Towards Learning Milestones From Demonstrations with Multimodal Sensory Input

Author Names Omitted for Anonymous Review. Paper-ID [add your ID here]

I. Introduction

Robotic Surgical Assistants (RSAs), such as Intuitive Surgical's da Vinci, are increasingly performing autonomous surgical sub-tasks such as debridement [2, 5] and precision cutting [6]. While recent results are promising, automating these sub-tasks requires immense manual effort in carefully tuning finite-state machines that integrate robot actions and perception. These hand-tuned policies can be sensitive small changes in the environment or errors in perception.

There is, however, a wealth of multimedia tele-operation data (video and kinematic recordings) of surgeons using RSAs. A key question is whether we can use this data to learn more robust execution policies in an unsupervised setting. Learning from Demonstration (LfD) is a popular framework for such policies [10]. A human operator repeatedly demonstrates (e.g., via tele-operation) to a robot how to successfully complete instances of a task. The premise of LfD is that simply mimicking the demonstrations can lead to unreliable performance. In these models, a robot learns the parameters to a parameterized policy that is invariant to small perturbations.

One exemplary model of robust LfD was proposed by Niekum et al. [7]. In this model, they automatically identified significant trajectory change points in each demonstration. Then, they specified these change points in the coordinate frame of every object in the environment. With this specification relative to each object, given a new arrangement of objects (a novel environment), they could reconstruct the demonstration data by interpolating between the change points. They successfully demonstrated this framework both in simulation and on the PR2.

The framework proposed by Niekum et al. makes a significant contribution towards learning policies that are robust to small changes in environment (e.g., no new objects are added). However, it does not address the problem of learning policies that are robust to errors in perception or manipulation. Consider a simple pick-and-place task, where a robot grasps an object and has to move that object to a target zone. If the robot has noisy perception of the object, then its grasp might fail and it is unclear what the robot should do next. On the other hand, a human operator could "recover" from the failure by re-trying the grasp.

From this example, we see that this task can be decomposed into "milestones"; sub-tasks that must be completed before proceeding. For example, successfully grasping the object is a necessary condition for overall task success. It is precisely at these points where error handling and recover is needed.

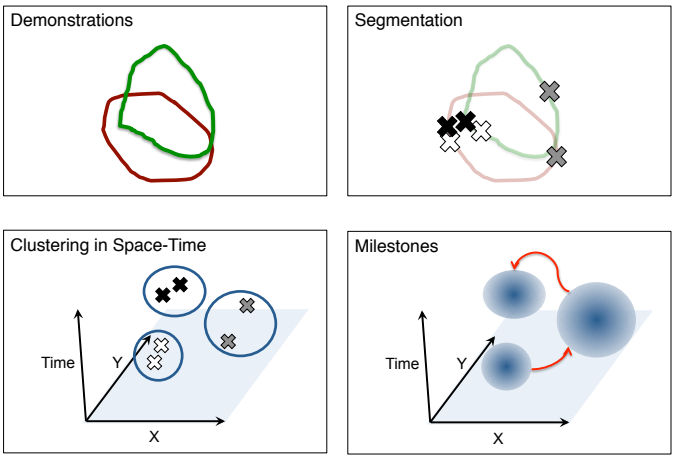

Figure 1: [Clockwise] (1) We record demonstration data that incorporates kinematics and other sensory input, (2) we identify points where the robot changes its spatial motion or interacts with an object in the environment, (3) we cluster these points both in space and time, and (4) the clusters define milestone regions.

In this paper, we take a step towards learning such milestones from demonstration. Our framework identifies features common to all demonstrations and uses these features to specify a region of the state-space at which a milestone occurs. These regions can be used as way-points for planning complex tasks. We can use a standard motion planner to plan between milestone regions. Once inside a milestone, the robot switches to user-specified behavior. The user specifies the robot actions and a success criterion, typically based on some sensor condition, for "clearing" a milestone. Once a milestone is cleared, we can proceed to the next milestone. If a milestone fails, then we can return to a previous milestone and try again. This procedure mitigates the effect of some types of recoverable failures.

One of the key insights of this work is that milestones need not be specified only spatially. Features from video can help us identify features that represent interactions with the environment such as contact with objects. We further specify milestones temporally by augmenting the feature space with time. The incorporation of time as a feature allows for theoretical analyses not possible in prior frameworks. We can formalize points in this space in terms of temporal *reachability*; a robot can reach a point $(\mathbf{x}_1, t_1)$ from $(\mathbf{x}_0, t_0)$ given physical constraints like maximum velocity. As milestones are defined both in space an time, reachability allows us to

formally define "missing" a milestone. It further allows us to handle cases where multiple sequences of operations can achieve the same goal state, as we can prune milestones that are not reachable.

To summarize our approach:

- Given a set of demonstrations $\mathcal{D}$
- We break each demonstration into segments. The segmentation criterion is based on interactions with the environment, which we formalize in Section 3.
- For each endpoint, we augment the state with time and call this a featurized endpoint.
- The featurized endpoints are candidate milestones and we cluster these candidate milestones using Dirichlet-Process Clustering [3].
- We return the cluster centers (milestones) and a distance d to the center of the cluster as a criterion for reaching the milestone.

II. Related Work

A. Motion Primitives and LfD

Using motion primitives in LfD was initially proposed by Ijspeert et al. [1]. They argued that while LfD was a theoretically attractive model, in practice, it suffered inefficiency in high-dimensional action spaces. They proposed a model where parametrized the action space in terms of motor primitives and learned a policy over the parameters rather than the high-dimensional action space. This work was extended by Pastor et al. [8], who proposed Dynamic Motion Primitives (DMP). The operator demonstrates characteristic motions which are then archived into a library of motions. Then given a start and an end state the framework chains these motions together.

Niekum et al. [7] explored this problem from a statistical perspective. The authors apply a Bayesian non-parametric model (BP-AR-HMM) to segment demonstrations. They maintain a set of coordinate frames for the robot and each known object in the environment. The endpoints of each segment are clustered across these coordinate frames, which implicitly allows understanding of which objects are moving and which are stationary. The clustered segments are converted into DMPs, which can be chained together to execute a new task.

The key limitation of this work is that it is only robust to location changes of objects in the environment. It does not address problems in perception that could affect interactions with objects. Our milestone framework is a step towards this problem. By learning points in space and time where the robot must interact with the environment, we learn natural points for error handling. In this work, the error handling is user-specified, but in the future, we hope to automatically learn this. There is also a further opportunity to extend segment-based LfD to multi-modal data. In this work, we explore how environment information from multimodal input can be incorporated into the segmentation phase, by lifting the state-space into a featurized space that includes information from the environment. The result is invariant points that captures interactions with the environment that can be detected from sensory input such as videos.

B. Surgical Task Recognition

In Surgical Robotics, researchers have studied the related problem of classifying and evaluating tele-operated surgery. Similar to the LfD problem, since the state and action space is so large, segmentation and decomposition into primitives helps parameterize the classification problem. In Lin et al. [4, 11], the authors take a 78-dimensional state space of da Vinci manipulator motions and use Linear Discriminant Analysis (LDA) to project this state space into a 6-dimensional space in which they can separate clusters of surgical motions. They used Bayesian classification to classify these motions and showed that the automatic classification matched manual segmentation except at transition points.

In Zappella et al. [12], explored this problem using both Kinematic and Video data. Given manually segmented videos, they use features from both the videos and kinematic data to classify surgical motion. Inspired by these results, we explore the use of multimodal data (e.g, video and kinematic features) for LfD. While the milestone problem is very different from surgical motion recognition, we see our results as complementary to this line of research. Alternative criterion for task segmentation such as LDA can be used in our framework perhaps leading to milestones that align with the surgical gestures discussed in [12, 4, 11]. Furthermore, the spatio-temporal clustering that we propose could be applied to complex task recognition when there are multiple sequences of gestures that can accomplish the same goal.

There are preliminary results extending this work to the LfD setting [9] in terms of motion reconstruction from clusters. Kinematic data from expert demonstrations are time-aligned using Dynamic Time Warping and grouped using a Mixture of Gaussians model. Then using a regression model smooth motions can be generated, however, the authors defer problems such as environment interactions, perception, and planning to future work. Our milestones can be valuable addition to this work. Transition points between the Gaussian mixtures are likely sensitive to error and noisy perception. The milestone framework allows us to learn these transition points and apply user-defined error recovery.

III. Problem Statement

A. Setting and Notation

We are given as input a set of demonstration trajectories $\mathcal{D} = \{d_i\}$. Each demonstration is a time-series of states where $d_i[t]$ is the robot's state at time t, where $t = 0$ is the start of each demonstration. The state captures the spatial information about the robot kinematics and dynamics.

For each demonstration, at every time t, there is also a sensor feature vector $s[t]$. The sensor feature vector represents any information about the environment not captured in the robot's state. These features could be continuous or discrete categorical. For instance, in a pattern cutting task in robotic surgical training, scissor position with respect to pattern can be featurized as euclidean distance from target. Similarly, scissor height with respect to gauze can be categorically featurized as

$\{-1, 0, 1\}$ based on whether they below, on or above the gauze respectively. These features are called sensor features since they are derived from sensory input such as haptic sensors or computer vision. Thus the full state information of the system is represented as $x_i[t] = (d_i[t], s_i[t])$.

B. Problem 1. Segmentation

The first problem that we address is segmentation of each demonstration d_i. This problem could be addressed in a supervised context where a human manually annotates a demonstration with meta-data specifying what the robot is doing. However, we look at this problem in an unsupervised setting where the only information we have are the states $x_i[t]$. In the segmentation problem, we take a demonstration d_i and identify a set of time points $c_1, c_2, ...c_k$ that signify changes in the way a robot interacts with the environment.

To formalize this, we consider the three following events as criterion for segmentation:

1. A force is applied to the robot accelerating the robot in any translational direction. Similarly, a torque is applied to the robot causing angular acceleration in any rotational direction.
2. A robot makes contact with an object in the environment.
3. The robot applies a force or a torque to a stationary object in the environment.

The first condition has been studied in prior work. However, the latter two conditions have not been explicitly formalized. At each point c in a demonstration where these conditions are true, we define a segmentation point. This gives us a set of segmentation points over all demonstrations $\mathcal{C}$.

C. Problem 2. Milestones

In an unsupervised setting, the algorithm does not know the meaning of each segmentation point c or how they correspond across demonstrations. In the next problem, we are given as input the set of demonstrations $\mathcal{D}$ and a set of segmentation points $\mathcal{C}$. We turn each segmentation point into an featurized point that includes the robot state, sensor features, and time:

$$e_i^{(j)} = (x_i[c_j], c_j)$$

The featurization procedure is visualized in our introduction image 1. Our goal is to cluser similar segmentation points into what we call *milestones*.

Since, each cluster of these featurized points defines a milestone; it gives us a constructive definition. A milestone is a region in the full-state space and time which signify robot interactions with the environment (i.e, segmentation points). Since they are clustered across all demonstrations, they represent the region (spatial and temporal) in this featurized space in which the interaction happens.

D. Problem 3. Planning Using Milestones

At each milestone the user needs to define a condition for successfully clearing the milestone. For example, if the milestone relates to grasping an object, this can be the condition that the object has been grasped. In this work, we do not learn these conditions from demonstration and rely on human specification, and we defer automatically learning these conditions to future work.

Based on these success conditions, we propose a model for planning using milestones. We start with the following assumptions:

1. Once a milestone is successfully completed, no subsequent action can nullify its success.
2. During execution failures only happen at interaction points, thus failures happen at milestones.
3. Failures that happen at milestones are recoverable; meaning there exists a set of actions that can result in a successfully completed task.
4. Failures do not change affect future milestones.

Granted, these assumptions are restrictive, but they allow us to use milestones as natural points for error handling and recovery. Between milestone regions, we can use a standard motion planner to move the robot as we can be confident that errors are rare. Once we enter a milestone region, we have to be more careful about our interactions. We switch to a user specified routine that handles the interaction and defines the criterion for success. Upon success, we plan to the next milestone. Upon failure, we return to the previous milestone.

IV. Case Study: da Vinci Peg Transfer

For a preliminary evaluation of our model, we apply our framework to identify milestones in a peg transfer task using the da Vinci. An expert surgeon provided 11 demonstrations of transferring a block between pegs.

A. Segmentation

To segment these demonstrations, we apply the segmentation criterion discussed in the problem formulation. We used the da Vinci gripper angle to detect when the robot contacted the block and applied force to the block (Figure 2). To identify changes in spatial motion, we calculated the acceleration of the joints of the arm and when it crossed a threshold we considered it a change. We plot the resulting segment endpoints in Figure 3.

B. Clustering

Next, we cluster these endpoints into milestone regions using DP-means clustering. We compare clustering these endpoints only spatially and spatio-temporally. In Figure 4, we apply the clustering to just the spatial features. The problem is two of the segment endpoint clusters overlap in spatially even though they correspond to different actions. This leads to a cluster center placed ambiguously.

When we add the temporal features (Figure 5) we find that we can disambiguate these clusters. The spatio-temporal features clearly separates into two clusters. Furthermore, the temporal features give an ordering of the milestones. We color the milestones green to red to show where they lie temporally. Semantically, the spatio-temporal milestones are: (1) start, (2) grasp block, (3) lift and clear the peg, (4) place block on other peg, and (5) lift and clear the destination peg.

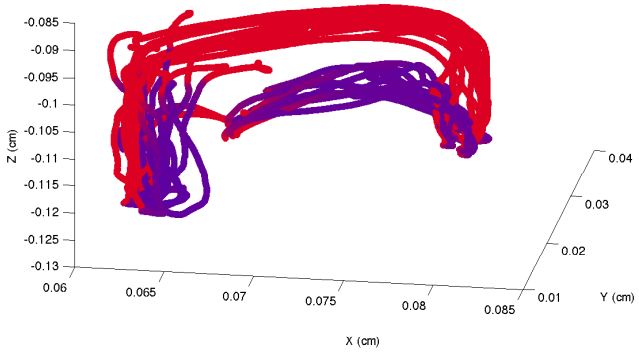

Figure 2: We visualize the demonstration trajectories colored by the gripper angle. Blue signifies a wide gripper angle (open) and red signifies closed. We use this criterion to segment based on interactions with the environment.

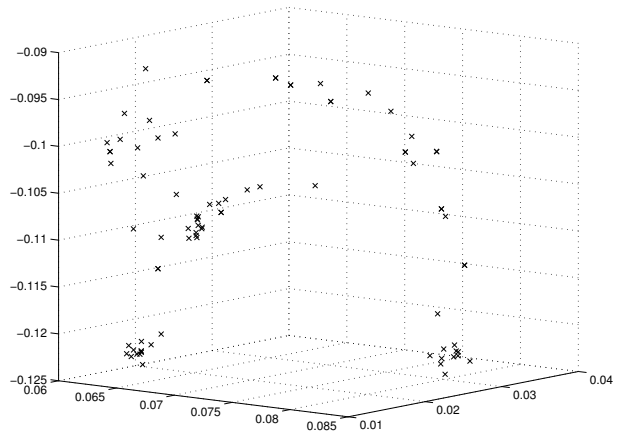

Figure 3: If we combine the gripper angle segments with accelration segments, we get the following distributions of end points in space.

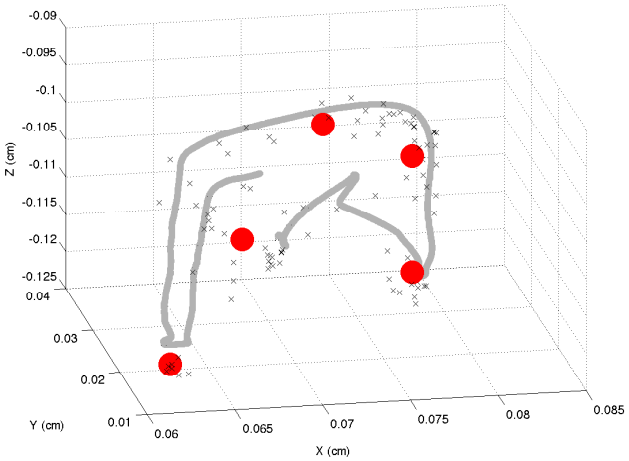

Figure 4: We plot the cluster centers (red) for clusters of the segment endpoints without temporal features; an example trajectory is plotted in gray. We find that there is some ambiguity since segments that are temporally separated are not spatially separated.

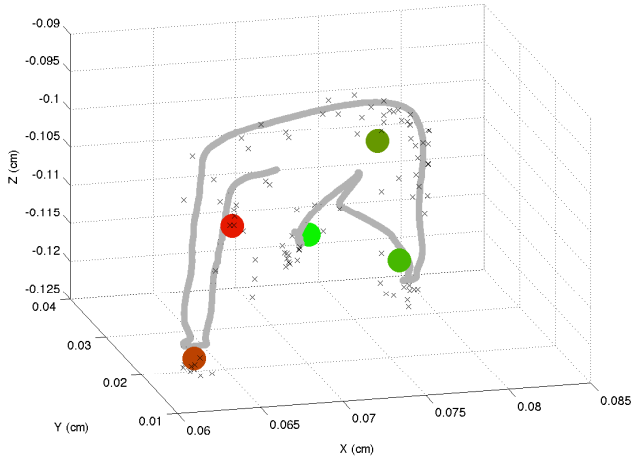

Figure 5: We plot the cluster centers using both spatial and temporal features. We find that the temporal features help disambiguate clusters that overlap in space.

REFERENCES

[1] Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal. Learning attractor landscapes for learning motor primitives. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, pages 1523–1530, 2002. URL http://papers.nips.cc/paper/2140-learning-attractor-landscapes-for-learning-motor-primitives II-A

[2] Ben Kehoe, Gregory Kahn, Jeffrey Mahler, Jonathan Kim, Alex Lee, Anna Lee, Keisuke Nakagawa, Sachin Patil, W Douglas Boyd, Pieter Abbeel, et al. Autonomous multilateral debridement with the raven surgical robot. In *International Conference on Robotics and Automation*, 2014. I

[3] Brian Kulis and Michael I Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. *arXiv preprint arXiv:1111.0352*, 2011. I

[4] Henry C Lin, Izhak Shafran, Todd E Murphy, Allison M Okamura, David D Yuh, and Gregory D Hager. Automatic detection and segmentation of robot-assisted surgical motions. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005*, pages 802–810. Springer, 2005. II-B

[5] Jeffrey Mahler, Sanjay Krishnan, Michael Laskey, Siddarth Sen, Adithyavairavan Murali, Ben Kehoe, Sachin Patil, Jiannan Wang, Mike Franklin, Pieter Abbeel, et al.

Learning accurate kinematic control of cable-driven surgical robots using data cleaning and gaussian process regression. I

[6] Adithyavairavan Murali, Siddarth Sen, Ben Kehoe, Animesh Garg, Seth McFarland, Sachin Patil, W Douglas Boyd, Susan Lim, Pieter Abbeel, and Ken Goldberg. Learning by observation for surgical subtasks: Multilateral cutting of 3d viscoelastic and 2d orthotropic tissue phantoms. I

[7] Scott Niekum, Sarah Osentoski, George Konidaris, Sachin Chitta, Bhaskara Marthi, and Andrew G Barto. Learning grounded finite-state representations from unstructured demonstrations. *The International Journal of Robotics Research*, page 0278364914554471, 2014. I, II-A

[8] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 763–768. IEEE, 2009. II-A

[9] Carol E Reiley, Erion Plaku, and Gregory D Hager. Motion generation of robotic surgical tasks: Learning from expert demonstrations. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 967–970. IEEE, 2010. II-B

[10] Stefan Schaal et al. Learning from demonstration. *Advances in neural information processing systems*, pages 1040–1046, 1997. I

[11] Balakrishnan Varadarajan, Carol Reiley, Henry Lin, Sanjeev Khudanpur, and Gregory Hager. Data-derived models for segmentation with application to surgical assessment and training. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*, pages 426–434. Springer, 2009. II-B

[12] Luca Zappella, Benjamín Béjar, Gregory Hager, and René Vidal. Surgical gesture classification from video and kinematic data. *Medical image analysis*, 17(7):732–745, 2013. II-B