Transferring Embodied Concepts between Perceptually Heterogeneous Robots

Zsolt Kira, *Student Member, IEEE*

***Abstract*— This paper explores methods and representations that allow two perceptually heterogeneous robots, each of which represents concepts via *grounded* properties, to transfer knowledge despite their differences. This is an important issue, as it will be increasingly important for robots to communicate and effectively share knowledge to speed up learning as they become more ubiquitous. We use Gärdenfors' conceptual spaces to represent objects as a fuzzy combination of properties such as color and texture, where properties themselves are represented as Gaussian Mixture Models in a metric space. We then use confusion matrices that are built using instances from each robot, obtained in a shared context, in order to learn mappings between the properties of each robot. These mappings are then used to transfer a concept from one robot to another, where the receiving robot was not previously trained on instances of the objects. We show in a 3D simulation environment that these models can be successfully learned and concepts can be transferred between a ground robot and an aerial quadrotor robot.**

I. INTRODUCTION

As autonomous robots become increasingly common, it is likely that there will be multiple robots that each learn through experience; that is, via embodied interaction with the world. This type of grounded learning, however, ignores social aspects of learning. With multiple robots, it is crucial for the robots to be able to share knowledge either through explicit communication or implicit means such as imitation. Such knowledge sharing speeds up learning significantly and can reduce the need for costly human teaching.

Several problems can prohibit effective sharing of knowledge, however. Knowledge learned via exploration of the world is often embodiment-specific. It is quite common to have some degree of heterogeneity among robots, however, and there can be slight perceptual differences even among two robots of the same model. For example, the camera color models may differ slightly. It is an even greater problem when different types of robots are used. Currently, there is a plethora of robotic systems in use in home environments (e.g. the Roomba and lawn mowing robots), research labs, and in various domains where task allocation to differing robots is necessary.

Symbols are often used to abstract raw sensory readings, and facilitate communication via language. However, even assuming that these symbols are grounded [1] within each robot, there is the problem of achieving a common grounding among multiple robots, an issue that has been raised as the social symbol grounding problem [2]. Approaches that ground symbols jointly in the environment by multiple robots at the same time exist [3], but require that the robots learn in the same environment and under the same conditions.

In order to allow knowledge sharing among such heterogeneous robots, we posit that the robots can first autonomously build models of their differences and similarities, and map symbols from each robot's representation to the other. Note that in order for this to be useful, there must be *some* similarity between the two robots; i.e. if the two robots can only sense non-overlapping features of objects (e.g. sound versus vision), then meaningful communication between them will be much more difficult or impossible. The point is to leverage whatever similarity exists between the robots, and to be cognizant when the differences are too substantial for effective knowledge sharing. Mappings between the robots' representations can be built after each robot has learned its respective representations.

The building of the models can be performed by leveraging similarity to deal with heterogeneity, specifically by establishing a physically shared context. This means that, when building mappings between the two robot's representations, it must be guaranteed to some degree that they are both looking at the same scene. We have demonstrated several robot behaviors for accomplishing this in previous work for learning parameterized models of low-level sensing differences [4]. We have also demonstrated, using real robot data obtained from Pioneer robots with different cameras, that correct property mappings can be learned despite the fact that the grounding of the properties differs between them [5]. In this paper, we combine these grounded properties to learn concepts, in this case representing the appearance of physical objects. We demonstrate that these concept models can be learned via supervised instances, and that the property mappings learned can be used to facilitate knowledge sharing. Specifically, the two robots can use the models to transfer a new concept from one robot to the other. In other words, given knowledge of which object properties correspond on the two robots, a robot can teach an entirely new concept to another robot. We also use automated image segmentation throughout the work, an improvement over our previous efforts.

We use conceptual spaces to anchor sensory data to learned object properties (e.g. color or texture) and concepts (e.g. object categories or specific objects) [11]. Conceptual spaces are geometric spaces that utilize similarity measures and concept prototypes as the basis of categorization. This geometric form of representation has several advantages. It has been elaborated upon and extended in several other works

Zsolt Kira is a Ph.D. candidate with the Mobile Robot Laboratory, College of Computing Georgia Institute of Technology, 85 5th ST NW, Atlanta, GA, 30332. zkira@gatech.edu

(e.g. [6]), and discussed and implemented to a limited degree in robotic systems [7],[8]. Most importantly, understanding how different properties can be mapped between different agents can be intuitively visualized in these spaces.

In this paper, we focus on color and texture object properties. We represent properties as Gaussian Mixture Models (GMM) in an RGB or HSV space (for color properties) or a metric space representing the output of Gabor filters (for texture properties), and show how they can be learned in a supervised manner. Once all these properties are learned, we then show that actual concepts (i.e. objects) can be learned from supervised instances as well. We then show that mappings between properties in each robot can be modeled and used to transfer concepts from one robot to the other. This capability essentially bootstraps and speeds up concept learning using information from another robot. We demonstrate results in a 3D simulation environment, USARSim, with a ground robot and a quadrotor aerial robot. Differences in these robots include the color space used (RGB vs. HSV), extra properties that exist in one robot but not the other, as well as large differences in perspectives and hence portions of the object that are used for learning and classification.

II. Related Work

The key issue in this paper is related to social symbol grounding, that is finding common symbols for similar concepts across a population of agents. This is related to language formation and has been studied extensively in linguistics and evolutionary or artificial life [2][9]. For example, work done by Luc Steels and his colleagues in the area of shared vocabulary development used shared attention to synchronize two robot's symbols [9]. This is a similar concept to ours, although they did not explicitly deal with the issue of robot heterogeneity where robots may have different feature spaces.

Another example of this in robotics includes work by Jung and Zelinsky, who studied two robots that perform the same task (vacuuming) but had different capabilities; one robot swept small pieces and reached into corners, while the other could only vacuum the larger piles and could not reach corners [3]. In that case, a shared ontology was developed by establishing a physically shared context during learning: The two robots followed each other around the room and agreed on symbols for specific locations in the environment. In a similar vein, Billard and Dautenhahn have looked at a situation involving two homogeneous robots where one teacher attempts to share its symbols with another robot via imitation, namely following [10].

Conceptual spaces, the representation used here, have been used in robotics in several works. LeBlanc and Saffiotti have looked into the fusion of properties coming from different sensors into a single domain, but have so far focused on spaces with identical dimensions [8]. In our case, the underlying dimensions and properties can differ; for example, in our experiments one robot uses an RGB color space while the other uses HSV. Overall, little work in robotics has focused on sensor heterogeneity, and bridging resulting conceptual differences. This paper builds upon our previous work and shows that a hierarchical concept representation can be used to transfer concepts once the robots determine which of the representation's lower-level building blocks are shared.

III. Concept Representation and Learning

A. Abstracting Sensory Data

Sensory data is often abstracted in order to improve learning or to enable communication. In this paper, we use Gärdenfors' conceptual spaces [11] in order to bridge lower-level representations and symbols. The most basic primitive of the representation is a dimension (also referred to as attribute), which takes values from a specific range of possible values (a domain in the mathematical sense, although it is not to be confused with the notion of a domain used in the next paragraph). For example, the hue of an object can be specified as an angle in the range $[0, 1]$. The values of these dimensions come from perceptual features processed from sensor data. For example, a camera sensor measures physical properties of the world (light), converting them into a digital representation consisting of multiple pixels in the form of an RGB space. A perceptual feature detector can convert regions of the image into an HSV space, and the H (hue) value can make up a dimension. The feature detector returns a set of these, one for each region of the image that it determines is salient.

Gärdenfors posits that there are integral dimensions that cannot be separated in a perceptual sense. For example, the HSV color space can be argued to consist of three integral dimensions. A set of such integral dimensions is referred to as a domain. A domain defines a space that consists of all possible values of the integral dimensions. It is useful to abstract and divide these values into specific regions, which define a property. For example, "blue" can be a property that corresponds to some region of the color space. The regions can be arbitrary shapes, although Gärdenfors defines what he calls natural properties with certain characteristics such as convexity. Note that a property corresponds to a region in a single domain.

We can now define a *conceptual space* K as made up of a set of domains. A specific concept in the conceptual space is a set of regions from the domains $D = \{d_1, d_2, ..., d_n\}$. A point in the conceptual space is called a knoxel $k =< k_1, k_2, ..., k_n >$, and specifies instances of the concept in the form of vectors. A knoxel can specify points in some of the domains, while leaving others unspecified, in the form of a partial vector. Note that a property is a specific type of concept that utilizes only one of the domains from the conceptual space. In order to facilitate communication, symbols are attached to properties and concepts. Each robot maintains a set of symbols X, each of which is grounded to a property (or in general, a concept) via the representation. Symbols correspond to labels or strings, which will be randomly assigned by the robot. A representation can be described as a function that returns the degree to which a specific knoxel k can be categorized as having the corresponding property represented by symbol $x \in X$; i.e. $R : (k, x) \rightarrow [0, 1]$. Each property has a prototype in the form of a knoxel, denoted as k_p. We now describe our implementation of properties within the framework of GMMs.

B. Learning Properties from Instances

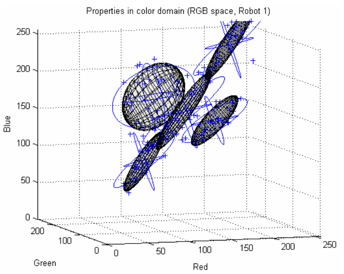

Fig. 1. Color properties, represented as a GMM, after training with multiple objects with five colors (obtained from real robot data [5]).

In order to learn a representation for object properties, we will scaffold the robot's learning by first providing it with multiple instances of data that contain a property. Note that no labels are given, and the robot creates its own random labels. Each scene, which can contain multiple properties and concepts, results in a set of knoxels K calculated from the output of the robot's perceptual feature detectors. In this paper, it is assumed that it is known which domain is to be trained for a set of instances. For each property p_i, we use a Gaussian Mixture Model (GMM) to characterize the regions, denoted as G_i.

Specifically, each property can be modeled as:

$$P(p_i|\theta) = \sum_{j=1} w_j P(p_i|u_j, \sigma_j) \tag{1}$$

where w_j is known as the mixing proportion and θ is a set containing all of the mixing proportions and model parameters (mean μ and standard deviation σ). An Expectation Maximization (EM) algorithm is used to determine these parameters [13]. Once models are learned, they are used to determine the membership of an instance in a property. Specifically, given sensory data, the membership of an instance i in property p is the Gaussian distance function $s(i, p$ to the nearest property cluster. Fig. 1 shows example properties in an RGB space.

C. Mapping Properties across Differing Embodiments

As mentioned properties are regions in domains, in our case represented as Gaussian clusters. The same property can be represented in two different robots as clusters with different characteristics (for example, different standard deviations) or even domains from different sensors (for example, the width of an object as detected by a camera or laser). Given these clusterings of a domain, the problem is to find associations between clusters from each robot. In order to do this, we use instances from each robot while viewing the same scene and compare properties that they see. In this paper, this is done manually and in a looser sense; manual selection of images is performed such that both robots see the same object, although not necessarily from the same perspective. Given a scene, each robot processes its sensory data to produce a set of knoxels where property memberships in relevant domains can be calculated. For each pair of properties (one from each robot), statistics described below are maintained in order to determine whether they represent similar physical attributes.

Table 1 - Confusion Matrix $PC^{A,B}$

	p_1^B	p_2^B	p_3^B	p_4^B	p_5^B
p_1^A	0.000	**0.527**	0.000	0.001	0.374
p_2^A	0.000	0.000	**0.488**	0.085	0.182
p_3^A	**0.718**	0.000	0.001	0.070	0.017
p_4^A	0.139	0.051	0.109	**0.310**	0.160

In order to do this, we seek to map individual clusters to each other by building confusion matrices between the properties (see [5] for a full description). Specifically, we utilize the confusion matrix to determine pairs of properties that may potentially represent the same physical property. Suppose that there are two clusterings G_j^A and G_k^B defining regions corresponding to properties p_j^A and p_k^B for robot A and B, respectively. Also, each clustering for robot A and B has and clusters n_j^A and n_k^B, respectively. Finally, suppose that we have a set of instances I from each robot (obtained using its own sensing) with a sufficiently high membership defined by a threshold for property p_j^A. The confusion matrix $PC^{A,B}$ is then updated with:

$$PC_{(j,k)}^{A,B} = \sum_{i}^{I} \frac{min(s(i, p_j^A), s(i, p_k^B))}{s(i, p_j^A)} \tag{2}$$

Here, $s(i, p)$ is the Gaussian membership function of instance i in property p. The min function is used to represent the intersection of property memberships, as is used commonly in fuzzy sets. For each property of a robot, the highest values in the corresponding property's row or column will be taken and it will be considered potentially corresponding to the respective property of the other robot. A threshold may be placed on this as well, although we do not in this paper. An example confusion matrix, obtained from experiments described below, is shown in Table 1. The highest entries in a row are bolded, showing corresponding properties in robots A and B.

D. Learning Concepts from Instances

There are several ways in which properties, with associated weights and correlations, can be combined to represent a concept. Often, properties are represented as well-defined regions in domains, and when an instance is in one of these regions, it is said to have the corresponding property [14]. In other words, instances or objects are defined via a conjunction of predicates (corresponding to properties), where the level of membership in the property is lost. For example, an instance may be on the edge of the region defined by the property (e.g. only slightly tall), but this information is not retained. Obviously, this type of discretization presents issues regarding uncertainty. In our work, we take into account the degree of membership for a property, as well as the degree of membership for a concept, allowing the robot to take this in consideration during the communication process and when deciding how to act.

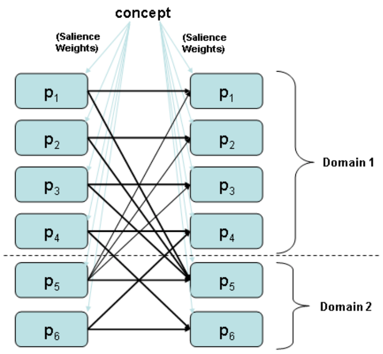

Fig. 2. Representation of a concept, consisting of correlations between five properties (figure adapted from [15]).

In order to do this, we use a modified extension of conceptual spaces to allow fuzzy memberships proposed by Rickard [15]. A concept is represented as a graph of nodes consisting of properties, with salience weights for the concept. In this paper, all saliency weights are set to one (since all domains are equally important). Nodes for pairs of properties p_j and p_k are connected with directional edges, with weight $C(j,k)$, corresponding to the conditional probability that the concept will have property p_k given that it has property p_j [15]. If the two properties are disjoint (non-overlapping regions) and are from the same domain then $C_{(j,k)} = 0$. The graph can be represented as a non-symmetric square connection matrix. The concept graph for a concept is depicted in Fig. 2, with edge weights represented by arrow thickness.

Once the properties have stabilized (in that their associations and parameters do not change greatly) concepts can also be learned via supervised learning [15]. Instances again take the form of sensory readings, with the robot attaching a random label, this time to a concept. The feature vectors are processed from sensory data and then placed as dimensions in their respective domains, and property memberships are calculated as described in subsection B. Specifically, given a set of instances processed into knoxels K_i, the instances are converted into a matrix I where $I_{i,j}$ contains the similarity between the property membership for property p_j in instance i and the prototype of property p_j (denoted as k_{p_j}). Each element of the concept matrix described previously is then calculated as follows:

$$C_{(j,k)} = \frac{1}{|I|} \sum_i \frac{min(I_{i,j}, I_{i,k})}{I_{i,j}} \qquad (3)$$

This equation calculates the average ratio of membership of the instances to both properties divided by their membership in the first property. Here, membership in both properties is represented as the minimum of the two. Note that as described in [15], the identity portion of the matrix all have a value of one. However, we modified this to contain the mean value of the property for all instances, in order to preserve the *strength* of a membership of a property in a concept (e.g. how red an ambulance is).

E. Concept Similarity and Transfer

In order to compare concepts and categorize instances, the concept matrix described above can be projected into a hypercube graph representation [15]. Specifically, the matrix C is converted into a vector c by concatenating subsequent rows together, so that values from row j and column k correspond to element $r = (j-1)N + k$ in the vector, where N is the dimensionality of the matrix (corresponding to the number of properties). The salience weights of the properties can be combined so that $w_r = w_j w_k$.

In order to categorize the degree to which a specific instance belongs to a concept, the instance is similarly converted to a graph (represented as a connection matrix as well) and compared to the concept matrix. Then, fuzzy mutual subsethood (a similarity measure between fuzzy sets) is used to obtain similarity values.

First, the instance must be converted into the hypercube representation used for concepts. Let D_c be the domains in the conceptual space and P_c be the set of all properties that are involved in the target concept, i.e., a property that has a nonzero connection between it and another property (or vice-versa). This can be defined as:

$$P_c = \{p \in P : \exists j \; s.t. \; C_{pj} > 0 \; or \; C_{jp} > 0\} \qquad (4)$$

The matrix I_c is then defined as:

$$I_{c,(i,j)} = \begin{cases} \max\limits_{j \in D_c, k \in D_c} \min\limits_{j \in P_c, k \in P_c} (s(i,p_j), s(i,p_k)) & \forall \, \mathrm{i,j} \in P_c \\ 0 & \text{otherwise} \end{cases}$$

where $s(i,p_j)$ represents the membership (similarity) of an instance to a property, which is derived from the regions representing the property (described in the previous subsection). These formulations are derived from fuzzy set theory, and their justifications are elaborated upon in [15]. They have also been successfully used in image matching tasks, for example [16]. Given this matrix, the membership of an instance to a concept can be defined using the fuzzy mutual subsethood equation:

$$s(c,i) = \frac{\sum_{i \in P_c} min(c_i, I_c)}{\sum_{i \in P_c} c_i} \qquad (5)$$

Finally, the ultimate goal of this hierarchical representation is the facilitation of knowledge transfer. Specifically, we take a concept matrix from one robot and use the property mappings learned earlier to modify it. Entries for properties that exist in the originating robot but not the receiving robots are removed entirely (by removing row i and column i from the concept matrix). Conversely, rows and columns are added for properties that exist in the receiving robot but not the originating robot, with the values being zeroed out. These entries will not contribute to classification of the concept due to the zero entries.

IV. Experimental Design & Results

A. Simulation Environment, Robots, and Concepts

In order to test the learning and transfer of concepts, we have conducted simulated experiments in USARSim, a realistic 3D simulation environment. We used an outdoor village environment containing a large number of buildings and objects. Two robots, a ground Talon robot and an aerial quadrotor robot, were used. The aerial robot flew approximately eight meters from the ground. A portion of the environment and the two robots can be seen in Fig. 3.

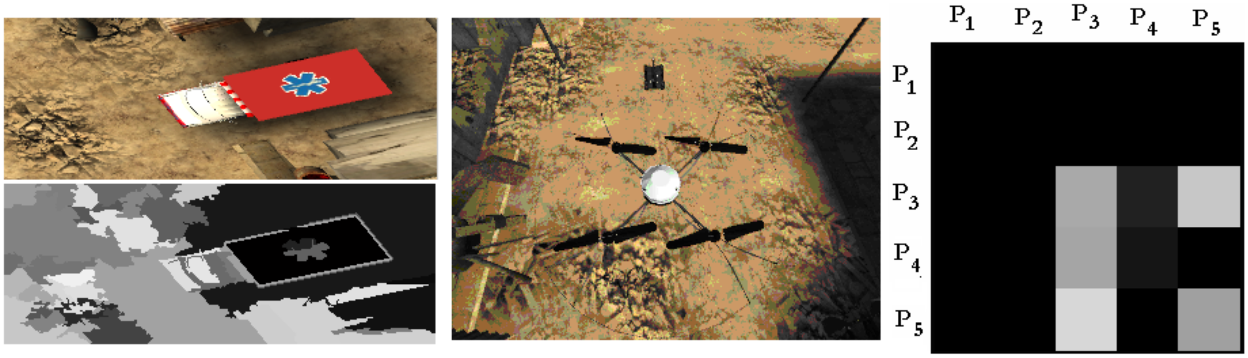

Fig. 3. Left: An object (ambulance) from the perspective of the aerial robot and (lower left) the resulting automatic segmentation of the image. The same object as seen from the ground robot can be seen in Fig. 4. Middle: Image showing the simulation environment, ground Talon robot, and aerial quadrotor robot. Right: Concept matrix for the race car object (aerial robot).

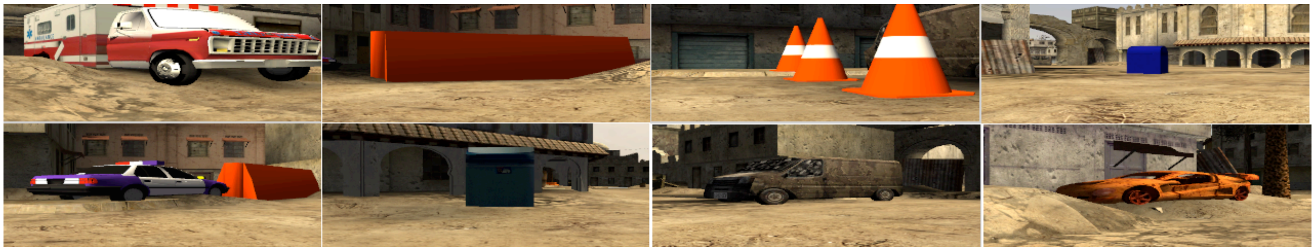

Fig. 4. All eight objects used for training and testing, as seen from the ground robot.

Eight objects were used for testing property and concept learning as well as transfer, all of which can be seen from the perspective of the ground robot in Fig. 4. A large number of images containing each of the objects in many different perspectives were gathered. For the van object, three different instances of the van in different lighting conditions were used to train the properties. The objects used were realistic, challenging, and were found under varied lighting. Out of these, 70 randomly chosen images were used for training, and 30 (different) randomly chosen images were chosen for testing. All images were automatically segmented using a graph-based image segmentation algorithm [17] (an example segmentation can be seen in Fig. 3).

B. Property Learning

In order to train the properties, each object was categorized as belonging into one of three color properties (see Table II). Both robots were trained with instances containing these properties, but the ground robot used an RGB color space while the aerial robot used an HSV space. The ordering of the properties were randomized for each robot. For texture, both robots used the same two dimensional space consisting of the mean and standard deviation output of a Gabor Filter. However, the ground robot only had one texture category corresponding to smooth objects such as the barricade or mailbox, while the aerial robot had an additional property corresponding to the texture pattern of the van (object 7 in Fig. 4). Hence, the ground robot had four total properties while the aerial robot had five. Note that not all objects were used in training of all of the properties (e.g. some objects were not used to train texture properties). While in simulation there is no heterogeneity in the cameras themselves, in this case heterogeneity originates from utilizing different metric spaces, one robot using an additional property that the other did not have, and large differences in perspectives which led to different portions of the objects being used during training.

For each property, 70 images of each object having that property were used to train the GMM. During training, a segment corresponding to the target property as well as the type of property being trained (e.g. color or texture) were hand-labeled. The target segment was then processed (e.g. median RGB, median HSV, or texture output values calculated for that segment), and the resulting set of data points were used for training the GMM. We used one cluster per property in this case. It is important to note that random symbols were assigned to these properties by each robot and were not given. For each property, only a group of images, the target segment per image, and the property type were given. Hence, the robots could not simply compare labels while learning property mappings, but instead had to utilize correlation statistics gathered in the form of confusion matrices.

TABLE II
PROPERTIES AND OBJECTS USED TO TRAIN CONCEPT LEARNING
(**Note: These were not given or used during training**)

Property # (ground/aerial)	Objects Incl.	Semantic Label
p_3 / p_1	Mailbox,Trash, Police Car	Blue
p_1 / p_2	Van	Brown
p_2 / p_3	Ambulance,Barricade Cone,Race Car	Red
p_4 / p_4	Mailbox,Barricade Police Car	Smooth
/ p_5	Van (3 types)	Textured

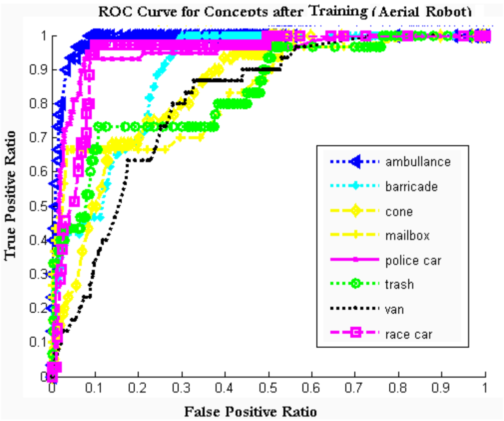

Fig. 5. ROC curve for eight concepts on aerial robot.

C. Concept Learning

After the property representations were learned, there was a second training period during which the concepts (i.e. objects) themselves were learned. Again, 70 images were used per concept along with a target segment that contained the target concept. As described previously, each concept was represented via a matrix containing correlations between each pair of properties. Fig. 3 show a gray-scale depiction of the matrix, where brighter values correspond to higher values (values range from 0 to 1, inclusive). As can be seen, high values were seen for property 3 (red) and property 5 (corresponding to textured as opposed to smooth). Correlations between properties 3 and 5 were seen as well, since whenever the race car had a high red property it also had a high value for the textured property (i.e. there was not much variety in the appearance of the race car).

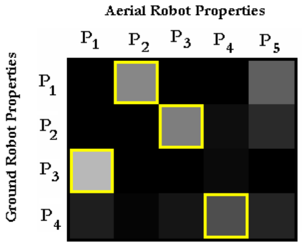

Fig. 6. Property mappings between the ground and aerial robots.

After the concepts were learned, we tested the accuracy of categorization of six hundred images, some of which contained the eight objects but many of which did not contain any of the learned concepts. In total, 600 images were used for testing, only 30 of which contained any one trained object. In this case, the process was fully automated: each image was segmented, the property membership values of each segment were calculated, and finally the concept membership values were calculated using Equation (5). Since each image contained 34 segments on average, this is a challenging categorization task as there can be many small segments that do not contain many pixels and spuriously lead to false positives.

Detection and categorization accuracy was determined using standard receiver operating characteristic (ROC) curves, which plots the true positive ratio against the false positive ratio. For each point on the curve, a threshold was used to decide whether a segment corresponded to a given concept or not, and as the threshold is loosened, more true positives are obtained but potentially resulting in higher false positives. The best possible classifiers would lie at the upper left corner, corresponding to only true positives and no false positives. One measure of total accuracy that can be used is the area under the ROC curve, where a value of one is perfect. Fig. 5 shows the curve for the aerial robot; the mean area under the ROC curve for all of the objects was 0.82 for the ground robot and 0.89 for the aerial robot, representing good categorization results given the challenges of using automatic segmentation and a large number of test images that did not contain any of the learned concepts. Overall the aerial robot categorized objects better, as HSV is a more effective color space and objects viewed from above vary much less than from below where occlusion and perspective differences can be a problem.

D. Property Mappings

We now demonstrate results for knowledge transfer, The first objective is to learn which underlying properties are shared between the two robots. In this paper, manual selection of image pairs (one from each robot) containing the same object was performed. However, since one robot was on the ground while the other was in the air, the perspectives were different. In the past, we have shown that this can be established using interaction such as following behaviors that are perceptually driven [4].

Fig. 6 shows the resulting matrix in gray-scale image format (Table 1 shows the numerical values). The ground truth mappings are highlighted and can be verified from Table II. The maximal values of each row correspond to the correct mapping, although there is an ambiguity between the first property of the ground robot (“brown”) and the fifth texture property of the aerial robot. This can be resolved in this case since it is less than the maximal property in the same row, but future work will look into mechanisms for disambiguating such potential false positives.

E. Concept Transfer

Finally, we look into actual concept transfer, as described in Section IIIE. We transfer all learned concepts from the ground robot to the aerial robot, and vice-versa, and test the resulting accuracy. Learned property mappings are used to modify the matrix as described previously. Note that the concept matrix is transferred, but during categorization the receiving robot’s own property memberships are used. Fig. 7 shows the resulting ROC curves for the aerial robot. Most objects were categorized with very similar accuracy, despite the robot never having seen any instances of the concept. The mean area under the ROC curve for the ground robot was 0.77 and 0.81 for the aerial robot, only slightly worse than when the robot learns itself. Interestingly, the order of accuracy between the objects sometimes changed.

Our last result shows that the robots can take the concept representation given to it by the other robot, and continue learning using new instances. We do this by averaging the received concept matrix with the newly learned concept matrix. With this additional learning, the accuracy of the

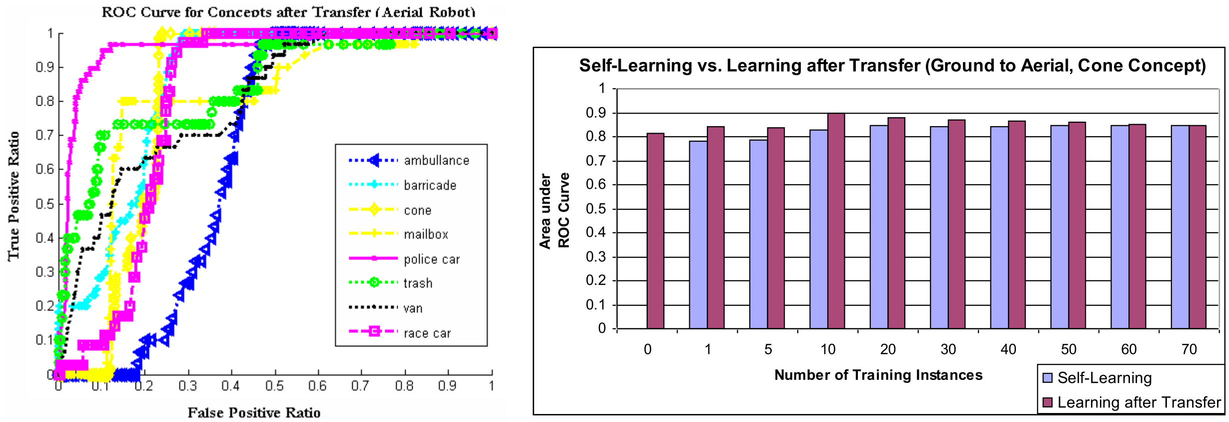

Fig. 7. Left: ROC curve for concept categorization after concept transfer. The concept matrices used are from the other robot entirely. Right: Area under the ROC curve for concept categorization accuracy as new instances are combined with the representation received from the other robot (cone concept).

robots closely approaches the accuracy had the robot learned the concept itself from the beginning. Fig. 7 shows the area under the ROC curve for one concept (cone), after an increasing number of new training instances are seen by the receiving robot. As can be seen, eventually the two performances almost converge. Note that for easier concepts where learning from a few instances results in high accuracy, the transferred concept does perform slightly worse than when the robot learns the concept itself. In all cases, though, the performance is comparable and if further training is performed the performances converge. This shows that a robot can receive a new concept from another robot (despite heterogeneity), successfully categorize the new objects almost as well as if the robot had learned the concept itself and then can continue learning. For harder concepts, such bootstrapping can allow the robot to perform well until it learns the concept itself.

V. Conclusions

In this paper, we have proposed a hierarchical representation based on conceptual spaces, where underlying properties are first learned and then used to train concepts. The power of this type of representation is that two heterogeneous robots can each represent properties (e.g. color or texture) via unique groundings specific to their embodiments and even in completely different spaces (e.g. RGB vs. HSV). If there is some overlap in the actual physical meanings to these properties, however, then robots can determine their mappings and transfer concepts among each other. Hence, robots can share knowledge regarding the appearance of objects and the receiving robot can classify entirely new objects. We have demonstrated results in this paper, in realistic 3D simulation experiments, that property mappings can be learned and concepts successfully transferred.

In this paper, we have used only a few properties, most of which were shared between the two robots. Future work will look into using a large number of properties, even from differing modalities (e.g. vision and SICK lasers), to accurately categorize a larger number of objects. Also, we have shown that successful concept transfer can occur when most of the properties were shared. In the future, we hope to characterize the level of similarity that is needed between the two robots in order to successfully transfer concepts.

References

[1] Harnad, S. (1990), 'The Symbol Grounding Problem', Physica D 42, 335-346.

[2] Vogt, P. & Divina, F. (2007), 'Social symbol grounding and language evolution', Interaction Studies 8(1).

[3] Jung, D. & Zelinsky, A. (2000), 'Grounded Symbolic Communication between Heterogeneous Cooperating Robots', Auton. Robots 8(3), 269–292.

[4] Kira, Z., Long, K., 'Modeling Robot Differences by Leveraging a Physically Shared Context', in Proceedings of the Seventh International Conference on Epigenetic Robotics, pp. 53-59, 2007.

[5] Kira, Z., 'Mapping Grounded Object Properties across Perceptually Heterogeneous Embodiments', accepted to the 22nd International FLAIRS Conference, 2009.

[6] Aisbett, J. & Gibbon, G. (2001), 'A general formulation of conceptual spaces as a meso level representation', Artificial Intelligence 133(1-2), 189–232.

[7] Balkenius, C.; Grdenfors, P. & Hall, L. (2000),'The Origin of Symbols in the Brain', Technical report, Lund University Cognitive Science.

[8] LeBlanc, K. & Saffiotti, A. (2007), 'Issues of Perceptual Anchoring in Ubiquitous Robotic Systems', in Proc. of the ICRA-07 Workshop on Omniscient Space.

[9] Steels, L. & Kaplan, F. (1999), 'Bootstrapping Grounded Word Semantics', in Briscoe, T., editor, Linguistic evolution through language acquisition: formal and computational models, pages 53-74, Cambridge University Press. Cambridge, UK.

[10] Billard, A. & Dautenhahn, K. (1998), 'Grounding communication in autonomous robots: an experimental study', Robotics and Autonomous Systems 1-2, 71-81.

[11] Gardenfors, P. (2000), Conceptual Spaces: The Geometry of Thought, MIT Press.

[12] Fowlkes, E. & Mallows, C. (1983), 'A method for comparing two hierarchical clusterings', Journal of the American Statistical Association 78(383), 553-569.

[13] Bilmes, J. (1998), 'A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models', International Computer Science Institute 4.

[14] Chella, S.C.; Frixione, M. & Saffiotti, A. (2004),Perceptual Anchoring via Conceptual Spaces, in 'Proc. of the AAAI-04 Workshop on Anchoring Symbols to Sensor Data', AAAI Press.

[15] Rickard, J. (2006), 'A concept geometry for conceptual spaces', Fuzzy Optimization and Decision Making 5(4), 311–329.

[16] Ionescu, A. (2006),Extension of the Concept Representation in Conceptual Spaces, in 'Fuzzy Information Processing Society, 2006. NAFIPS 2006. Annual meeting of the North American', pp. 296-301.

[17] P.F. Felzenszwalb and D.P. Huttenlocher, 'Efficient Graph-Based Image Segmentation', International Journal of Computer Vision, Volume 59, Number 2, September 2004.