Transition State Clustering: Unsupervised Surgical Trajectory Segmentation For Robot Learning

Sanjay Krishnan*[1], Animesh Garg*[2], Sachin Patil[1], Colin Lea[3], Gregory Hager[3], Pieter Abbeel[1], Ken Goldberg[1,2] *denotes equal contribution

Abstract Over 500,000 Robot-Assisted Minimally-Invasive Surgeries were performed in 2014 [11]. There is a large and growing corpus of kinematic and video recordings that have potential to facilitate human training and the automation of subtasks. A key step is to segment these multi-modal trajectories into meaningful contiguous sections in the presence of significant variations in spatial and temporal motion, noise, and looping (repetitive attempts). Manual segmentation is prone to error and impractical for large datasets. We propose Transition State Clustering (TSC), which segments a set of surgical trajectories by detecting and clustering transitions between linear dynamic regimes. TSC aggregates transition states from all demonstrations into clusters using a hierarchical Dirichlet Process Gaussian Mixture Model in two phases, first over states and then temporally. After a series of merging and pruning steps, the algorithm adaptively optimizes the number of segments, and this process gives TSC additional robustness in comparison to other Gaussian Mixture Models (GMMs) algorithms. In a synthetic case study with two linear dynamical regimes, when demonstrations are corrupted with noise and temporal variations, TSC finds up to a 20% more accurate segmentation than GMM-based alternatives. On 67 recordings of surgical needle passing and suturing tasks from the JIGSAWS surgical training dataset [8], supplemented with manually annotated visual features, TSC finds 83% of needle passing segments and 73% of the suturing segments found by human experts.

1 Introduction

Kinematic and fixed-camera video recordings from robot-assisted minimally invasive procedures (RMIS) are used in a number of applications such as surgical skill assessment [8], development of finite state machines for automation [12, 24], learning from demonstration (LfD) [28], and calibration [21]. However, even in a consistent environment (e.g., on identical tissue phantoms), leveraging the raw data is challenging. Surgical tasks are often multi-step procedures that have complex interactions with the environment, and as a result, demonstrations can vary widely.

One approach is *segmentation* of a trajectory by grouping states in locally similar segments. Existing segmentation work in robotic surgery considers the *supervised*

[1] EECS, [2]IEOR, UC Berkeley, e-mail: {sanjaykrishnan,animesh.garg, sachinpatil,pabbeel,goldberg}@berkeley.edu
[3] Computer Science Department, The Johns Hopkins University, e-mail: clea1@jhu.edu, hager@cs.jhu.edu

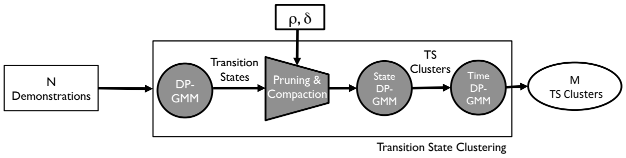


Fig. 1: (1) The TSC fits a switched linear dynamical system model via DP-GMM to identify transitions, (2) prunes and compacts transition states, (3) clusters these states spatially, and (4) subclusters temporally.

problem setting, either requiring manual segmentation of example trajectories or using a set of pre-defined primitive motions called "surgemes" [20, 29, 35]. Manual labelling requires specifying consistent segmentation criteria and applying these criteria to all demonstrations, which can be time-consuming and unreliable if applied inconsistently. On the other hand, using a dictionary of task-independent surgemes can lead to missed segments (ones not in the dictionary), and avoiding this can require defining surgemes at a very fine resolution thereby missing larger task structures.

Unsupervised segmentation, where the criteria is learned from data without labels or a pre-defined dictionary, has the potential to address these problems. Gaussian Mixture Models (GMM) have been applied to segment demonstrations to facilitate improved generalization in LfD [5, 19, 15]. However, a challenge in many LfD frameworks is coping with temporal variation across demonstrations, which is further amplified in surgery where demonstrations: (1) can vary by minutes, (2) exhibit looping behavior where surgeons repeatedly try an action until success, and (3) contain inconsistent sequences of actions for the same task. Alignment techniques such as Dynamic Time Warping are justified when variations are relatively small [13]. For larger variations, one approach is to model a demonstration as a latent finite state Markov Chain (Baum-Welch GMM+HMM model) [6]. However, recent analyses suggest that this model is very sensitive to the amount of training data and mis-specification of the number of states (segments) [30]. Non-parametric Bayesian models (e.g., Beta-Process Autoregressive HMM models [25]) and model selection criterion (e.g., Bayesian Information Criterion) can address the parameter tuning problem, but slight mis-specifications are inevitable. The surgical setting requires techniques robust to this problem.

In this paper, we propose the Transition State Clustering (TSC) algorithm (Figure 1). TSC clusters together similar (spatially and temporally) transition events that happen in most demonstrations. To do this, it hierarchically applies Dirichlet Process Gaussian Mixture Models (DP-GMM) to first identify transition states in each demonstration (i.e., states that mark changes in linear dynamical system motions) and then clusters these states across demonstrations after a series of merging and pruning steps (controlled by user-specified parameters δ, ρ). This affords some robustness to spurious states and transitions that happen in a few inconsistent demonstrations. For example, if in one suturing demonstration a surgeon pulls the needle through the tissue in a different direction the algorithm will prune this spurious action out but still consider those actions in the demonstration that were consistent.

One challenge is to leverage the video data that accompanies kinematic data in surgical demonstration recordings. In this work, we explore improving segmentation through hand-engineered visual features. We describe the video data with two features: a binary variable identifying object grasp events and a scalar variable indicating surface penetration depth. While in our experiments we construct these features via annotation, these features can be automatically calculated such as in [18]. We evaluate results with and without visual features (Section 5.4) deferring the perception problem to future work.

2 Related Work and Background

Learning From Demonstrations: The use of Motion Primitives to model complex demonstrations as a sequence on smaller segments has been well studied [10, 26]. The motion primitives are manually identified short segments of robot state-space trajectories and most of these techniques apply segmentation to discretize the action space, and this facilitates faster convergence on smaller datasets. Manschitz et al. studied modeling looping behavior and temporal variations using predefined primitives [22].

Niekum et al. [25] proposed an unsupervised extension to the motion primitive model by learning a set of primitives from demonstrations using the Beta-Process Autoregressive Hidden Markov Model (BP-AR-HMM). To incorporate environment information, after segmentation, they represent each segment in a relative coordinate frame w.r.t to every object in the environment–allowing them to generalize to new scenes using the segments. In this work, we consider a similar Bayesian non-parametric model (Dirchlet Process) which also consider environment features relevant to surgery.

Calinon et al. [2] characterizes segments from demonstrations as skills that can be used to parametrize imitation learning. This work builds on a vast body of literature of unsupervised skill segmentation including the task-parameterized movement model [4], and GMMs for segmentation [5]. In this paper, we extend this line work by applying non-parametric clustering on a GMM based model, and accounting for specific challenges such as looping and inconsistency in surgical demonstrations.

Handling Temporal Inconsistency: By far the most common model for handling demonstrations that have varying temporal characteristics is Dynamic Time Warping (DTW). When there are significant variations due to looping or additional actions (e.g., demonstrations for suturing vary between 3-5 mins), this model can give unreliable results [13]. Another model for incorporating temporal structure is to include time as a feature in the segmentation, that is a state space that is both spatial and temporal. Like DTW, this model suffices for small temporal variations. To handle larger variations, requires constructing a similarity metric that considers both space and time–which might be highly non-convex to handle structures like loops.

Another common model for modeling temporal inconsistencies is the Finite State Markov Chain model with Gaussian Mixture Emissions (GMM+HMM) [1, 3, 14, 32]. These models, also called Baum-Welch models, impose a probabilistic grammar on the segment transitions and can be learned with an EM algorithm. However, they can be sensitive to hyper-parameters such as the number of segments and the amount of data [30]. The problem of robustness in GMM+HMM (or closely related variants) has been addressed using down-weighting transient states [16] and sparsification [9]. In TSC, we explore whether it is sufficient to know *transition states* without having to fully parametrize a Markov Chain for accurate segmentation.

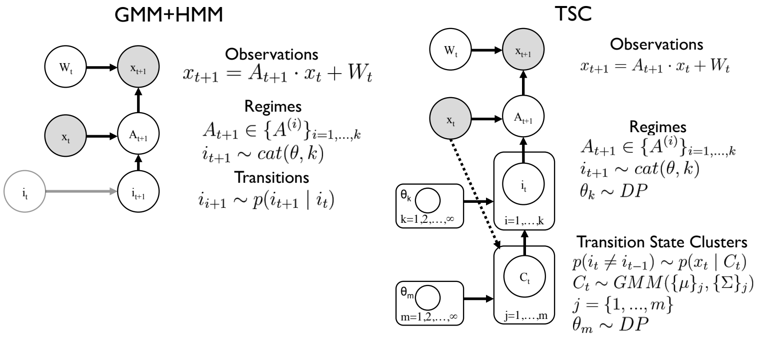

GMM+HMM
W_t
x_{t+1}
x_t
A_{t+1}
i_t
i_{t+1}
Observations
$x_{t+1} = A_{t+1} \cdot x_t + W_t$
Regimes
$A_{t+1} \in \{A^{(i)}\}_{i=1,\ldots,k}$
$i_{t+1} \sim cat(\theta, k)$
Transitions
$i_{i+1} \sim p(i_{t+1} \mid i_t)$
TSC
W_t
x_{t+1}
x_t
A_{t+1}
θ_k
$k=1,2,\ldots,\infty$
i_t
$i=1,\ldots,k$
θ_m
$m=1,2,\ldots,\infty$
C_t
$j=1,\ldots,m$
Observations
$x_{t+1} = A_{t+1} \cdot x_t + W_t$
Regimes
$A_{t+1} \in \{A^{(i)}\}_{i=1,\ldots,k}$
$i_{t+1} \sim cat(\theta, k)$
$\theta_k \sim DP$
Transition State Clusters
$p(i_t \neq i_{t-1}) \sim p(x_t \mid C_t)$
$C_t \sim GMM(\{\mu\}_j, \{\Sigma\}_j)$
$j = \{1, \ldots, m\}$
$\theta_m \sim DP$

Fig. 2: (1) A finite-state Hidden Markov Chain with Gaussian Mixture Emissions (GMM+HMM) , and (2) TSC model. TSC uses Dirchilet Process Priors and the concept of transition states to learn a robust segmentation.

We design TSC to be robust to some types of variations in demonstrations. In Figure 2, we compare the graphical models of GMM+HMM, and TSC. The TSC model applies Dirichlet Process priors to automatically set the number of hidden states (regimes). The goal of the TSC algorithm is to find spatially and temporally similar transition states across demonstrations. On the other hand, the typical GMM+HMM Baum-Welch model learns a $k \times k$ transition matrix. We empirically find that the TSC model is robust to noise and temporal variation.

Locally Linear Models: Many unsupervised segmentation models either implicitly or explicitly assume that the dynamics are locally linear. It is important to note that locally linear dynamics does not imply linear motions, as spiraling motions can be represented as linear systems. In [7], videos are modeled as transitions on a lower-dimensional linear subspace and segments are defined as changes in these subspaces. Willsky et al [34] proposed BP-AR-HMM, which was applied by Niekum et al. in robotics [25]. This model is explicitly linear by fitting a autoregressive model to time-series, where time $t+1$ is a linear function of times $t-k,\ldots,t$, to windows of data. The linear function switches according to an HMM with states parametrized by a Beta-Bernoulli model (i.e., Beta Process).

In fact, even the works that apply Gaussian Mixture Models for segmentation [5, 19, 15], implicitly fit a locally linear dynamical model. Moldovan et al. [23] proves that a Mixture of Gaussians model is equivalent to Bayesian Linear Regression; i.e., when applied to a time window it fits a linear transition between the states.

Local linear models, including the one in this work, can be extended to locally non-linear models in a straight-forward way through kernelization or increasing time window. Other non-linear models have been proposed in literature such as Locally Weighted Regression (e.g., see evaluation in [5]). These variants can be thought of as soft-time windows using weighted averages. The choice of linear vs. non-linear is orthogonal to our research contribution of segmentation robust to temporal variation.

Surgical Task Recognition: Surgical robotics has largely studied the problem of supervised segmentation using either segmented examples or a pre-defined dictionary of motions (similar to motion primitives). For example, given manually segmented videos, Zappella et al. [35] use features from both the videos and kinematic data to classify surgical motions. Simiarly, Quellec et al. [27] use manually segmented examples as training for segmentation and recognition of surgical tasks based on archived cataract surgery videos. The dictionary-based approaches are done with a domain-specific set of motion primitives for surgery called “surgemes”. A number of works (e.g., [20, 33, 31, 18]), use the surgemes to bootstrap learning segmentation.

3 Problem Setup

The TSC model is summarized by the hierarchical graphical model in the previous section (Figure 2). Here, we formalize each of the levels of the hierarchy and describe the assumptions in this work.

Dynamical System Model: Let $\mathscr{D} = \{d_i\}$ be the set of demonstrations where each d_i is a trajectory $\mathbf{x}(t)$ of fully observed robot states and each state is a vector in $\mathbb{R}^d$. We model each demonstration as a switched linear dynamical system. There is a finite set of $d \times d$ matrices $\{A_1,...,A_k\}$, and an i.i.d zero-mean additive Gaussian Markovian noise process $W(t)$ which accounts for noise in the dynamical model:

$$\mathbf{x}(t+1) = A_i\mathbf{x}(t) + W(t) : A_i \in \{A_1,...,A_k\}$$

Transitions between regimes are instantaneous where each time t is associated with exactly one dynamical system matrix $1,...,k$

TSC Model (With Visual Sensing) This model can similarly be extended to states derived from sensing. Suppose at every time t, there is a feature vector $z(t)$. Then the augmented state of both the robot spatial state and the features denoted is:

$$\mathbf{x}(t) = \begin{pmatrix} x(t) \\ z(t) \end{pmatrix}$$

In our experiments, we worked the da Vinci surgical robot with with two 7-DOF arms, each with 2 finger grippers. Consider the following feature representation which we used in our experiments:

1. *Gripper grasp*. 1 if there is an object between the gripper, 0 if not.
2. *Surface Penetration*. In surgical tasks, we often have a tissue phantom. This feature describes whether the robot (or something the robot is holding like a needle) has penetrated the surface. We use an estimate of the truncated penetration depth to encode this feature. If there is no penetration, the value is 0. If there is penetration, the value of the feature is the robot’s kinematic position in the direction orthogonal to the tissue phantom.

Transition States and Times: Transition states are defined as the last states before a dynamical regime transition in *each* demonstration. Each demonstration d_i follows a switched linear dynamical system model, therefore there is a time series of regimes $A(t)$ associated with each demonstration.

Therefore, there will be times t at which $A(t) \neq A(t+1)$. A transition state is the state $x(t)$ at time t. For a demonstration i, we denote the sequence of transitions states as $U_i = [u_i^1,...,u_i^J]$. J is the number of transition states where $J \ll T_i$ where T_i is the time-length of d_i.

Transition State Clusters: Across all demonstrations, we are interested in aggregating nearby (spatially and temporally) transition states together. A *transition state* *cluster* is defined as a clustering of the set of transition states across all demonstrations; partitioning these transition states into m non-overlapping similar groups:

$$\mathscr{C} = \{C_1, C_2, ..., C_m\}$$

In principle, any similarity-based clustering model can be applied, and in the next section, we describe using a hierarchical application of DP-GMM by first applying a GMM to states, and then sub-clustering by applying a GMM to times. Other prior segmentation works use time as a feature to the GMM, however, this leads to an issue of designing a similarity metric that considers both spatial states and time. Every U_i can be represented as a sequence of integers indicating that transition states assignment to one of the transition state clusters $U_i = [1,2,4,2]$.

Consistency: We assume, demonstrations are *consistent*, meaning there exists a non-empty sequence of transition states $\mathscr{U}^*$ such that the partial order defined by the elements in the sequence (i.e., s_1 happens before s_2 and s_3) is satisfied by every U_i. For example,

$$U_1 = [1,3,4], U_2 = [1,1,2,4], \mathscr{U}^* = [1,4]$$

A counter example,

$$U_1 = [1,3,4], U_2 = [2,5], \mathscr{U}^* \text{ no solution}$$

Intuitively, this condition states that there have to be a consistent ordering of actions over all demonstrations up to some additional regimes (e.g., spurious actions).

Loops: Loops are common in surgical demonstrations. For example, a surgeon may attempt to insert a needle 2-3 times. When demonstrations have varying amounts of retrials it is challenging. In this work, we assume that these loops are modeled as repeated transitions between transition state clusters, which is justified in our experimental datasets, for example,

$$U_1 = [1,3,4], U_2 = [1,3,1,3,1,3,4], \mathscr{U}^* = [1,3,4]$$

Our algorithm will *compact* these loops together into a single transition.

Minimal Solution: Given a consistent set of demonstrations, that have additional regimes and loops, the goal of the algorithm is to find a *minimal solution*, $\mathscr{U}^*$ that is loop free and respects the partial order of transitions in all demonstrations.

Problem 1 (Transition State Clustering Problem). *Given a set of demonstrations $\mathscr{D}$, the Transition State Clustering problem is to find a set of transition state clusters $\mathscr{C}$ such that they represent a minimal parametrization of the demonstrations.*

4 Transition State Clustering

In this section, we describe the hierarchical clustering process of TSC. This algorithm is a greedy approach to learning the parameters in the graphical model in Figure 2. We decompose the hierarchical model into stages and fit parameters to the generative model at each stage. The full algorithm is described in Algorithm 1.

4.1 Background: Bayesian Statistics

One challenge with mixture models is hyper-parameter selection, such as the number of clusters. Recent results in Bayesian statistics can mitigate some of these problems. The basic recipe is to define a generative model, and then use Expectation Maximization to fit the parameters of the model to observed data. The generative model that we will use is called a mixture model, which defines a probability distribution that is a composite of multiple distributions.

One flexible class of mixture models are Gaussian Mixture Models (GMM), which are described generatively as follows. We first sample some c from a categorical distribution, one that takes on values from (1...K), with probabilities ϕ, where ϕ is a K dimensional simplex:

$$c \sim cat(K, \phi)$$

Then, given the event $\{c = i\}$, we specify a multivariate Gaussian distribution:

$$x_i \sim N(\mu_i, \Sigma_i)$$

The insight is that a stochastic process called the Dirichlet Process (DP) defines a distribution over discrete distributions, and thus instead we can draw samples of $cat(K, \phi)$ to find the most likely choice of K via EM. The result is the following model:

$$(K, \phi) \sim DP(H, \alpha) \qquad c \sim cat(K, \phi) \qquad X \sim N(\mu_i, \Sigma_i) \tag{1}$$

After fitting the model, every observed sample of $x \sim X$ will have a probability of being generated from a mixture component $P(x \mid c = i)$. Every observation x will have a most likely generating component. It is worth noting that each cluster defines an ellipsoidal region in the feature space of x, because of the Gaussian noise model $N(\mu_i, \Sigma_i)$.

We denote this entire clustering method in the remainder of this work as DP-GMM. We use the same model at multiple levels of the hierarchical clustering and we will describe the feature space at each level. We use a MATLAB software package to solve this problem using a variational EM algorithm [17].

4.2 Transition States Identification

The first step is to identify a set of transition states for each demonstration in $\mathscr{D}$. To do this, we have to fit a switched dynamic system model to the trajectories. Suppose there was only one regime, then this would be a linear regression problem:

$$\arg\min_A \|AX_t - X_{t+1}\|$$

where X_t is a matrix where each column vector is $x(t)$, and X_{t+1} is a matrix where each column vector is the corresponding $x(t+1)$. Moldovan et al. [23] proves that fitting a Jointly Gaussian model to $n(t) = \binom{\mathbf{x}(t+1)}{\mathbf{x}(t)}$ is equivalent to Bayesian Linear Regression.

Therefore, to fit a switched linear dynamical system model, we can fit a Mixture of Gaussians (GMM) model to $n(t)$ via DP-GMM. Each cluster learned signifies a different regime, and co-linear states are in the same cluster. To find transition states, we move along a trajectory from $t = 1, ..., t_f$, and find states at which $n(t)$ is in a different cluster than $n(t+1)$. These points mark a transition between clusters (i.e., transition regimes).

4.3 Transition State Pruning

We consider the problem of outlier transitions, ones that appear only in a few demonstrations. Each of these regimes will have constituent vectors where each $n(t)$ belongs to a demonstration d_i. Transition states that mark transitions to or from regimes whose constituent vectors come from fewer than a fraction ρ demonstrations are *pruned*. ρ should be set based on the expected rarity of outliers. In our experiments, we set the parameter ρ to 80% and show the results with and without this step.

4.4 Transition State Compaction

Once we have transition states for each demonstration, and have applied pruning, the next step is to remove transition states that correspond to looping actions, which are prevalent in surgical demonstrations. We model this behavior as consecutive linear regimes repeating, i.e., transition from i to j and then a repeated i to j. We apply this step after pruning to take advantage of the removal of outlier regimes during the looping process. These repeated transitions can be compacted together to make a single transition.

The key question is how to differentiate between repetitions that are part of the demonstration and ones that correspond to looping actions–the sequence might contain repetitions not due to looping. To differentiate this, as a heuristic, we threshold the L2 distance between consecutive segments with repeated transitions. If the L2 distance is low, we know that the consecutive segments are happening in a similar location as well. In our datasets, this is a good indication of looping behavior. If the L2 distance is larger, then repetition between dynamical regimes might be happening but the location is changing.

For each demonstration, we define a segment $\mathbf{s}^{(j)}[t]$ of states between each transition states. The challenge is that $\mathbf{s}^{(j)}[t]$ and $\mathbf{s}^{(j+1)}[t]$ may have a different number of observations and may be at different time scales. To address this challenge, we apply Dynamic Time Warping (DTW). Since segments are locally similar up-to small time variations, DTW can find a most-likely time alignment of the two segments.

Let $\mathbf{s}^{(j+1)}[t^*]$ be a time aligned (w.r.t to $\mathbf{s}^{(j)}$) version of $\mathbf{s}^{(j+1)}$. Then, after alignment, we define the L_2 metric between the two segments:

$$d(j, j+1) = \frac{1}{T}\sum_{t=0}^{T}(\mathbf{s}^{(j)}[i] - \mathbf{s}^{(j+1)}[i^*])^2$$

When $d \leq \delta$, we compact two consecutive segments. δ is chosen empirically and a larger δ leads to a sparser distribution of transition states, and smaller δ leads to more transition states. For our needle passing and suturing experiments, we set δ to correspond to the distance between two suture/needle insertion points–thus, differentiating between repetitions at the same point vs. at others.

However, since we are removing points from a time-series this requires us to adjust the time scale. Thus, from every following observation, we shift the time stamp back by the length of the compacted segments.

4.5 State-Space Clustering

After compaction, there are numerous transition states at different locations in the state-space. If we model the states at transition states as drawn from a GMM model:

$$x(t) \sim N(\mu_i, \Sigma_i)$$

Then, we can apply the DP-GMM again to cluster the state vectors at the transition states. Each cluster defines an ellipsoidal region of the state-space space.

Algorithm 1: The Transition State Clustering Algorithm

```
Input: $\mathscr{D}$, $\rho$ pruning parameter, and $\delta$ compaction parameter.
$n(t) = \binom{\mathbf{x}(t+1)}{\mathbf{x}(t)}$.
Cluster the vectors $n(t)$ using DP-GMM assigning each state to its most likely cluster.
*Transition states* are times when $n(t)$ is in a different cluster than $n(t+1)$.
Remove states that transition to and from clusters with less than a fraction of $p$ demonstrations.
Remove consecutive transition states when the L2 distance between these transitions is less than $\delta$.
Cluster the remaining transition states in the state space $\mathbf{x}(t+1)$ using DP-GMM.
Within each state-space cluster, sub-cluster the transition states temporally.
Output: A set $\mathscr{M}$ of clusters of transition states and the associated with each cluster a time interval of transition
    times.
```

4.6 Time Clustering

Without temporal localization, the transitions may be ambiguous. For example, in circle cutting, the robot may pass over a point twice in the same task. The challenge is that we cannot naively use time as another feature, since it is unclear what metric to use to compare distance between $\binom{\mathbf{x}(t)}{t}$. However a second level of clustering by time within each state-space cluster can overcome this issue.Within a state cluster, if we model the times which change points occur as drawn from a GMM:

$$t \sim N(\mu_i, \sigma_i)$$

then we can apply DP-GMM to the set of times. We cluster time second because we observe that the surgical demonstrations are more consistent spatially than temporally. This groups together events that happen at similar times during the demonstrations. The result is clusters of states and times. Thus, a transition states m_k is defined as tuple of an ellipsoidal region of the state-space and a time interval.

5 Results

5.1 Experiment 1. Synthetic Example of 2-Segment Trajectory

In our first experiment, we segment noisy observations from a two regime linear dynamical system. Figure 3 illustrates examples from this system under the different types of corruption.

Evaluation Metric: Since there is a known a ground truth of two segments, we measure the precision (average fraction of observations in each segment that are from the same regime) and recall (average fraction of observations from each regime segmented together) in recovering these two segments. We can jointly consider precision and recall with the *F1 Score* which is the harmonic mean of the two:

$$f1 = \frac{2 \cdot precision \cdot recall}{precision\ +\ recall}$$

We compare three techniques against TSC: K-Means (only spatial), GMM+T (using time as a feature in a GMM), GMM+HMM (using an HMM to model the grammar). For the GMM techniques, we have to select the number of segments, and we experiment with $k = 1,2,3$ (i.e., a slightly sub-optimal parameter choice compared to $k = 2$). In this example, for TSC, we set the two user-specified parameters to

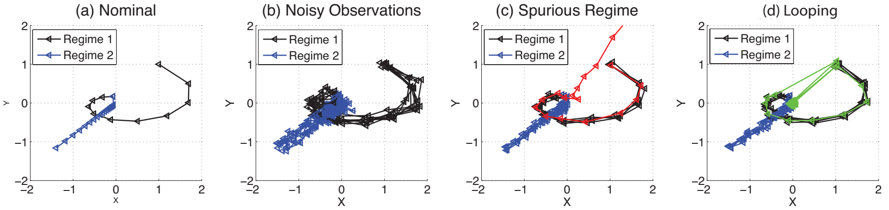


Fig. 3: (a) Observations from a dynamical system with two regimes, (b) Observations corrupted with Gaussian Noise, (c) Observations corrupted with a spurious inserted regime (red), and (d) Observations corrupted with an inserted loop(green).

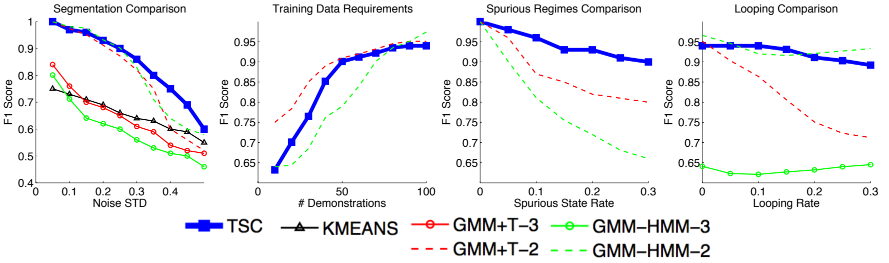


Fig. 4: Accuracy as a function of noise: TSC, K-Means, GMM+T (GMM with time), GMM+HMM (GMM with HMM). (a) The Dirichlet Process used in TSC reduces sensitivity to parameter choice and is comparable to GMM techniques using the optimal parameter choice, (b) HMM based approaches require more training data as they have to learn transitions, (c) TSC clusters are robust to spurious regimes, and (d) TSC clusters are robust to loops–without having to know the regimes in advance.

$\delta = 0$ (merge all repeated transitions), and $\rho = 80\%$ (prune all regimes representing less than 80% of the demonstrations).

First, we generate 100 noisy observations (additive zero mean Gaussian noise) from the system without loops or spurious states–effectively only measuring the DP-GMM versus the alternatives. Figure 4a shows the F1-score as a function of the noise in the observations. Initially, for an appropriate parameter choice $k = 2$ both of the GMM-based methods perform well and at low noise levels the DP-GMM used by our work mirrors this performance. However, if the parameter is set to be $k = 3$, we see that the performance significantly degrades. $k = 1$ corresponds to a single segment which has a F1 score of 0.4 on all figures. The DP-GMM mitigates this sensitivity to the choice of parameter by automatically setting the value. Furthermore, as the noise increases, the 80% pruning of DP-GMM mitigates the effect of outliers leading to improved accuracy.

In Figure 4b, we look at the accuracy of each technique as a function of the number of demonstrations. GMM+HMM has more parameters to learn and therefore requires more data. GMM+T converges the fastest, TSC requires slightly more data, and the GMM+HMM requires the most.

In Figure 4c, we corrupt the observations with spurious dynamical regimes. These are random transition matrices which replace one of the two dynamical regimes. We vary the rate at which we randomly corrupt the data, and measure the performance of the different segmentation techniques as a function of this rate. Due to the pruning, TSC gives the most accurate segmentation. The Dirichlet process groups the random transitions in different clusters and the small clusters are pruned out. On the other hand, the pure GMM techniques are less accurate since they are looking for exactly two regimes.

In Figure 4d, introduce corruption due to loops and compare the different techniques. A loop is a step that returns to the start of the regime randomly, and we vary this random rate. For an accurately chosen parameter $k = 2$, for the GMM-HMM, it gives the most accurate segmentation. However, when this parameter is set poorly $k = 3$, the accuracy is significantly reduced. On the other hand, using time as a GMM feature (GMM+T) does not work since it does not know how to group loops into the same regime.

5.2 Surgical Experiments: Evaluation Tasks

We describe the three tasks used in our evaluation, and show manually segmented versions in Figure 5. This will serve as ground truth when qualitatively evaluating our segmentation on real data.

Circle Cutting: In this task, we have a 5cm diameter circle drawn on a piece of gauze. The first step is to cut a notch into the circle. The second step is to cut clockwise. Next, the robot transitions to the other side cutting counter clockwise. Finally, the robot finishes the cut at the meeting point of the two incisions. As the left arm's only action is maintain the gauze in tension, we exclude it from the analysis. In Figure 5a, we mark 6 manually identified transitions points for this task from [24]: (1) start, (2) notch, (3) finish 1st cut, (4) cross-over, (5) finish 2nd cut, and (6) connect the two cuts. For the circle cutting task, we collected 10 demonstrations by non-experts familiar with operating the da Vinci Research Kit (dVRK).

We apply our method to the JIGSAWS dataset[8] consisting of surgical activity for human motion modeling. The dataset was captured using the da Vinci Surgical System from eight surgeons with different levels of skill performing five repetitions each of

Needle Passing: We applied our framework to 28 demonstrations of the needle passing task. The robot passes a needle through a loop using its right arm, then its left arm to pull the needle through the loop. Then, the robot hands the needle off from the left arm to the right arm. This is repeated four times as illustrated with a manual segmentation in Figure 5b.

Suturing: Next, we explored 39 examples of a 4 throw suturing task (Figure 5c). Using the right arm, the first step is to penetrate one of the points on right side. The next step is to force the needle through the phantom to the other side. Using the left arm, the robot pulls the needle out of the phantom, and then hands it off to the right arm for the next point.

5.3 Experiment 2. Pruning and Compaction

In Figure 6, we highlight the benefit of pruning and compaction using the Suturing task as exemplar. First, we show the transition states without applying the compaction step to remove looping transition states (Figure 6a). We find that there are many more transition states at the "insert" step of the task. Compaction removes the segments that correspond to a loop of the insertions. Next, we show the all of the clusters found by DP-GMM. The centroids of these clusters are marked in Figure

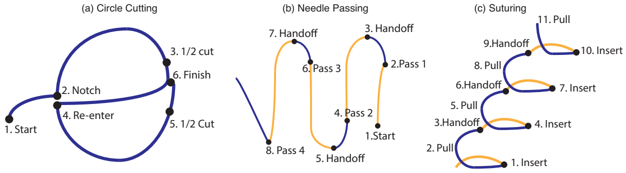
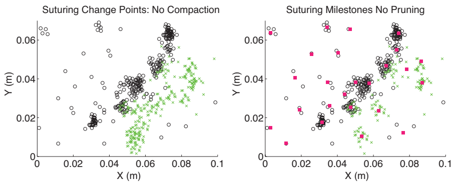


Fig. 5: Hand annotations of the three tasks: (a) circle cutting, (b) needle passing, and (c) suturing. Right arm actions are listed in dark blue and left arm actions are listed in yellow.

6b. Many of these clusters are small containing only a few transition states. This is why we created the heuristic to prune clusters that do not have transition states from at least 80% of the demonstrations. In all, 11 clusters are pruned by this rule.

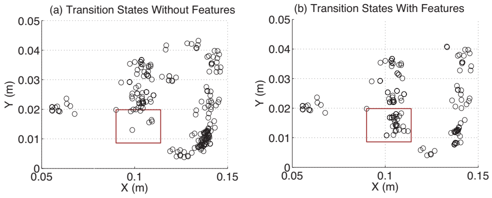
Fig. 6: We first show the transition states without compaction (in black and green), and then show the clusters without pruning (in red). Compaction sparsifies the transition states and pruning significantly reduces the number of clusters.



5.4 Experiment 3. Can Vision Help?

In the next experiment, we evaluate TSC in a featurized state space that incorporates states derived from vision (Described in Section 5.1). We illustrate the transition states in Figure 7 with and without visual features on the circle cutting task. At each point where the model transitions, we mark the end-effector (x,y,z) location. In particular, we show a region (red box) to highlight the benefits of these features. During the cross-over phase of the task, the robot has to re-enter the notch point and adjust to cut the other half of the circle. When only using the end-effector position, the locations where this transition happens is unreliable as operators may approach the entry from slightly different angles. On the other hand, the use of a gripper contact binary feature clusters the transition states around the point at which the gripper is in position and ready to begin cutting again. In the subsequent experiments, we use the same two visual features.

Fig. 7: (a) We show the transition states without visual features, (b) and with visual features. Marked in the red box is a set of transitions that cannot always be detected from kinematics alone.



5.5 Experiment 4. TSC Evaluation

Circle Cutting: Figure 8a shows the transition states obtained from our algorithm. And Figure 8b shows the TSC clusters learned (numbered by time interval midpoint). The algorithm found 8 clusters, one of which was pruned out using our $\rho = 80\%$ threshold rule.

The remaining 7 clusters correspond well to the manually identified transition points. It is worth noting that there is one extra cluster (marked $2'$), that does not correspond to a transition in the manual segmentation. At $2'$, the operator finishes a notch and begins to cut. While at a logical level notching and cutting are both penetration actions, they correspond to two different linear transition regimes due to the positioning of the end-effector. Thus, TSC separates them into different clusters even though a human annotator may not do so.

Needle Passing: In Figure 9a, we plot the transition states in (x,y,z) end-effector space for both arms. We find that these transition states correspond well to the logical segments of the task (Figure 5b). These demonstrations are noisier than the circle cutting demonstrations and there are more outliers. The subsequent clustering finds 9 (2 pruned). Next, Figures 9b-c illustrate the TSC clusters. We find that again TSC learns a small parametrization for the task structure with the clusters corresponding well to the manual segments. However, in this case, the noise does lead to a spurious cluster (4 marked in green). One possible explanation is that the two middle loops are in close proximity and demonstrations contain many adjustments to avoid colliding with the loop and the other arm while passing the needle through leading to numerous transition states in that location.

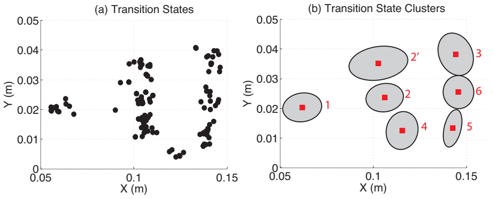


Fig. 8: (a) The transition states for the circle cutting task are marked in black. (b) The TSC clusters, which are clusters of the transition states, are illustrated with their 75% confidence ellipsoid.

Suturing: In Figure 10, we show the transition states and clusters for the suturing task. As before, we mark the left arm in orange and the right arm in blue. This task was far more challenging than the previous tasks as the demonstrations were inconsistent. These inconsistencies were in the way the suture is pulled after insertion (some pull to the left, some to the right, etc.), leading to transition states all over the state space. Furthermore, there were numerous demonstrations with looping behaviors for the left arm. In fact, the DP-GMM method gives us 23 clusters, 11 of which represent less than 80% of the demonstrations and thus are pruned (we illustrate the effect of the pruning in the next section). In the early stages of the task, the clusters clearly correspond to the manually segmented transitions. As the task progresses, we see that some of the later clusters do not. This is likely due to an error accumulation, where actions that were slightly different at the start became increasingly varied at the end.

5.6 *Experiment 5. Comparison to "Surgemes"*

Surgical demonstrations have an established set of primitives called surgemes, and we evaluate if segments discovered by our approach correspond to surgemes. In Table 1, we compare the number of TSC segments for needle passing and suturing to the number of annotated surgeme segments. A key difference between our segmentation and number of annotated surgemes is our compaction and pruning steps.

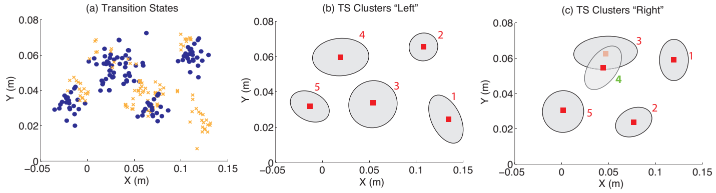


Fig. 9: (a) The transition states for the task are marked in orange (left arm) and blue (right arm). (b-c) The TSC clusters, which are clusters of the transition states, are illustrated with their 75% confidence ellipsoid for both arms

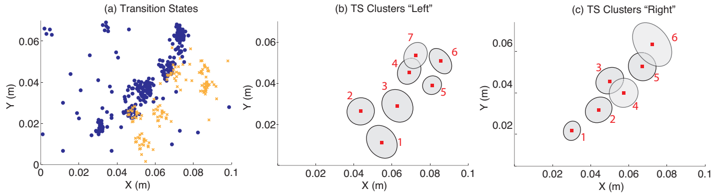


Fig. 10: (a) The transition states for the task are marked in orange (left arm) and blue (right arm). (b-c) The clusters, which are clusters of the transition states, are illustrated with their 75% confidence ellipsoid for both arms

Table 1: Surgemes specify primitives at a finer-scale than TSC. Nonetheless, after compaction and pruning, 83% and 73% of transition clusters for needle passing and suturing respectively contained exactly one surgeme transition.

	No. of Surgeme Segments	No. of Segments + C/P	No. of TSC	TSC-Surgeme	Surgeme-TSC
Needle Passing	19.3 ± 3.2	14.4 ± 2.57	11	83%	74%
Suturing	20.3 ± 3.5	15.9 ± 3.11	13	73%	66%

To account for this, we first select a set of surgemes that are expressed in most demonstrations (i.e., simulating pruning), and we also apply a compaction step to the surgeme segments. In case of consecutive appearances of these surgemes, we only keep the 1 instance of each for compaction. We explore two metrics: **TSC-Surgeme** the fraction of TSC clusters with only one surgeme switch (averaged over all demonstrations), and **Surgeme-TSC** the fraction of surgeme switches that fall inside exactly one TSC clusters.

6 Conclusion and Future Work

TSC models a set of demonstrations as linear dynamical system motions that transition, i.e., switch between linear systems, when they enter ellipsoidal regions of the state space called *transition state clusters*. To learn these clusters, TSC uses a hierarchical application of Dirichlet Process Gaussian Mixture Models (DP-GMM) with a series of merging and pruning steps. Our results on a synthetic example suggest that the hierarchical clusters are more robust to looping and noise, which are prevalent in surgical data. We further applied our algorithm to three surgical datasets and found that the transition state clusters correspond well to hand annotations and transitions w.r.t motions from a pre-defined surgical motion vocabulary called surgemes. We believe that the growing maturing of Convolutional Neural Networks can facilitate transition state clustering directly from raw data (e.g., pixels), as opposed to the features studied in this work, and is a promising avenue for future work.

Acknowledgments: This research was supported in part by a seed grant from the UC Berkeley Center for Information Technology in the Interest of Society (CITRIS), by the U.S. National Science Foundation under Award IIS-1227536: Multilateral Manipulation by Human-Robot Collaborative Systems. This work has been supported in part by funding from Google and and Cisco. We also thank Florian Pokorny, Jeff Mahler, and Michael Laskey.

References

[1] Asfour, T., Gyarfas, F., Azad, P., Dillmann, R.: Imitation learning of dual-arm manipulation tasks in humanoid robots. In: Humanoid Robots, 2006 6th IEEE-RAS International Conference on, pp. 40–47 (2006)

[2] Calinon, S.: Skills learning in robots by interaction with users and environment. In: Ubiquitous Robots and Ambient Intelligence (URAI), 2014 11th International Conference on, pp. 161–162. IEEE (2014)

[3] Calinon, S., Billard, A.: Stochastic gesture production and recognition model for a humanoid robot. In: Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, vol. 3, pp. 2769–2774 vol.3 (2004)

[4] Calinon, S., Bruno, D., Caldwell, D.G.: A task-parameterized probabilistic model with minimal intervention control. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on, pp. 3339–3344 (2014)

[5] Calinon, S., D'halluin, F., Sauser, E.L., Caldwell, D.G., Billard, A.G.: Learning and reproduction of gestures by imitation. Robotics & Automation Magazine, IEEE **17**(2), 44–54 (2010)

[6] Calinon, S., Halluin, F.D., Caldwell, D.G., Billard, A.G.: Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework. In: Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on, pp. 582–588. IEEE (2009)

[7] Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 2790–2797. IEEE (2009)

[8] Gao, Y., Vedula, S., Reiley, C., Ahmidi, N., Varadarajan, B., Lin, H., Tao, L., Zappella, L., Bejar, B., Yuh, D., Chen, C., Vidal, R., Khudanpur, S., Hager, G.: The jhu-isi gesture and skill assessment dataset (jigsaws): A surgical activity working set for human motion modeling. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI) (2014)

[9] Grollman, D.H., Jenkins, O.C.: Incremental learning of subtasks from unsegmented demonstration. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pp. 261–266. IEEE (2010)

[10] Ijspeert, A., Nakanishi, J., Schaal, S.: Learning attractor landscapes for learning motor primitives. In: Neural Information Processing Systems (NIPS), pp. 1523–1530 (2002)

[11] Intuitive Surgical: Annual report 2014. URL http://investor.intuitivesurgical.com/phoenix.zhtml?c=122359&p=irol-IRHome

[12] Kehoe, B., Kahn, G., Mahler, J., Kim, J., Lee, A., Lee, A., Nakagawa, K., Patil, S., Boyd, W., Abbeel, P., Goldberg, K.: Autonomous multilateral debridement with the raven surgical robot. In: Int. Conf. on Robotics and Automation (ICRA) (2014)

[13] Keogh, E.J., Pazzani, M.J.: Derivative dynamic time warping. SIAM

[14] Kruger, V., Herzog, D., Baby, S., Ude, A., Kragic, D.: Learning actions from observations. Robotics & Automation Magazine, IEEE **17**(2), 30–43 (2010)

[15] Krüger, V., Tikhanoff, V., Natale, L., Sandini, G.: Imitation learning of non-linear point-to-point robot motions using dirichlet processes. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on, pp. 2029–2034. IEEE (2012)

[16] Kulić, D., Nakamura, Y.: Scaffolding on-line segmentation of full body human motion patterns. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, pp. 2860–2866. IEEE (2008)

[17] Kurihara, K., Welling, M., Vlassis, N.A.: Accelerated variational dirichlet process mixtures. In: Advances in Neural Information Processing Systems, pp. 761–768 (2006)

[18] Lea, C., Hager, G.D., Vidal, R.: An improved model for segmentation and recognition of fine-grained activities with application to surgical training tasks. In: WACV (2015)

[19] Lee, S.H., Suh, I.H., Calinon, S., Johansson, R.: Autonomous framework for segmenting robot trajectories of manipulation task. Autonomous Robots **38**(2), 107–141

[20] Lin, H., Shafran, I., Murphy, T., Okamura, A., Yuh, D., Hager, G.: Automatic detection and segmentation of robot-assisted surgical motions. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 802–810. Springer (2005)

[21] Mahler, J., Krishnan, S., Laskey, M., Sen, S., Murali, A., Kehoe, B., Patil, S., Wang, J., Franklin, M., Abbeel, P., K., G.: Learning accurate kinematic control of cable-driven surgical robots using data cleaning and gaussian process regression. In: Int. Conf. on Automated Sciences and Engineering (CASE), pp. 532–539 (2014)

[22] Manschitz, S., Kober, J., Gienger, M., Peters, J.: Learning movement primitive attractor goals and sequential skills from kinesthetic demonstrations. Robotics and Autonomous Systems (2015)

[23] Moldovan, T., Levine, S., Jordan, M., Abbeel, P.: Optimism-driven exploration for nonlinear systems. In: Int. Conf. on Robotics and Automation (ICRA) (2015)

[24] Murali, A., Sen, S., Kehoe, B., Garg, A., McFarland, S., Patil, S., Boyd, W., Lim, S., Abbeel, P., Goldberg, K.: Learning by observation for surgical subtasks: Multilateral cutting of 3d viscoelastic and 2d orthotropic tissue phantoms. In: Int. Conf. on Robotics and Automation (ICRA) (2015)

[25] Niekum, S., Osentoski, S., Konidaris, G., Barto, A.: Learning and generalization of complex tasks from unstructured demonstrations. In: Int. Conf. on Intelligent Robots and Systems (IROS), pp. 5239–5246. IEEE (2012)

[26] Pastor, P., Hoffmann, H., Asfour, T., Schaal, S.: Learning and generalization of motor skills by learning from demonstration. In: Int. Conf. on Robotics and Automation (ICRA), pp. 763–768. IEEE (2009)

[27] Quellec, G., Lamard, M., Cochener, B., Cazuguel, G.: Real-time segmentation and recognition of surgical tasks in cataract surgery videos. Medical Imaging, IEEE Transactions on **33**(12), 2352–2360 (2014)

[28] Reiley, C.E., Plaku, E., Hager, G.D.: 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, pp. 967–970. IEEE (2010)

[29] Rosen, J., Brown, J.D., Chang, L., Sinanan, M.N., Hannaford, B.: Generalized approach for modeling minimally invasive surgery as a stochastic process using a discrete markov model. Biomedical Engineering, IEEE Transactions on **53**(3), 399–413 (2006)

[30] Tang, H., Hasegawa-Johnson, M., Huang, T.S.: Toward robust learning of the gaussian mixture state emission densities for hidden markov models. In: Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, pp. 5242–5245. IEEE (2010)

[31] Tao, L., Zappella, L., Hager, G.D., Vidal, R.: Surgical gesture segmentation and recognition. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013, pp. 339–346. Springer (2013)

[32] Vakanski, A., Mantegh, I., Irish, A., Janabi-Sharifi, F.: Trajectory learning for robot programming by demonstration using hidden markov model and dynamic time warping. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **42**(4), 1039–1052 (2012)

[33] Varadarajan, B., Reiley, C., Lin, H., Khudanpur, S., Hager, G.: Data-derived models for segmentation with application to surgical assessment and training. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 426–434. Springer (2009)

[34] Willsky, A.S., Sudderth, E.B., Jordan, M.I., Fox, E.B.: Sharing features among dynamical systems with beta processes. In: Advances in Neural Information Processing Systems, pp. 549–557 (2009)

[35] Zappella, L., Bejar, B., Hager, G., Vidal, R.: Surgical gesture classification from video and kinematic data. Medical image analysis **17**(7), 732–745 (2013)