Semantically Interpretable Predictive State Representation

Johannes A. Stork Carl Henrik Ek Danica Kragic

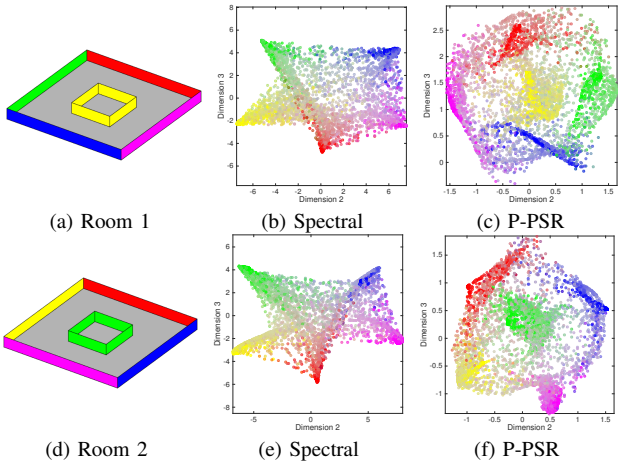
***Abstract*—Predictive State Representations (PSRs) allow modeling of dynamical systems directly in observables and without relying on latent variable representations. However, it is often hard to attribute semantic meaning to PSR representations. In this paper, we present the idea of introducing prior information to PSR learning (P-PSRs) in order to learn representations which are more suitable for generalization, planning, and interpretation. For this, we learn an embedding of test features such that belief points of similar semantic share the same region of a subspace. The resulting spacial relationship facilitates generalization and semantical interpretation of the representation. We demonstrate how our approach can handle biased training data and allows feature selection such that the resulting representation emphasizes observables that relate to the planning task. We show that our P-PSRs result in qualitatively meaningful representations and present quantitative results that indicate improved suitability for planning.**

I. INTRODUCTION

We can characterize intelligence as the capability to act appropriately in an uncertain environment. Humans achieve this by—among other things—reducing uncertainty through incorporation of previous experience and knowledge. Likewise, robotics as a field strives to create intelligent artificial agents that are capable of acting when the state of the world is uncertain. To accomplish this goal we need to investigate how to best utilize prior knowledge.

It has been argued that prior knowledge should consist of architectural constraints and fundamental truths such as physical laws. The remaining concepts—such as the world model or sensory-motor loops—must be *learned* by the agent with the help of priors.[1] A list of generic priors for representation learning in AI has been proposed by [1], while arguing that the success of machine learning algorithms generally depends on data representation. However, these generic priors have been qualified as too weak and stronger and more specific priors have been proposed for representation learning for robots [5], such as *causality* and *temporal coherence*.

However, in order to incorporate such priors we need to devise a representation of the environment, the current state of the robot, and the system dynamics. In many scenarios this task is very challenging. One appraoch designed to *learn* such representations is denoted Predictive State Representations (PSRs) [6]. PSRs maintain probability distributions over a set of future events conditioned on a history [6, 9]. This relation



(a) Room 1 (b) Spectral (c) P-PSR

(d) Room 2 (e) Spectral (f) P-PSR

Fig. 1. Learn representations to navigate rooms with colored walls from camera images: We visual compare results by the spectral approach (b)&(e) and our P-PSR (c)&(f). Execution traces are embedded as belief points and colored by the average color of the last seen camera image. Both times, the spectral approach's topology is arranged according to RGB-value similarity. In contrast, our approach captures the geometrical structure of the environment, arranging the features such that the topology of the *environment* is retained for the navigation task.

is formalized in the PSR's fundamental equation for the probability of a future event τ given the history h: $P(\tau|h) = P(o_{t+1}, \ldots, o_{t+n}|h||a_{t+1}, \ldots, a_{t+n}) = P(\tau^{\mathcal{O}}|h||\tau^{\mathcal{A}})$, where o_i is the observation after performing action a_i. The symbol $||$ denotes the agent's intervention or plan [7]. Directly representing a system in terms of observables (i.e. τ and h) allows for learning a grounded representation by exploration of the environment, observing change induced by actions. In case of PSRs two challenges need to be addressed: (a) Discovery of a sufficient set of future events, and (b) learning state update parameters. Most frequently, the discovery problem is addressed as subspace identification (transformed PSRs [8]). In terms of an AI agent, a (learned) state space can be seen simply as a means to perform a task. However, the *spectral* learning approach is ignorant of the way in which observables are relevant for the task and focuses on representing the maximum amount of variance in the training data. Naturally, this leads to underrepresentation of relevant information in the state if redundant, distracting, or unrelated data is observed as well. We therefore need to introduce priors and labels to the learning process to allow identification of the meaningful information.

Having conceived a PSR, plans can be formulated and executed directly in terms of observable quantities. This *closing of the loop* [2] was initially proposed as a motivation for using PSRs, because the state space could first be learned by exploration and subsequently used for planning.

In this paper, we argue learning the representation and

The authors are with the Computer Vision and Active Perception Lab, Centre for Autonomous Systems, School of Computer Science and Communication, KTH Royal Institute of Technology, Stockholm, Sweden, {jastork,chek,dani}@kth.se. This work was supported by FLEXBOT (FP7-ERC-279933), the Swedish Research Council and the Swedish Foundation for Strategic Research.

[1]Leslie P. Kaebling. Keynote Lecture: Robot Intelligence. AAAI Conference on Artificial Intelligence. Atlanta, Georgia, USA, July 2010. http://people.csail.mit.edu/lpk/AAAI10LPK.pdf

planning should not be considered completely independent, because different observable quantities are different importance dependent on the task. Considering for example a vision-based navigation task in a room (see Fig. 1), it is important that the PSR reflects the distances and spacial relations for optimal path planning instead of the similarity of the environments visual features. Therefore, we extend PSRs to exploit prior information about the task, called P-PSRs. Our approach is analogous to metric learning [13] where we pose geometric preferences on the learning problem. Our work brings priors into the field of subspace identification-based PSR learning and for the first time allows for informed constraints on the features that form the PSR state.

II. METHODOLOGY

Our goal is to learn a TPSR that is optimized for semantical interpretation of geometry, exploitation for geometry-based heuristics, and generalization for belief space planning. This is achieved when distances in the learned representation mirror the semantics in the true system or beliefs over them. To this end, we learn a test feature embedding subject to priors. Below, we formulate our learning problem as an optimization problem, describe the optimization parameters, the loss-functions, and the training data.

A. Projecting Features of Tests

Feature-based TPSR [2] represent state as vector of linear combinations of test feature expectations $\mathbf{m}$. This is defined in the TPSR state update equation below, where the transformation matrix, $\mathbf{Z} = \mathbf{J}\Phi^{\mathcal{T}}\mathbf{R}$, factorizes as a map of test probabilities, $\mathbf{R}$, a map from tests to features, $\Phi^{\mathcal{T}}$, and a map, $\mathbf{J}$, that defines a change of basis to the subspace of sufficient statistics.

$$\mathbf{Z}\mathbf{m}_{t+1} = \frac{\mathbf{Z}\mathbf{M}_{ao}\mathbf{Z}^{-1}\mathbf{Z}\mathbf{m}_t}{((\mathbf{Z}^{-1})^{\intercal}\mathbf{m}_{\infty})^{\intercal}\mathbf{Z}\mathbf{M}_{ao}\mathbf{Z}^{-1}\mathbf{Z}\mathbf{m}_t} \tag{1}$$

Usually, the number of test features $n_{\mathcal{T}}$ is large (e.g. several hundred) and more than sufficient for the actual lower-dimensional system dynamics. Therefore, a matrix $\mathbf{J}$ that projects to a low-dimensional embedding space $\mathcal{S}$ is used.

$$\mathbf{J}\colon \mathbb{R}^{n_{\mathcal{T}}} \to \mathbb{R}^d, \qquad \mathbf{J}\colon \phi^{\mathcal{T}} \mapsto \mathbf{s} \tag{2}$$

Consequently, $\mathbf{J}$ controls the *dimensionality* and *structure* of the embedding space and performs the task of feature selection. However, in recent TPSR literature this role of $\mathbf{J}$ is not considered to the full extend. Instead, $\mathbf{J}$ is chosen in a way that disregards embedding space structure and semantics as in [2, 4].

In contrast, we want to employ $\mathbf{J}$ to impose constraints or priors on the geometric structure of the embedding space $\mathcal{S}$ which is learned by the TPSR. Unfortunately, the TPSR equations require the existence of a pseudo inverse, which imposes a difficult optimization constraint. However, by redefining the TPSR subspace transform $\mathbf{Z} = \tilde{\mathbf{U}}^{\intercal}\tilde{\mathbf{J}}\Phi^{\mathcal{T}}\mathbf{R}$ we can instead of $\mathbf{J}$ consider a matrix in a similar role, $\tilde{\mathbf{J}}$, that allows for a proper pseudo inverse [3]. (Here, $\boldsymbol{P}_{\mathcal{T},\mathcal{H}}$ is an observable matrix, $\mathbf{R}$ maps from core tests to a large test set and $\Phi^{\mathcal{T}}$ maps those tests to features.) Now, we can modify $\tilde{\mathbf{J}}$ as long as $\tilde{\mathbf{J}}\boldsymbol{P}_{\mathcal{T},\mathcal{H}}$ has full rank to optimize the embedding space $\tilde{\mathcal{S}}$ generated by the mapping $\tilde{\mathbf{J}}\colon \phi^{\mathcal{T}} \mapsto \tilde{\mathbf{s}}$.

B. Optimization Problem

In order to learn the matrix $\tilde{\mathbf{J}}$ and thus also $\mathbf{J}$, we formulate an optimization problem over the matrix $\tilde{\mathbf{J}}$ of dimension $d \times n_{\mathcal{T}}$. Thereby denotes d the dimensionality of the subspace and $n_{\mathcal{T}}$ the number of features. To reduce the number of parameters, we parameterize $\tilde{\mathbf{J}}$ by the first m left singular vectors of $\boldsymbol{P}_{\mathcal{T},\mathcal{H}}$, denoted by $\mathbf{U}_{1:m}$.

$$\tilde{\mathbf{J}} = \mathbf{A}\mathbf{U}_{1:m}^{\intercal}, \qquad \mathbf{A} \in \mathbb{R}^{d\times m} \tag{3}$$

The optimization parameters are hence the $d \times m$ entries of the matrix $\mathbf{A}$. We impose our priors on the embedding space with the loss-functions L_i. Usually, loss-functions will operate on the image of test features under $\tilde{\mathbf{J}}$ which we denote by $\tilde{\mathbf{s}} = \mathbf{A}\mathbf{U}_{1:m}^{\intercal}\phi^{\mathcal{T}} = \tilde{\mathbf{J}}\phi^{\mathcal{T}}$. We therefore formulate an unconstrained optimization problem using L_i on pairs of states $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{s}}'$ and their respective labels ℓ and ℓ':

$$\underset{\mathbf{A}\in\mathbb{R}^{m\times d}}{\text{minimize}} \sum_i L_i(\delta_i(\tilde{\mathbf{s}}, \tilde{\mathbf{s}}'), \bar{\delta}_i(\ell, \ell'), \mathbf{A}) \tag{4}$$

Here δ_i and $\bar{\delta}_i$ are distance measures in state and label space respectively. Labels take the form of annotations endowed with a distance metric $\bar{\delta}_i$ which assists the underlying planning problem.

C. Priors as Loss-functions

In [5], five robotic priors are presented for learning a representation in a *fully observable* setting: *simplicity*, *temporal coherence*. *proportionality*, *causality*, and *repetability*. These priors are derived from the fact that the laws of physics govern the change of the world and the effects of the robot's actions. In this work, we consider the partially observable domain and learn a belief space representation. Therefore, not all of mentioned priors are applicable to be directly implemented as L_i. In addition to [5] we learn a representation including *label consistency*.

III. EXPERIMENTS

A. Different Rooms

In this experiment, we compare the learned representation of the spectral learning approach [2] and our prior-based approach by projecting test sequence features into the model space $\mathcal{S}$. The application is a challenging synthetic robot navigation task similar to the one used for representation learning in [2, 5, 10]. In specific, the robot needs to learn a representation of a room by observing images from a simulated camera in an egocentric view.

Experiment domain: A robot moves in a 45×45 unit large room with colored 4 unit high walls (see Fig. 1). Six actions are possible as the product of 0 or 1 unit forward translation and rotation by -15, 0, and +15 degrees. In case where the robot contacts with a wall, it stops, mimicking an

elastic collision. Note that the agent only perceives 10-by-10 pixel RGB images of a simulated forward-looking camera with 45 degrees view angle as depicted in Fig. 2. From a single observation it is not possible to discriminate the location in the room. This means that in order to discover its state, the robot needs to integrate observations over time. The training data consists of 5000 randomly sampled execution traces of 7 random actions. We use an RBF kernel applied to a "stacked" image (3×300 dimensions) with 2000 basis functions and 500 observation kernel centers. The projection matrix $\mathbf{J}$ is learned with $d = 5$ and $m = 20$ using position and orientation of the robot as labels. This results in a dimensionality reduction from 2000 to 5.

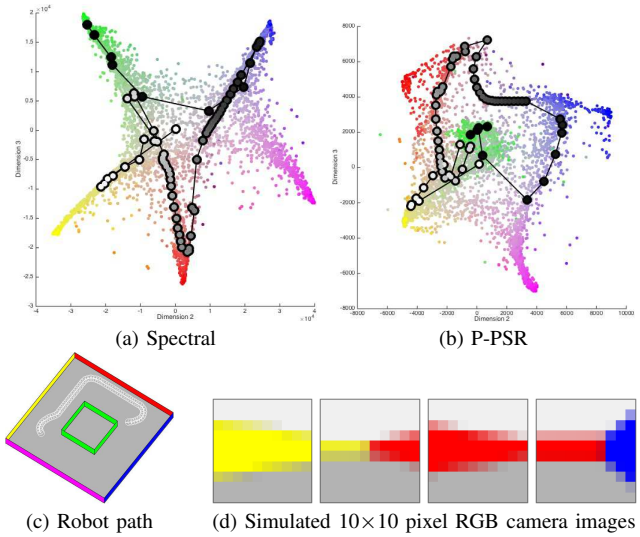
The motivation to perform this experiment comes from [2] where the feature-based spectral TPSR was introduced. It is not trivial to manually define an observation-to-state mapping for this problem. Since the observations are high-dimensional a feature-based representation is required. As such, it is an ideal example to show the benefits of TPSRs.

It has been described that the topology of the agent's visual *environment* gets represented in the low-dimensional embedding [2] (see Fig. 1(b)). However, in the spectral approach there is nothing explicitly encouraging this. Rather, the structure of the embedding comes from the structure of the observation space, i.e. the structure of the RGB space. The discover of the room's topology is purely coincidental by the fact that the ordering of the color of the walls coincides with their distance in RGB space. To show this, we altered the layout of the room and exchanged the *blue* with the *magenta* and the *green* with the *yellow* wall (see Fig. 1 (a) & (d)). As can be seen from the resulting embedding in Fig. 1, the representation learned by the spectral approach is very similar for the two rooms. The *magenta* wall is placed between the *blue* and the *red* wall for both rooms. This is because *magenta* is, in RGB space, a mix of *red* and *blue* and the representation has very little to do with the configuration of the walls. Differently from the spectral method, our approach discovers the underlying structure of the room—due to the priors—and the two embeddings retain the different structure. Further, our representation is capable of placing the inner-wall of the room equally distant from the outer walls reflecting the true topology of the room in a geometric representation. We argue that the representation from our approach is better suited for planning purposes as it reflects the structure of the problem domain.

B. Belief Space

In the previous section, we inspected the feature embedding space $\mathcal{S}$, which is the image space for test features $\phi^{\mathcal{T}}$. In this section, we investigate the geometric properties of a filtered trace of actions and observations in the PSR belief space $\mathcal{B}$. The PSR belief space is related to the test feature embedding space since the first represents expectations of test features and the latter defines the structure of test features.

1) Tracking a long path: Embedding a long history allows us to understand connectivity in the learned representation. We track the PSR belief state of the trace depicted in Fig. 2(c), starting next to the yellow wall. The tracking result can be seen in Fig. 2(a)&(b).



(a) Spectral (b) P-PSR

(c) Robot path (d) Simulated 10×10 pixel RGB camera images

Fig. 2. We embed (track) a set of different robot paths in the virtual environment with simulated camera input and color the belief points by the average color of the last camera image ((a) & (b)). For the long example path (c), the robot needs to derive its state from raw image input only, starting at the position next to the yellow wall. The corresponding trace in the PSR space $\mathcal{B}$ is colored from white to black from start to end and shown in (a) & (b).

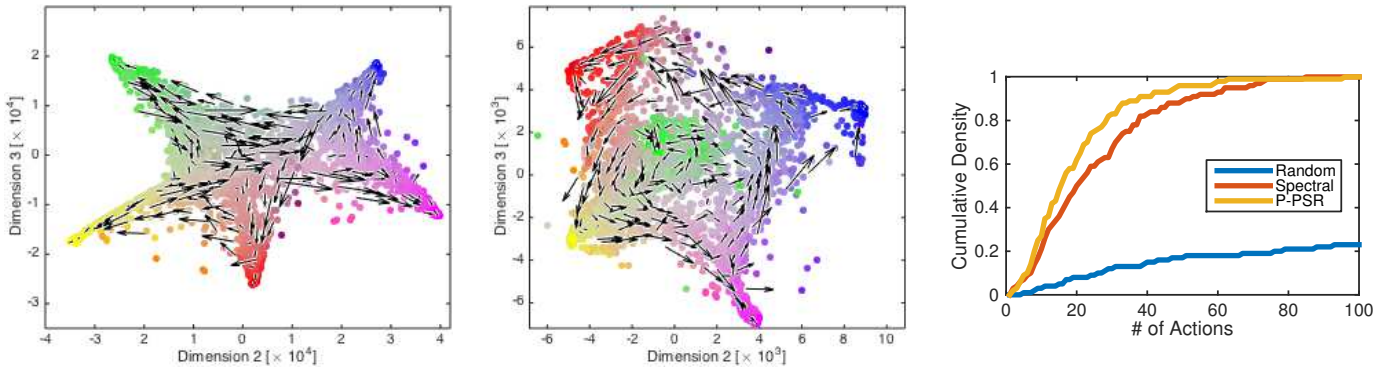
In the case of the representation learned by the spectral approach, the trace starts in the *gray* area, describing that the average initial state *sees* mostly *gray* pixels. After that the state moves to *yellow*, *green* and *red*. When the robot turns from the *red* wall to the *blue* wall, the trace needs to travel to the opposite side of the PSR belief space to represent states which see mostly *blue* pixels.

In our representation, the initial average state sees a mixture between *green* and *gray* pixels, which relates to the fact that the central obstacle and the floor are visible in about half of the possible robot poses. The trace then travels towards *green* and later to *red* and *blue* in small and regular steps. When the robot turns towards the *blue* wall, the sensor input changes rapidly to *blue* pixels and a larger step can be seen in the belief space. The same phenomena occurs when the robot finally turns towards central obstacle.

Comparing the two different embeddings we argue that the P-PSR belief space is more suitable for planning as it is more temporally coherent not showing such drastic "jumps" as the spectral method.

2) Representation of actions: For planning or policy learning it can be advantageous when the same action has similar effect in neighboring states because it allows to generalize. In Fig. 3 we show the results of applying an action which turns the robot counter-clock wise and moves it forward.

We can observe that in the spectral-based representation all actions are correctly represented, e.g. the states which *see* the *magenta* wall turn towards the *blue* wall. However, the motion of belief points from *yellow* to *magenta* overlaps with the motion from *blue* to *red*. Additionally, we can observe



(a) Spectral (b) P-PSR (c) Analysis of planning performance

Fig. 3. *Left and middle:* Representation of actions in the PSR belief space. Arrows show the motion of belief points in $\mathcal{B}$ for the move forward and turn left action ((a) and (b)). In the spectral model similar belief points move in different directions, which implies different reward. In the P-PSR model belief point a similar region move in a homogenous manner. *Right:* Analysis from policy execution from 100 sampled starting positions to a target image depicting the blue wall in the two learned models and my a model-free agent. The agent using the P-PSR model has improved performance by using shorter paths.

that each of the "corners" exhibits a simultaneous in- and outward motion not far apart. This is semantically correct and relates to the rotational action, however it impairs belief space generalization.

Different from that, the P-PSR representation displays a circular motion of belief such that neighboring states are transformed in a similar manner.

C. Planning in the learned model

Finally, we quantitatively compare how pertaining the learned representations are to subsequent reinforcement learning as this is the main motivation for representation learning. For this, we learn a policy to face the *blue* wall and describe the goal states as seeing an average color with an RGB space distance less than 0.2. The belief points for value iteration are generated by tracking 5000 sample traces for 3 steps. Positive reward (+100) is defined for goal states and wall collisions are penalized (reward -1000). Additionally, we set -100 reward for states that directly face a different wall. As a measure of performance, we compare the number of steps to reach the goal for 100 randomly sampled starting states as in [2]. Less steps indicate a better performance. Note that the robot never has access to pose, only images.

In Fig. 3(c) we can see that the baseline agent that performs random actions results in much longer action sequences than both of the model-based agents. The spectral model results in an substantial improvement over the baseline (as also reported by [2]) and the P-PSR agent again improves upon this result. Importantly reducing its uncertainty more rapidly early in the execution. We assume that this improvement is due to a better representation since the policy learning algorithm exploits belief space similarity for generalization.

D. Biased Sample Set

When learning representations for robotic systems that interact with the world collecting training examples can be very time consuming, expensive, or at least tedious. Additionally, it can be difficult to collect an unbiased set of examples or to design an automatic exploration policy that generates a uniformly distributed and representative set.

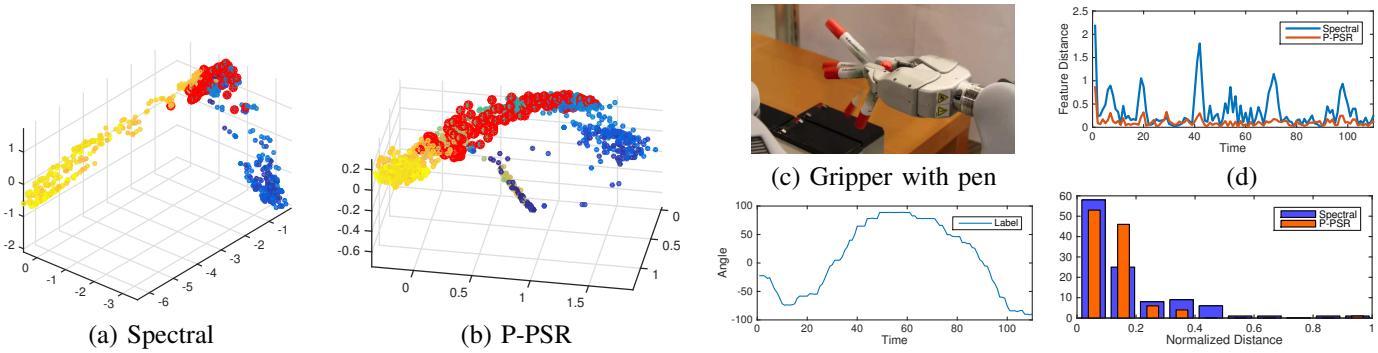
In this experiment, we consider data from the robotic manipulation domain of our previous work [11]. The task is to change the orientation of a grasped pen by pushing it against a table edge in different ways as illustrated in Fig. 4(c). The robot can only access pressure readings from its fingertips and the two most extreme regions of angles (+90 and -90 degrees) have identical readings. In all positions of the pen, at least one of the six actions can affect its pose, but when the pen points straight forward, all push actions result in change. This generates a bias in the sample set in such that certain pen angles are less frequently represented.

We learn a 5 dimensional representation with $m = 20$ and all our objective functions using the visually obtained pen angles labels. For the kernelization we use 342 basis function and 12 observation kernel centers. The 1312 training examples are obtained with the suffix algorithm [12].

As we can observe in Fig. 4(a), the spectral method places many different pen states (angles of -38 to +54 degrees) into the region indicated by red color. In contrast, the P-PSR spreads out all angles evenly. The small line in the center represents all traces which remain indistinguishable because they never leave the region where the observations are aliased. In accordance with the priors, these traces are places in the middle between the states to which they are most similar.

IV. CONCLUSIONS

We have presented an approach that extends PSRs to learn representations that are meaningful for planning. Our approach incorporates prior information about the planning task during the learning phase encouraging the model to retain information that are relevant for the task. We presented experimental results showing the benefits of our approach using quantitative and qualitative measures. In specific, we show how our approach is capable of learning in scenarios where the training dataset is biased and when only a subset of the variations in the observation features are relevant. The generality of our



(a) Spectral (b) P-PSR (c) Gripper with pen (d) (e) (f)

Fig. 4. Learning a representation for rotating a grasped pen with tactile feedback: The learned representation using (a) the spectral and (b) P-PSR method. Colors *yellow* to *blue indicate the angel of the pen.* Red *indicates the angles between -38 and + 54 degrees of the pen. The spectral model represents many different angles at the same place* (red *color) while the P-PSR model spreads out all angles evenly. When tracking a the manipulation with angles shown in (e), the P-PSR representation is more continuous (more smooth) as compared to the spectral model in terms of feature distance (d)&(f).*

framework allows for future development including additional priors thereby significantly extending the domains where PSRs are applicable.

References

[1] Y. Bengio, A. Courville, and P. Vincent. "Representation learning: A review and new perspectives". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.8 (2013), pp. 1798–1828.

[2] B. Boots, S. Siddiqi, and G. Gordon. "Closing the Learning Planning Loop with Predictive State Representations". In: *IJRR* 30 (2011), pp. 954–956.

[3] W. Hamilton, M. M. Fard, and J. Pineau. "Efficient Learning and Planning with Compressed Predictive States". In: *Journal of Machine Learning Research* 15 (2014), pp. 3395–3439.

[4] W. L. Hamilton, M. M. Fard, and J. Pineau. "Modelling sparse dynamical systems with compressed predictive state representations". In: *ICML*. 2013, pp. 178–186.

[5] R. Jonschkowski and O. Brock. "State Representation Learning in Robotics: Using Prior Knowledge about Physical Interaction". In: *RSS*. Berkeley, USA, 2014.

[6] M. Littman, R. Sutton, and S. Singh. "Predictive Representations of State". In: *NIPS*. 2002, 1555–1561.

[7] J. Pearl. *Causality: models, reasoning and inference*. Vol. 29. Cambridge University Press, 2000.

[8] M. Rosencrantz, G. Gordon, and S. Thrun. "Learning low dimensional predictive representations". In: *ICML*. ACM. 2004, p. 88.

[9] S. Singh, M. R. James, and M. R. Rudary. "Predictive state representations: A new theory for modeling dynamical systems". In: *Conference on Uncertainty in artificial intelligence*. AUAI Press. 2004, pp. 512–519.

[10] N. Sprague. "Predictive Projections." In: *21st International Joint Conference on Artificial Intelligence (IJCAI)*. 2009, pp. 1223–1229.

[11] J. Stork, C. Ek, Y. Bekiroglu, and D. K. "Learning Predictive State Representation for In-Hand Manipulation". In: *IEEE ICRA*. Seattle, USA, 2015.

[12] B. Wolfe, M. R. James, and S. Singh. "Learning predictive state representations in dynamical systems without reset". In: *ICML*. ACM. 2005, pp. 980–987.

[13] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell. "Distance Metric Learning with Application to Clustering with Side-Information." In: *Advances in Neural Information Processing Systems*. 2002, pp. 505–512.