Discovering Natural Kinds of Robot Sensory Experiences in Unstructured Environments

Daniel H Grollman and **Odest Chadwicke Jenkins** and **Frank Wood**
Brown University Department of Computer Science
Providence, RI 02906
{*dang,cjenkins,fwood*}*@cs.brown.edu*

Abstract

We derive categories directly from robot sensor data to address the symbol grounding problem. Unlike model-based approaches where human intuitive correspondences are sought between sensor readings and facets of an environment (corners, doors, etc.), our method learns intrinsic categories (or natural kinds) from the raw data itself. We approximate a manifold underlying sensor data using Isomap nonlinear dimension reduction and use Bayesian clustering (Gaussian mixture models) with model identification techniques to discover kinds. Applying our technique to sensor data of different modalities and from different physical spaces we demonstrate robustness with respect to noise and robot location. We also demonstrate a method for applying learned kinds to new sensor data (out-of-sample readings) in real time to show the efficacy of our technique as a foundation for topological mapping and autonomous control. Lastly, we discuss the application of our technique toward massive (250,000 datapoint) data sets.

1. Introduction

The symbol grounding problem in robotics deals with connecting arbitrary symbols with entities in the robot's world. Names such as 'door', 'hallway, and 'corner' must be associated with sensor readings so that an autonomous robot can reason about them at a higher level. Traditionally, a human programmer is relied upon to provide these connections by identifying areas in the world that correspond to preconcieved labels and building *models* of how they would appear to the robot. However, actual sensory information is dictated by the robot's embodiment and may not accord with models of sensor function. Consequently, our understanding of a robot's perception of the world is often biased and heuristic.

A data-driven approach to sensor analysis could discover a more appropriate interpretation of sensor readings. Sensor data collected during robot operation are observations of the underlying sensory process and, if teleoperation is involved, the control policy of the operator. We posit that the intrinsic structure underlying robot sensor data can be uncovered using recent techniques from manifold learning. Once uncovered, sensory structures can provide a solid foundation for autonomous sensory understanding as the robot's perceptual system is allowed to develop classes of sensor data based on its own, unique, experiences.

We present here a data-driven method for classifying robot sensor input via unsupervised dimension reduction and Bayesian clustering. We view the input of the system as a high dimensional space where each dimension corresponds to a reading from one of the robot's sensors. Due to structure in the robot's environment, this space is sparsely explored. Our approach is to embed sensor data into a lower-dimensional manifold that condenses this space and captures latent structure. Distances on this manifold correspond to variance

among sensor readings as the robot moves about its environment. By clustering in this embedded space we generate simpler probability densities while grouping together areas that appear similar to the robot. Each cluster of sensor readings then corresponds to a kind[1] of entity as viewed by the robot.

Once classes are learned, we can quickly apply them to new sensor readings with an out-of-sample (OOS) classification procedure. This procedure projects new samples into the embedding space where they are classified with a Gaussian mixture model (GMM). When a location is revisited, this procedure should embed the new readings near the old ones, allowing them to be classified the same.

2. Related Work

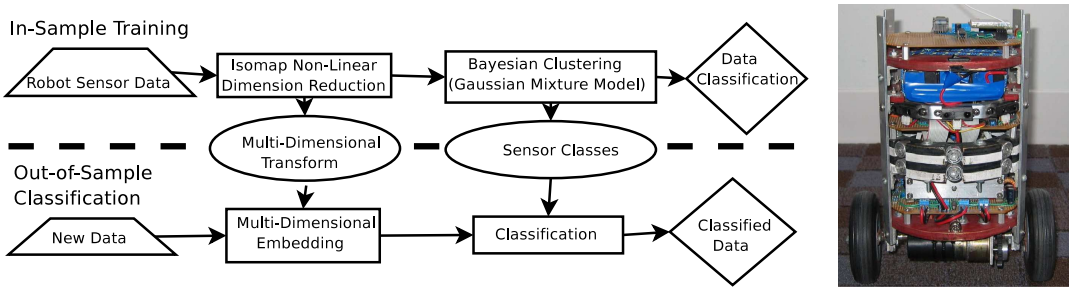
Topological mapping depends on the ability to discover regions in an explored area (Thrun, 1998). This process is usually done by extracting features from sensor data that indicate the robot's current location. When a human decides which region types exist in the robot's world and which features are important (Tomatis et al., 2003), biases from models of sensor operation are introduced. We attempt to remove these biases by deriving classes directly in sensor space.

Localization techniques also depend on region identification. Landmarking, or the identification of unique places, is commonly used to let a robot know when it has returned to a previously visited location on a map (i.e., revisiting, loop closure). The revisiting problem is key when it comes to map-making because it allows a robot to discover loops in the world (Howard, 2004) or, in the case of multiple exploration robots, it allows one robot to discover when it has entered space explored by another (Stewart et al., 2003). Often, landmarking is accomplished by modifying the environment to disambiguate similar places. We hypothesize that with a data-driven classification technique, it will become clearer which areas of the world look similar to the robot and require disambiguation. Without landmarking, localization depends on estimating the location of the robot using, for example, a Hidden Markov Model (Shatkay, 1998) or the connections between regions already seen (Howard et al., 2001). All of these approaches require a robust way of identifying the kind of space that the robot currently occupies.

A semi-supervised approach to discovering clusters in vision data is introduced in Grudic and Mulligan (2005). By allowing each cluster to self-optimize its parameters, they are able to discover clusters that more accurately correspond to the predefined ones, as well as detect outlying points that do not belong to any cluster. However, the original clusters must be decided upon by human operators and exemplar photographs of each cluster are provided to the algorithm. In contrast, our approach is completely unsupervised and allows for the discovery of space classes and outliers that are potentially non-obvious to humans.

In order to tie sensing and action together, Klingspor et al. (1996) learn sensory and action concepts directly from the sonar data of a robot, after the data is segmented and categorized by hand. By utilizing sensor information related to actions (such as wheel encoder data), we can determine the usual action performed in each space class in an unsupervised way and use these actions as a first-attempt control policy.

1. Philosophically, a natural kind is a collection of objects that all share salient features. For instance, the 'Green Kind' includes all green objects. We use the terms 'kind', 'class' and 'category' interchangably.



(a) Method (b) Crunch

Figure 1: Our method in flowchart form. Data from robot sensors are analyzed with Isomap to obtain a low-dimensional embedding. The embedded data is then clustered to develop sensor classes. Out-of-sample data can be quickly transformed and classified using the information developed during the in-sample training.

Advancements in dimension reduction (DR) and manifold learning have been shown to have benificial effects on reinforcement learning (Roy and Gordon, 2003; Mahadevan, 2005). We believe our approach can help bring these benefits to autonomous robot understanding.

3. Methodology

Our method, outlined in figure 1(a), views d-dimensional robot sensor data as lying on a manifold in $\mathbb{R}^d$. We model each sensor datum $\vec{x}$ as having been generated by a mixture model on this manifold, where each mixture density corresponds to a natural kind. Here we closely follow the methods and notation of Bengio et al. (2004).

For training, let $D = \{\vec{x}_1, \ldots, \vec{x}_N\}$ be the collection of readings from S sensors at N time instances. We compute an affinity matrix M by approximating the geodesic distance between points on the sensor data manifold. As in Tenenbaum et al. (2000), we define the geodesic distance between points a and b to be:

$$\tilde{D}(a,b) = \min_p \sum_i d(p_i, p_{i+1})$$

where p is a sequence of points of length $l \geq 2$ with $p_1 = a$, $p_l = b$, and $p_i \in D \; \forall i \in \{2, \ldots, l-1\}$ and (p_i, p_{i+1}) are neighbors as determined by a k-nearest neighbors algorithm. We compute $\tilde{D}$ by applying Dijkstra's algorithm (Cormen et al., 1990) to the graph $V = D, E = \{p_i, p_{i+1}\}$ where edge length is the Euclidian distance between neighbors.

M is formed with elements $M_{ij} = \tilde{D}^2(x_i, x_j)$ and then converted to equivalent dot products using the "double-centering" formula to obtain $\tilde{M}$. In practice, this grows as N^2 and is thus currently infeasible to calculate for more than a few thousand points.

The k dimensional embedding $\vec{e}_i$ of each sensor output $\vec{x}_i$ on the sensor data manifold is approximated by the vector $\vec{e}_i = [\sqrt{\lambda_1} v_{1i}, \sqrt{\lambda_2} v_{2i}, \ldots, \sqrt{\lambda_k} v_{ki}]$ where λ_k is the k^{th} largest eigenvalue of $\tilde{M}$ and v_{ki} is the i^{th} element of the corresponding eigenvector. We reduce the dimensionality of the sensor data by setting $k < d$, thus removing many of the low eigenvalue coordinates of the embedding. Let $E = \vec{e}_1, \ldots, \vec{e}_N$ be the reduced dimensionality embedding of the training sensor data D, henceforth refered to as the "sensor embedding."

Initially, we assume that the sensor embeddings were generated by exactly J statistically distinct intrinsic classes of sensor readings. We assume that the distribution of each of these classes is Gaussian and fit E with a mixture model with J components.

The probability that $\vec{e}_i$ was output by the robot's sensors while it was in a physical space corresponding to sensor class j, $1 < j < J$ given these assumptions is:

$$P(\vec{e}_i|j) = \frac{1}{(2\pi)^{\frac{k}{2}}\sqrt{det(\Sigma_j)}} exp(-\frac{1}{2}(\vec{e}_i - \mu_j)^T \Sigma_j^{-1}(\vec{e}_i - \mu_j))$$

where μ_j and Σ_j are the mean and covariance of the sensor output while in class j.

Assuming that each sensor datum is independent, then the probability of E according to the mixture model is:

$$P(E) = \prod_{i=1}^{N} \sum_{j=1}^{J} \alpha_j P(\vec{e}_i|j)$$

where the $\alpha_j > 0$ are mixing coefficients and $\sum_{j=1}^{J} \alpha_j = 1$.

The EM algorithm (McLachlan and Basford, 1988) is used to maximize $P(E)$ by solving for optimal distribution parameters and membership weights. This maximization is accomplished by the iterative optimization of a log likelihood function:

$$\log(\mathcal{L}(\Theta|E, \mathcal{Y})) = \sum_{i=1}^{N} \log(\sum_{j=1}^{J} \alpha_{y_i} P(\vec{e}_i|\Sigma_j, \mu_j))$$

where $\Theta = \{\mu_1, \ldots, \mu_J, \Sigma_1, \ldots, \Sigma_J\}$ is a set of unknown parameters corresponding to the mean sensor data embeddings and covariance matrices for the J classes and

$$\mathcal{Y} = \{y_i\}_{i=1}^{N}, 1 < y_i < J, y_i \in \mathbb{Z}$$

is an array of unknown variables such that $y_i = j$ if $\vec{e}_i$ came from mixture component j.

Model selection is a central issue in clustering and corresponds to determining the number of clusters (intrinsic classes) in the data. We do not solve this problem; instead we use two empirical criteria for choosing the model. One, we look for an inflection or maxima in a penalized training data log-likelihood. We used the Bayesian Information Criteria (BIC) which penalizes the likelihood as a function of the complexity of the model. If κ is the number of free parameters in the model, then we calculate the BIC as:

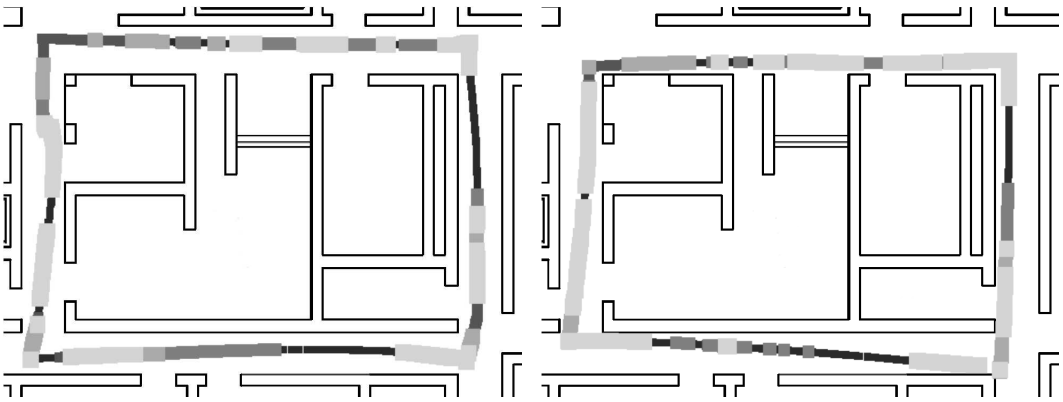
$$-2\log(\mathcal{L}(\Theta|E, \mathcal{Y})) - \kappa(\log(N) + 1)$$

In practice, the BIC often doesn't sufficiently penalize complex models, so we additionally use cross-validation on held-out data to check for overfitting: We train our model on half the training data and then compute the unpenalized likelihood of the remainder. When too many classes are posited, i.e. the model may be over-fit, the likelihood of the held-out data may decrease relative to simpler models. These two techniques are often in agreement.

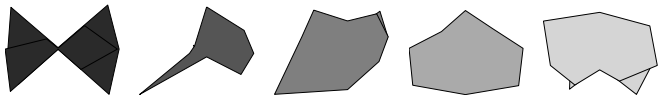
Online classification of a new point $\vec{p}$ is simple and rapid. We refer the reader to Bengio et al. (2004) for full details. The embedding is given by:

$$e_k(\vec{p}) = \frac{1}{2\sqrt{\lambda_k}} \sum_i v_{ki}(E_{\vec{x}}[\tilde{D}^2(\vec{x}, \vec{x}_i)] + E_{\vec{x}'}[\tilde{D}^2(\vec{p}, \vec{x}')] - E_{\vec{x},\vec{x}'}[\tilde{D}^2(\vec{x}, \vec{x}')] - \tilde{D}^2(\vec{x}_i, \vec{p}))$$

where E is an average over the training data set. Using the GMM from the training stage, we determine the probability of this newly embedded point belonging to each cluster.



(a) Trip1, 4th Floor (b) Trip2, 4th Floor



(c) Class1 (d) Class2 (e) Class3 (f) Class4 (g) Class5

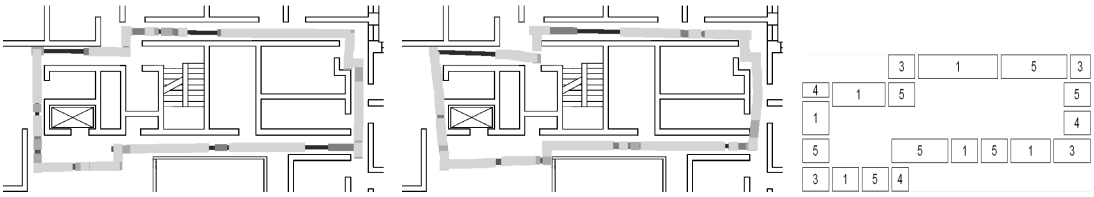
Figure 2: Results from running Crunch on the fourth floor of the Brown University CIT. 2(a): Sensor data from trip 1 has been clustered into 5 classes. A unique width and grayscale value for each class is overlaid on registered odometry to show the classification of regions of space. 2(b): The learned classes were used to classify data from trip 2. 2(c)-2(g): Show the expected sensor readings for each class.

4. Experiments

We collected unmodified sensor data from a small, cylindrical, inverted pendulum robot named Crunch, pictured in figure 1(b). Crunch has eight sonar and eight IR sensors arranged in dual rings around its body as well as wheel encoders that record wheel rotation. During operation, these sensors are sampled and transmitted back to a base laptop where they are logged at around 10Hz. Our data consists of the robot's logs as it was driven in multiple trips along paths in two different office environments. Every data point is the current sonar and IR readings, as well as the distance each wheel has turned since the last reading.

Our first experiment was designed to test the consistency of our classification when a location is revisited. We first used data from one trip along a path to learn data classes using the described method. Then, data from a second trip along a similar path were classified using our learned model. After computing the geodesic distance and MDS embedding of the first trip, we retained 8 of the resulting dimensions for future processing. Based on the BIC and holdout calculations, we judged that there were 5 classes in the data. The resulting mixture model was used to assign each datapoint to a class. For display purposes, we manually registered the odometry with the underlying floor plan and overlaid these classes on the path that the robot followed. This assignment is illustrated in figure 2(a).

Using the embedding and the classes derived from the first trip, we classified all the data from the second trip. As the robot followed the same general path as it did in the first trip, we expected the sensory readings along the path to be classified similarly. The results from the out-of-sample classification are shown in figure 2(b).



(a) Trip1, 5th Floor (b) Trip2, 5th Floor (c) Topological Map

Figure 3: Using sensor classes from 2, Crunch took two trips around a different area of our building and classified each location. As before, the two trips are classified similarly, showing that the learned classes are applicable in other (although similar) locations. 3(c): A topological map derived from the area classifications.

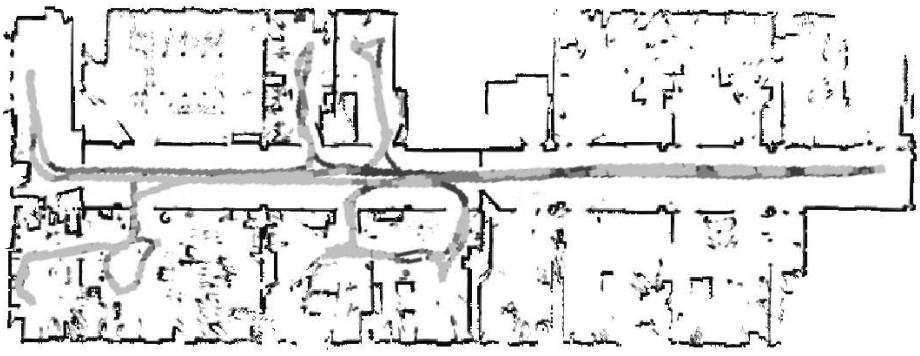
We used the registered odometry to compare classifications of the same physical space across trips. Given a position (x, y) in the first trip that has been classified as generating sensor readings of kind k, we compare it to all points from the second trip within a one foot radius. If more than a third of these points have also been classified as kind k, we declare a match. Under this metric, 60% of locations were consistently classified across the two trips.

Sensor readings from the same class of sensor space should be similar to each other and different from those from other classes. Figures 2(c)-2(g) show expected readings from each of the 5 classes discovered by our method. These images were generated using a "ray model" of Crunch's IR and sonar sensors and all values were computed from a weighted average of all datapoints across both trips. Under this model, many of these shapes are hard to interpret as corresponding to a hallway, doorway, corner, etc, but these are the sensor readings that are most distinguishable to the robot.

Using the classes derived from a trip around the fourth floor of our building, we classified data readings from a trip around the fifth floor using our real-time out-of-sample classification technique. Results are shown in figure 3. If the learned classes were non-applicable to the space, that is, if areas that looked similar to the robot were not assigned to the same cluster, we would expect to see successive data points assigned to different classes. Instead, there are several large contiguous sections of points that are all assigned to the same class. Furthermore, by repeating the consistency test from above, and classifying data from a second trip on the fifth floor using the same classes, we see that these classifications are usable in this area, even though they were learned in another.

4.1 Mapping and Control

By consistently categorizing the robot's physical location, our system paves the way for the creation of topological maps of the robot's environment. Such "robot-centric" maps (Grudic and Mulligan, 2005) require that the robot accurately recognize when it is in certain types of space. By combining our classification with odometric or ground truth data, rough topological maps can be derived. Figure 3(c) shows a topological map derived from 3(a) by dividing the space into regions based on classification. Further processing with loop-closure algorithms and landmark identification techniques (Howard, 2004) can refine these maps into useful tools for autonomous robot operation.



(a) Fr079

(b) Loop Closeup (c) Class1 (d) Class2 (e) Class3

Figure 4: The Fr079 data set was processed with our technique to determine the embedding and classes. 4(b): A closeup of a loop in the robot's path where locations are classified consistently. 4(c)-4(e): The expected laser ranges of each of the 3 classes.

In addition, control algorithms can be derived from the motor data associated with each class. By using these data in the training process, areas that are clustered together not only look similar, but are areas where the robot should behave similarly as well (at least according to the teleoperator). Leveraging this, we can use the average movement of the robot in each space class as a first-pass control policy for what the robot should do if it finds itself in that class. We envision using this ability to perform robotic learning by demonstration. After being led through a task by a human teleoperator, a robot can segement the task and associate actions with each segment in an unsupervised manner. As the task is repeated, more data become available to fine tune the robot's actions.

4.2 Towards Embedding Massive Data Sets

Theoretically, our approach applies to any amount of data in any number of dimensions. However, computational and temporal limits restrict a robot's ability to process data during operation. Thus, it is useful for a system to do much processing offline and apply what it learns to new data in real time. The OOS extensibility of our system provides just this ability. However, problems still arise when dealing with the large amounts of data accumulated by long-running robots. Specifically, it is currently infeasable to run an unsupervised manifold learning technique on data sets of more than a few thousand points.

In order to test the scaling of our technique with respect to input size and sensor modality, we obtained the Fr079 data set from Radish (Howard and Roy, 2003). Fr079 consists of $\sim$3000 datapoints of 360 degree laser rangefinder scans. This data set also contains several loops which can be examined for classification consistency. Figure 4 displays our results from the Fr079 data set and figure 4(b) shows a closeup of one of the loops made

(a) Swiss Roll (b) HLC (c) OOS

Figure 5: 5(a) Swiss roll data: a rectangular 2D manifold embedded in 3D space. On 250,000 points, HLC performace 5(b) is highly degraded, while OOS with hierarchical landmarking fares much better 5(c).

by the robot in its exploration. Each time the robot is at the same location in the world (or very nearby), the system detects it as being from the same class, showing that the classification produced by this system is consistent.

Unfortunately, the Fr079 data sets pushes the computational limits of our current technique. Our extension to Isomap, Hierarchical Landmark Charting (HLC) (Jenkins and Ketpreechasawat, 2005), was designed to address these computational limitations. HLC partitions the data into overlapping charts and takes one point from each chart as a global landmark which then becomes common to every chart. Each chart is embedded by itself and the global landmarks are used to align the charts into a global embedding. As the amount of data increases, so do the number of global landmarks. Once this number grows past computational feasability, another charting step is performed, and new, higher level, landmarks are obtained. Since the number of points in each chart is well below the computational limits of Isomap, processing is greatly sped up.

We tested HLC on swiss roll data to show proof of concept. These data, shown in 5(a), are a rectangular 2D manifold embedded in 3D space. With current computing systems, Isomap operates well to around 5,000 datapoints. As reported by de Silva and Tenenbaum (2003), landmark approaches to global manifold learning are sensitive to the number and placement of landmarks with respect to the size and shape details of the input data. HLC attempts to combine the sparsity of current landmark approaches with reasonable placement and hierarchical division of computation.

With HLC, we obtained good results up to around 50,000 datapoints. Unfortunately, as we grew the number of points up to 250,000, performance was greatly degraded (See figure 5(b)). We belive that HLC fails due to accumulated errors in aligning the charts. As the number of charts increases, so do these errors, until the discovered embedding no longer resembles the true one. To address this issue, we combine HLC with the OOS technique discussed above. Taking the global landmarks discovered by HLC as the in-sample training data, we embed them in a lower dimensional space and treat the rest of the data as out-of-sample readings. Since each point is embedded individually with respect to the global landmarks, it has no effect on the embeddings of other points and therfore alignment errors do not accumulate. The results of this approach on the 250,000 point swiss roll can be seen in figure 5(c). We believe we can further improve upon this technique by leveraging the hierarchical way in which the landmarks are chosen: Top level points can be used to embed points in the next level down, which in turn operate as landmarks for the levels below them. The goal of this work is to extend our ability to process data into the multi-million datapoint range.

5. Discussion and Conclusion

This paper presents an extensible method for data-driven discovery of intrinsic classes in robot sensor data. The discovered classes are consistently recognizable and reapplicable to new data using out-of-sample techniques. We discuss modifications to the technique that will allow it to deal with massive data sets.

We attempt to remove human bias from the analysis of robotic sensor data by identifying latent structure in the sensor readings themselves. Currently, we empirically determine the neighborhood function and size, the number of embedding coordinates to retain, and the number of intrinsic sensor classes. In theory, each of these can be determined automatically, and perhaps even adaptively, from the data. Here we demonstrate that intrinsic sensor classes may form a better foundation for applications that require classifying physical space from sensor data. In addition, we treat each sensor reading as independent. Better performance may result from modeling spatial and temporal correlations as in ST-Isomap (Jenkins and Matarić, 2004). Parametric Embedding (Iwata et al., 2004) is a an approach that preserves associations between data objects and mixture components during embedding, which could be useful in this context.

Because our technique operates in a space defined by robot sensors, the results are sometimes difficult to reconcile with human intuition. In particular, when the "canonical" sensor reading for a Crunch class is examined, it does not correspond to any class that we, as humans, would have developed for the robot. In fact, even the *number* of classes in the space differs. However, as Crunch is a small wheeled robot equipped with sonar and IR and we are tall humans with eyes, it makes sense that our world views, and our divisions of that world into categories, would be different. Our intuition is further bolstered by noting that armed with the kinds discovered by our system, a human crawling on his hands and knees through the area explored by Crunch can see how they match up.

Alas, there is no "ground truth" we may use to evaluate our model. By design we cannot determine the "correct" classification of each point in robot sensor space. At most, we can use an ad-hoc metric to test classifications for consistency. The metric described here is highly sensitive to registration errors and constant selection. It serves only to help us intuit that our classification scheme is consistent, reapplicable, and above all, useful.

References

Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems 16*. 2004.

T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press/McGraw-Hill, 1990.

V. de Silva and J. B. Tenenbaum. Global versus local methods for nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems*, 15:705–712, 2003.

G. Grudic and J. Mulligan. Topological mapping with multiple visual manifolds. In *Robotics: Science and Systems (R:SS 2005)*, 2005.

A. Howard. Multi-robot mapping using manifold representations. In *IEEE International Conference on Robotics and Automation*, pages 4198–4203, New Orleans, Louisiana, Apr 2004.

A. Howard, M. J. Matarić, and G. S. Sukhatme. Relaxation on a mesh: a formalism for generalized localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1055–1060, Wailea, Hawaii, Oct 2001.

A. Howard and N. Roy. The robotics data set repository (radish), 2003.

T. Iwata, K. Saito, N. Ueda, S. Stromsten, T. L. Griffiths, and J. B. Tenenbaum. Parametric embedding for class visualization. In *Neural Information Processing Systems*, 2004.

O. C. Jenkins and S. Ketpreechasawat. Hierarchical landmark charting for 3d volume skeletonization. Unpublished, 2005.

O. C. Jenkins and M. J. Matarić. A spatio-temporal extension to isomap nonlinear dimension reduction. In *The International Conference on Machine Learning (ICML 2004)*, pages 441–448, Banff, Alberta, Canada, Jul 2004.

V. Klingspor, K. J. Morik, and A. D. Rieger. Learning concepts from sensor data of a mobile robot. *Machine Learning*, 23(2-3):305–332, 1996.

S. Mahadevan. Proto-value functions: Developmental reinforcement learning. In *International Conference on Machine Learning*, 2005.

G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, 1988.

N. Roy and G. Gordon. Exponential family pca for belief compression in pomdps. *Advances in Neural Information Processing Systems*, 15, 2003.

H. Shatkay. *Learning Models for Robot Navigation*. PhD thesis, Brown University, 1998.

B. Stewart, J. Ko, D. Fox, and K. Konolige. The revisiting problem in mobile robot map building: A hierarchical bayesian approach. In *The Conference on Uncertainty in Artificial Intelligence (UAI 2003)*, 2003.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290, 2000.

S. Thrun. Learning maps for indoor mobile robot navigation. *Artificial Intelligence*, 99: 21–71, 1998.

N. Tomatis, I. Nourbakhsh, and R. Siegwart. Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems*, 44:3–14, 2003.