," http://ceres-solver.org.

[33] G. Ye and R. Alterovitz, "Demonstration-guided motion planning," in *Proc. Int. Symp. Robotics Research (ISRR)*, Aug. 2011.

[34] D. Sun and J. Berger, "Objective Bayesian analysis for the multivariate normal model," *Bayesian Statistics*, vol. 8, pp. 525–547, 2007.

[35] M. Luby and P. Ragde, "A bidirectional shortest-path algorithm with good average-case behavior," *Algorithmica*, vol. 4, no. 1-4, pp. 551–567, 1989.