

# Learning Geometry from Sensorimotor Experience

Jeremy Stober and Risto Miikkulainen  
Department of Computer Science  
The University of Texas at Austin  
{stober,risto}@cs.utexas.edu

Benjamin Kuipers  
Computer Science and Engineering  
University of Michigan  
kuipers@umich.edu

**Abstract**—A baby experiencing the world for the first time faces a considerable challenge sorting through what William James called the “blooming, buzzing confusion” of the senses [1]. With the increasing capacity of modern sensors and the complexity of modern robot bodies, a robot in an unknown or unfamiliar body faces a similar and equally daunting challenge.

In order to connect raw sensory experience to cognitive function, an agent needs to decrease the dimensionality of sensory signals. In this paper a new approach to dimensionality reduction called *sensorimotor embedding* is presented, allowing an agent to extract spatial and geometric information from raw sensorimotor experience.

This approach is evaluated by learning the geometry of GRIDWORLD and ROVINGEYE robot domains. The results show that sensorimotor embedding provides a better mechanism for extracting geometric information from sensorimotor experience than standard dimensionality reduction methods.

## I. INTRODUCTION

In the early stages of development infants form ego-centric models of the world, which then serve as a basis for learning more advanced concepts. A robot waking up in an unfamiliar body faces a similar challenge, and acquiring an ego-centric model that includes details of sensor, space, and object geometry would facilitate learning more advanced concepts. One immediate barrier to acquiring geometric knowledge is the real-time, high-dimensional nature of uninterpreted sensorimotor signals, which poses a real challenge for existing state-of-the-art manifold learning and dimensionality reduction methods.

With this in mind consider the following fundamental problem for developmental robotics, called the *sensorimotor geometry problem* in this paper.

*Design a developmental process that, for any roughly humanoid robot, starting only with a basic set of sensor primitives and motor reflexes, progresses through a period of sensorimotor development that results in knowledge of body, sensor and object location and geometry.*

Modern approaches for reducing data dimension applied to sensory data alone do not take advantage of the power of interaction between an agent and the environment. This interaction between an agent and its environment provides a rich source of sensor and motor data that allows for the more powerful dimensionality reduction approach presented in this paper.

## II. RELATED WORK

There are a number of related approaches to extracting structure from raw sensory experience, including several previous attempts to solve the sensorimotor geometry problem introduced in the previous section. For instance, Pierce and Kuipers used dimensionality reduction methods to learn geometry of robot sensors and motors [2]. This work was later expanded on in a number of papers based on recent advancements in dimensionality reduction and manifold learning. In particular, Philipona et al. developed a sensorimotor approach using embodied ISOMAP for learning spatial structure from uninterpreted sensors and effectors [3], and Bowling et al. developed the action respecting embedding (ARE) method while working towards a solution to the problem of *subjective localization* (solving SLAM problems without sensor and action models) [4].

These methods fall short of solving the sensorimotor geometry problem. For example, dimensionality reduction alone may not be able to learn spatial and geometric concepts beyond sensor organization because these methods are sensitive to, but do not take advantage of, the policy used to control the agent during data collection (Stober et al. [5] provide examples of problematic policies). Advanced approaches such as ARE are based on maximum-variance unfolding (MVU), which does not scale well on large datasets.

Stober et al. [6] demonstrated that sensor structure for foveated sensors can be learned from sensorimotor experience through careful analysis of *saccade* policies.<sup>1</sup> The key insight was that careful analysis of learned policies can make implicit geometric knowledge about sensor structure explicit. However, that work was specific to discovering sensor structure. This paper is motivated by the same insight, but extends it to a larger class of geometric discovery and dimensionality reduction problems.

## III. ACQUIRING GEOMETRIC KNOWLEDGE

An agent undergoing sensorimotor development needs to learn to represent certain geometric concepts in order to solve the sensorimotor geometry problem and progress through later stages of development. Geometry describes the relative arrangement of objects or constituent parts. To learn geometry,

<sup>1</sup>Foveated sensors are non-uniform arrays of sensor elements that have a high density fovea and a low density periphery. Saccades are ballistic sensor motions.

an algorithm should take in sensorimotor data and produce a set of low dimensional points whose relative arrangement closely follows the true arrangement of the agent-environment system when that system generated the sensor data.

For example, suppose an agent navigating the world received a sequence sensor signals  $\{z_i\}_0^T$  at unknown poses  $\{x_i\}_0^T$ . Sensor signals  $z_i \in \mathbb{R}^n$  are typically of much higher dimension than the unknown poses  $x_i \in \mathbb{R}^m$ , e.g.  $m \ll n$ . To acquire geometric knowledge of pose, the agent seeks a set of low dimensional points  $\{y_i\}_0^T$  and a mapping from sensor signals to these points,  $z_i \rightarrow y_i$ , such that the points  $y_i$  are approximately the same dimension as the unknown poses  $x_i$  and  $\forall i, j, |x_i - x_j| \approx |y_i - y_j|$ . In other words, the relative arrangement of the learned representative points closely follows the true arrangement of poses. These learned representative points will sometimes be referred to as embeddings.

The usual approach to dimensionality reduction with unsupervised learning methods uses distances among sensor signals,  $\delta(z_i, z_j)$  where  $\delta$  is a metric over  $\mathbb{R}^n$ , as a basis for generating a set of representative low dimensional points  $\{y_i\}_i^T$ . In practice, distance measurements between sensor signals behave quite a bit differently than distance measurements between poses, but modern methods of dimensionality reduction minimize this difference by relying on only local distance measurements and additional constraints in order to generate reasonable low-dimensional representations.

Unlike most methods, both action respecting embedding and the embodied ISOMAP approach cited earlier utilize both sensor and action signals in determining low-dimensional representations. Similarly, sensorimotor embedding takes into account sequences of agent actions when constructing geometric representations out of sensorimotor experience.

As an example of why actions may be useful when constructing geometric representations, consider the following example. Let  $z$  and  $z'$  be two sensor signals separated by a pose change  $\Delta x$ . One reasonable way to describe the distance between  $z$  and  $z'$  is to set it equal to the magnitude of the underlying change in agent position,  $\delta(z, z') = |\Delta x|$ . Since the pose change is not known to the agent, another approach might be to identify some method of comparing sensory inputs that approximates the underlying change in pose  $\delta(z, z') \approx |\Delta x|$ . If this metric is locally accurate, then a global “pose space” can be inferred using any number of manifold learning methods (e.g. [7]–[9]).

The sensorimotor approach is different. Since the agent can manipulate the environment, it can measure the magnitude of actions taken to estimate the distance between sensor signals. For methods like embodied ISOMAP, if  $z$  and  $z'$  are related by an intermediate action  $u$ , then  $\delta(z, z') \approx |u|$ . In other words, the magnitude of the action separating two sensory inputs is approximately the magnitude of the pose change.

In practice, the dimension of the desired pose space is smaller than the action space. As long as the magnitude of the actions is similar to the underlying magnitude of the pose changes, any number of techniques, including sensorimotor embedding, exist for inferring a low-dimensional representa-

tion of global pose from this kind of local estimate. With sensorimotor embedding as described in the next section, the agent takes into account a *sequence of actions* that take it from  $z$  and  $z'$  to a shared goal in order to determine  $d(z, z')$ . The primary advantage of this approach over others is that it provides a good proxy for the distance between sensor states even if those sensor states are not connected by a small local action.

#### IV. SENSORIMOTOR EMBEDDING

With sensorimotor embedding, geometric knowledge is inferred from the geometry of actions. The basic framework for sensorimotor embedding involves three steps:

- 1) Learn an optimal policy for acquiring a perceptual goal;
- 2) Compute distances between perceptual states by comparing policy trajectories; and
- 3) Perform dimensionality reduction using the resulting distance matrix.

This framework is quite flexible in that every step allows for specific algorithmic choices. The first step requires specifying a method for generating an optimal policy. The second step requires choosing a method for computing the distances between agent trajectories, e.g. two sequences of state-action pairs. The third step requires choosing a method for generating a low-dimensional representation based on a distance matrix.

##### A. Policy Improvement

Much of the power of this approach comes from the ability of an agent to make improvements to its policy for navigating to a goal state. Both value function and policy search methods can be applied in this setting. The improved policy is not analyzed directly. Rather, the (simulated or actual) trajectories a policy generates are used to determine distances between perceptual states.

Perceptual goals can be “built in” as in the GRIDWORLD experiments or arise out of the application of simple gradient following behaviors as in the IMAGEBOT experiments. Though not evaluated here, these perceptual goals could also arise out of the application of self-organizing distinctive state abstractions in general high-diameter reinforcement learning problems or bootstrapped topological maps in navigation domains [10], [11].

The experiments presented below use both a value function based method in the GRIDWORLD domain and a policy search method in the IMAGEBOT domain. One key question for sensorimotor embedding is whether the policy improvements also “improve” an agent’s metric understanding of sensor space and a key empirical result established in this paper is that any method of improving an agent’s policy will also improve the accuracy of the representation generated using sensorimotor embedding.

##### B. Comparing Trajectories

In the experiments presented below, dynamic time warping is used to compare trajectories. Consider two trajectories that

end at a perceptual goal state  $\bar{z}$ . These trajectories are created by following a policy  $\pi : \mathcal{Z} \rightarrow \mathcal{U}$  and so have the form

$$\begin{aligned} z_1 &\xrightarrow{u_1} z_2 \xrightarrow{u_2} z_3 \xrightarrow{u_3} \dots \xrightarrow{u_{n-1}} z_n = \bar{z} \\ z'_1 &\xrightarrow{u'_1} z'_2 \xrightarrow{u'_2} z'_3 \xrightarrow{u'_3} \dots \xrightarrow{u'_{m-1}} z'_m = \bar{z}, \end{aligned}$$

where each action is the result of applying the policy  $\pi$  at each state in the trajectory, e.g.  $\pi(z_j) = u_j$ . Using dynamic time warping (e.g. [12]), it is possible to compare the sequence of actions for each trajectory in a way that finds the minimum difference between the two sequences. Using this difference, it is possible to define the following dissimilarity measure on the sensor space  $\mathcal{Z}$  relative to the policy  $\pi$ :

$$\delta_\pi(z_1, z'_1) \equiv \text{DTW}(\langle u_i \rangle_1^{n-1}, \langle u'_j \rangle_1^{m-1}),$$

where DTW represents the minimum distance between two action sequences under dynamic time warping.<sup>2</sup>

Informally, the distance between sensory states is the distance between the sequence of actions that bring about a particular perceptual goal.  $\delta_\pi(z, z')$  may be zero for some sensory states  $z$  and  $z'$  that differ but require the same sequence of actions to reach a perceptual goal state  $\bar{z}$ . This means that this method aliases sensory states with identical dynamics. One unique aspect of this approach is that it depends just on the sequence of actions for the distance computation. Perceptual information is used only for learning and applying the policy, not directly for determining distances.

In cases with stochastic dynamics or policies, comparing individual trajectories may yield inaccurate estimates of the sensorimotor distance. In this case an agent would need to use many trajectory samples in order to accurately estimate the distance between sensory states.

Just like methods of dimensionality reduction applied to only sensor data, the result of sensorimotor embedding is a non-parametric map between sensor states  $\{z_i\}$  and corresponding low-dimensional representative points  $\{y_i\}$ . There are many possible approaches to deal with new points including interpolating among nearest neighbors to infer low-dimensional points for new data or performing regression on the set of sensor state, low-dimensional point pairs  $\{z_i, y_i\}$ .

Ballistic policies are an important special case for sensorimotor embedding.

**Definition (Ballistic Policy).** A policy  $\pi$  is a *ballistic policy* if  $\pi(z)$  results in an action that takes an agent immediately to a goal state  $\bar{z}$ .

If the agent can learn a ballistic policy in the region of a perceptual goal, then it can associate with each sensor signal an action space coordinate given by the ballistic policy. The agent can then infer an embedding directly, without the

intermediate step of multidimensional scaling using inferred distances discussed below. In any case, the key difference between sensorimotor embedding and other methods is the intermediate step of learning and analyzing a policy.

### C. Dimensionality Reduction

For the experiments presented here, multidimensional scaling [7] is used to generate a set of low-dimensional points (referred to as an embedding) based on the interpoint distances computed in step two. In principle many other methods, including non-linear approaches can be applied at this stage.

Classical multidimensional scaling (MDS) was chosen since it is a widely available, efficiently computable linear method still in common use. MDS could also be applied to distance matrices computed using sensor distances directly, allowing for a comparison that highlights the contribution of the novel dissimilarity measure introduced in this paper without the confounding effects of other dimensionality reduction strategies. MDS still requires that the dimension of the output be chosen, but this dimension can be determined by analyzing the eigenvalues associated with each dimension of the transformed representation (Figure 3).

## V. EXPERIMENTS

The steps described above are evaluated in GRIDWORLD and ROVINGEYE domains. The GRIDWORLD domain is a simple discrete type of Markov decision process meant to establish whether geometric information concerning the location of states in the domain can be extracted from policy trajectories using sensorimotor embedding. The ROVINGEYE domain provides an environment analogous to the visual ego-sphere of a developing robot.

### A. GRIDWORLD Experiments

GRIDWORLDS provide a simple discrete environment for analyzing the ability of different sensorimotor methods to recover the spatial layout of the world from the sensorimotor experience of the agent. There are many algorithms for learning optimal policies, and GRIDWORLDS provide a simple abstract model for testing these approaches.

In Figure 1, trajectories generated using a random policy did not lead to a reasonable embedding of the corresponding states. However, after learning an optimal policy with Least-Squares Policy Iteration [14] the same analysis resulted in a far more accurate reconstruction of the underlying state geometry. This result shows that performing sensorimotor embedding using trajectories from an optimal policy leads to a low-dimensional embedding of the states that closely follows the ground-truth arrangement. Alternatively, this result demonstrates that optimal policies contain implicit information about environment geometry that sensorimotor embedding makes explicit.

Figure 2 highlights the advantage of using sensorimotor embedding over approaches that only use local distances. The sensorimotor embedding approach is able to correctly determine the relative locations of states that are adjacent

<sup>2</sup>Dynamic time warping is not a proper metric over the space of action sequences since the triangle inequality does not hold (see [13] for an example), though experiments show that dynamic time warping works well in practice.

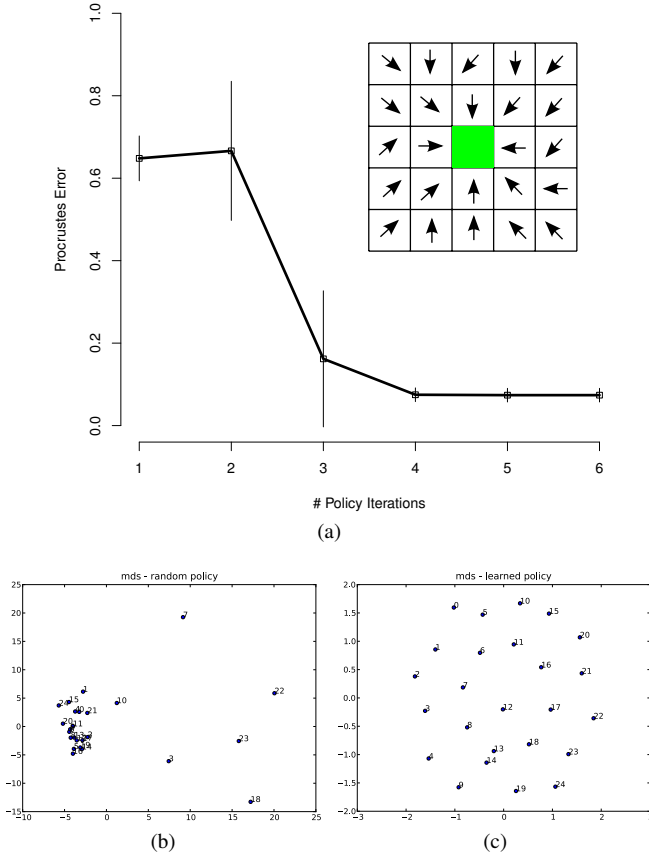


Fig. 1: Figure (a) shows how the error decreases as the policy improves with each iteration of least-squares policy iteration (LSPI) [14]; the subplot is a visualization of an optimal policy in the GRIDWORLD domain used for this experiment. Figure (b) shows the result of inferring distances from a random walk policy. Figure (c) shows multidimensional scaling [7] applied to the distance matrix inferred from policies learned using LSPI. Sensorimotor embedding is able to recover the state space geometry using the learned optimal policy.

in the original environment, but separated by a barrier that prevents any direct movement between them. Approaches that only use local distance cues, like ISOMAP and ARE, fail to capture the global geometric structure of the domain in only two dimensions. By using trajectories to the goal state, sensorimotor embedding can provide a two-dimensional representation of the state geometry that is close to the ground truth.

## B. ROVINGEYE Experiments

In the ROVINGEYE domain, a simulated eye moves around a static image. The goal of the agent is to learn to localize. This domain was used in related work learning sensor geometry (e.g. [2], [6]) and learning embeddings using action labels [15]. Unlike the simpler GRIDWORLD domain, the ROVINGEYE domain involves continuous action spaces and high-dimensional perceptual inputs in the form of sub-images of a natural scene. This provides a good test for the ability of this scheme to

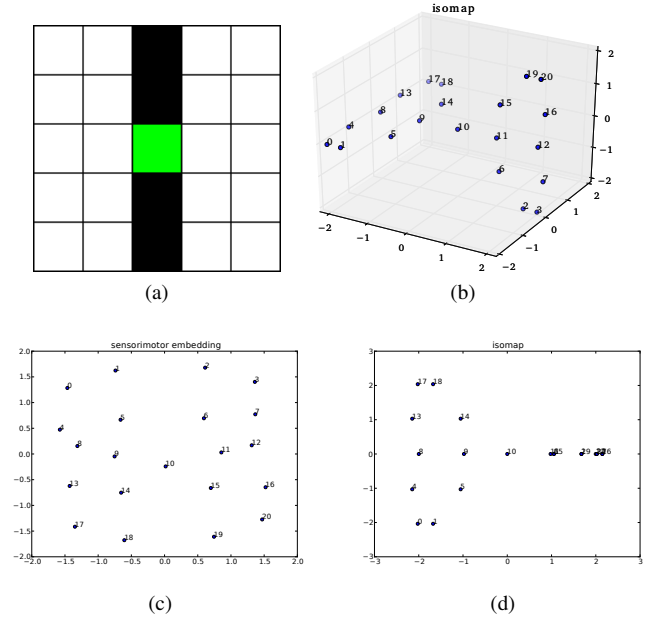


Fig. 2: Figure (a) shows the GRIDWORLD environment used for this experiment. Figure (d) shows a 2D embedding generated using ISOMAP with distances drawn from the magnitude of the local actions that move between states. Figure (b) shows the 3D embedding using the same approach. Figure (c) shows the result of applying sensorimotor embedding to full trajectories. By using the full trajectories to the shared goal state to determine interstate distances, sensorimotor embedding is able to generate an accurate representation of the relative locations using only two dimensions.

reduce the dimension of the input data.

In the experiments in this paper, the eye was a 128x128 array of pixels (Figure 4). The perceptual goal states for this domain were generated by specifying a gradient policy using a set of directional filters applied to the intensity image. By following the gradient policy from each starting point in the image, the agent can identify a much smaller set of local maxima, that when clustered, form a reasonable set of perceptual targets for learning. Following the gradient from each point in the image results in a trajectory that terminates at one of these perceptual goals. The clusters represent the regions of attraction for these local maxima. The filters used and the resulting clusters are shown in Figure 4.

The principle goal of this experiment is to establish a link between the quality of the embedding and the efficacy of the trajectories that bring the agent from points in the environment to perceptual goals. To this end several different approaches to generating trajectories of varying quality were used.

For each of the multi-step policies, the action space is limited to a set of 16 discrete actions representing movements of length 5px in 16 different directions. The first type of trajectories used for sensorimotor embedding resulted from just following the gradient. For the second approach,  $\epsilon$ -gradient, the agent followed the gradient but choose random

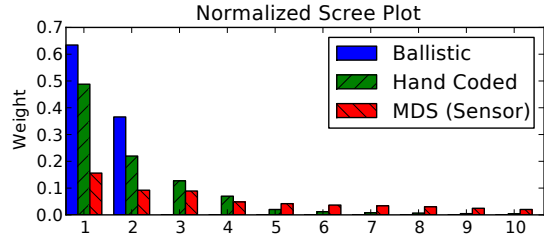


Fig. 3: The scree diagram shows the normalized weight of the first ten components in the new representation. A compact representation, such as that generated using ballistic trajectories, should have a small number of high weight components. A non-compact representation, such as that produced by MDS applied directly to sensor distances, will have a less concentrated weight distribution.

actions with a probability of 15%. The agent used the highest-scoring sample trajectories as the input for sensorimotor embedding. The third approach used a near-optimal hand-coded policy. Fourth, the agent learned a ballistic policy using the same stochastic estimation method found in [6].

Example trajectories are shown in Figure 5. These trajectories all attempt to acquire the same perceptual goal. When terminating, the agent receives a reward based on the distance to the goal. The score for a trajectory is discounted by the number of actions taken to reach the final state. This has the effect of assigning higher scores to shorter, more efficient policies. The hand-coded policy generated the highest scoring trajectories.

Procrustes analysis [16] is used to evaluate the quality of the embedding that results from applying sensorimotor embedding using each set of generated trajectories. This analysis corrects for rotation and scale differences between sets of points before computing the residual embedding error. A lower error implies that points are a better statistical fit to the ground truth data, which consists of the true pose of the roving eye corresponding to each sensor signal. The scores along with corresponding errors are shown in Table I.

Note that as the average score of the trajectories (measured over a sampling of points in the region of a single perceptual goal) increases, the error after Procrustes analysis decreases. For comparison, classic multi-dimensional scaling was applied to the raw intensity images, using pixel differences as a measure of dissimilarity. That approach (the classic linear dimensionality reduction approach) resulted in the highest error. Trajectories that score higher (i.e. are more efficient) result in lower error after performing sensorimotor embedding. Figure 3 shows the importance of each component in the new representation. The better performing methods, such as sensorimotor embedding applied to ballistic trajectories, have the most weight concentrated on a small number of components in the new representation.

Figure 6 shows the result of sensorimotor embedding on randomly selected points used in the analysis in Table I. For clarity, only the ground truth poses and the result of embedding

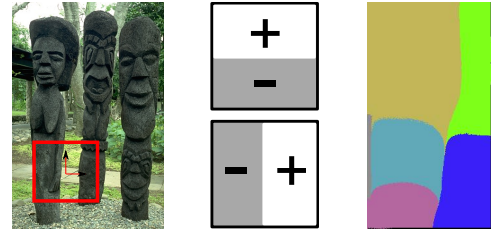


Fig. 4: In the ROVINGEYE domain, a simulated eye moves around a background image. In these experiments, the intensity image filters determine the gradient. Following the gradient results in a local maxima which serves as a perceptual goal. The image on the right shows the clustering of pixels according to the goal state that results from following the gradient policy at each pixel. Subsequent figures show the result of applying sensorimotor embedding to points in the largest cluster.

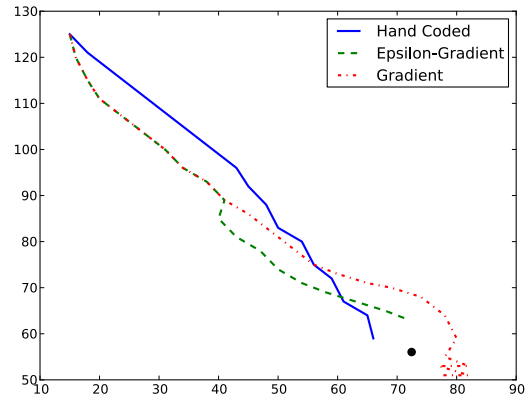


Fig. 5: This shows three example trajectories (gradient,  $\epsilon$ -gradient, and hand-coded). The action sequences are used to determine interpoint distances in the corresponding embedding. The more efficient policies result in more accurate embeddings.

gradient and ballistic trajectories are shown.

The ballistic trajectories result in a more accurate embedding than the gradient trajectories, as indicated by the Procrustes analysis in Table I. The difference in quality between using optimal multi-step trajectories and learned ballistic trajectories indicates that discretizing the action space reduces the representational power of this approach. Similar but less substantial improvements are observable with other methods of generating trajectories.

## VI. DISCUSSION AND FUTURE WORK

The experiments in this paper demonstrate that sensorimotor embedding provides a mechanism for representing geometry using sensorimotor experience, and that improvements in policies result in better embeddings. This allows agents to learn local geometry in an incremental and scalable way. In addition, since spatial representations are derived from actions using sensorimotor embedding, the resulting geometric representations are naturally calibrated to the agent's own body.

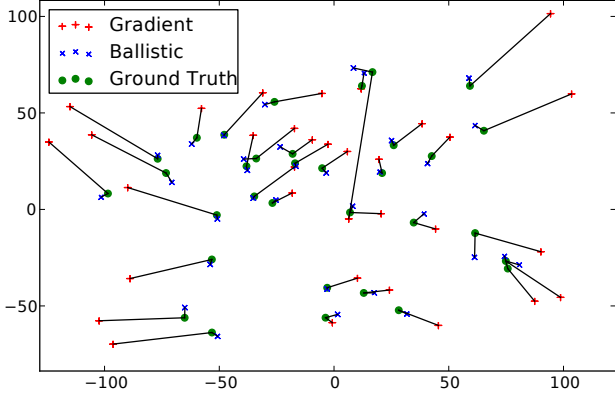


Fig. 6: This shows the ground truth, gradient and ballistic sensorimotor embeddings for a set of randomly chosen points within the region of the largest goal state cluster. Both ballistic and gradient embeddings are connected to the ground truth with line segments. The ballistic embedding provides the best approximation of the ground truth arrangement of the points.

	MDS (Sensor)	Gradient	$\epsilon$ -Gradient	HC	Ballistic
Score	NA	0.35	0.51	0.62	0.67
Error	0.80	0.20	0.11	0.05	0.01

TABLE I: As the average trajectory score increases, the residual error after Procrustes analysis decreases. The ballistic trajectories result in the smallest error, in part because the ballistic trajectories are capable of expressing the precise distance relationships between points and goal states. Multi-step trajectories using discrete actions (even with the near-optimal hand-coded policy) are only capable of approximating the ground truth interpoint distances.

Manifold learning methods have been used on a variety of different kinds of data sets for many different reasons. In only a limited number of cases have these methods been used to evaluate sensorimotor data, and in fewer cases still have these methods been applied to policies and policy trajectories. This work shows the potential benefits of utilizing policy trajectories in learning geometric knowledge.

There are several important avenues for future work. First, the experiments in this paper did not involve the kind of complex dynamics of fully humanoid robots. Showing that this method is robust in more complex domains is a key focus of future work. Since the quality of the knowledge derived from sensorimotor experience depends crucially on the ability to learn robust policies, a key issue in scaling this method involves learning policies in these more complex domains.

A second area of future work involves comparisons with the results of human experiments. Models that utilize manifold learning combined with sensorimotor experience, and sensorimotor policies, may provide some constructive clues as to certain observable but unexplained perceptual biases for tasks

that involve both perception and action.

## VII. CONCLUSION

Sensorimotor embedding is a new approach to solving the sensorimotor geometry problem. Unlike other methods that use only perceptual data or local distances, sensorimotor embedding takes full advantage of the interactive experience of an embodied agent.

The experiments show that agents can use sensorimotor embedding applied to interactive experience to recover the geometry of the environment in both the GRIDWORLD and ROVINGEYE domains. In addition, policies that improve on gradient ascent result in more accurate embeddings, demonstrating that agents can acquire geometric knowledge incrementally and robustly through policy improvements.

## ACKNOWLEDGMENTS

This work has taken place in the Intelligent Robotics Labs at the University of Texas at Austin and at the University of Michigan. Research in the Intelligent Robotics Labs is supported in part by grants from the National Science Foundation (IIS-0713150 to UT Austin and CPS-0931474 to UM) and from the TEMA-Toyota Technical Center to UM.

## REFERENCES

- [1] W. James, *The principles of psychology*. Henry Holt and Co, 1890, vol. 1.
- [2] D. Pierce and B. Kuipers, "Map learning with uninterpreted sensors and effectors," *Artificial Intelligence*, vol. 92, no. 1-2, pp. 169–227, 1997.
- [3] D. Philipona and J. O'Regan, "The sensorimotor approach in CoSy: The example of dimensionality reduction," *Cognitive Systems*, pp. 95–130, 2010.
- [4] M. Bowling, D. Wilkinson, A. Ghodsi, and A. Milstein, "Subjective localization with action respecting embedding," *Robotics Research*, vol. 28, pp. 190–202, 2007.
- [5] J. Stober, L. Fishgold, and B. Kuipers, "Sensor map discovery for developing robots," in *AAAI Fall Symposia Series: Manifold Learning and Its Applications*, 2009.
- [6] —, "Learning the sensorimotor structure of the foveated retina," in *Proceedings of the Ninth International Conference on Epigenetic Robotics*, 2009.
- [7] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*. CRC Press, 2001.
- [8] J. Tenenbaum, V. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, p. 2319, 2000.
- [9] K. Weinberger, F. Sha, and L. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proceedings of the 21st International Conference on Machine Learning*, 2004.
- [10] J. Provost, B. Kuipers, and R. Miikkulainen, "Developing navigation behavior through self-organizing distinctive-state abstraction," *Connection Science*, vol. 18, no. 2, pp. 159–172, 2006.
- [11] B. Kuipers and P. Beeson, "Bootstrap learning for place recognition," in *Proceedings of the National Conference on Artificial Intelligence*, 2002.
- [12] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.
- [13] Müller, M., *Information Retrieval for Music and Motion*. Springer, 2007.
- [14] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *The Journal of Machine Learning Research*, vol. 4, pp. 1107–1149, 2003.
- [15] M. Bowling, A. Ghodsi, and D. Wilkinson, "Action respecting embedding," in *Proceedings of the 22nd International Conference on Machine Learning*, 2005.
- [16] I. Dryden and K. Mardia, *Statistical Shape Analysis*. Wiley New York, 1998.