Glover, Arren & Wyeth, Gordon (2011) Emergent Task-specific Object Semantics through Distributed Experience Networks. In *International Conference on Intelligent Robots and Systems (IROS 2011) Workshop on Active Semantic Perception and Object Search in the Real World*, 25-30 September 2011, Hilton San Francisco Union Square, San Francisco, CA.

Emergent Task-Specific Object Semantics through Distributed Experience Networks

Arren J. Glover, and Gordon F. Wyeth, *Member, IEEE*

***Abstract*— Autonomous development of sensorimotor coordination enables a robot to adapt and change its action choices to interact with the world throughout its lifetime. The Experience Network is a structure that rapidly learns coordination between visual and haptic inputs and motor action. This paper presents methods which handle the high dimensionality of the network state-space which occurs due to the simultaneous detection of multiple sensory features. The methods provide no significant increase in the complexity of the underlying representations and also allow emergent, task-specific, semantic information to inform action selection. Experimental results show rapid learning in a real robot, beginning with no sensorimotor mappings, to a mobile robot capable of wall avoidance and target acquisition.**

I. Introduction

THE semantics of an object stem from its purpose or use. While high-level object semantics can come from multiple domains, initial understanding of object use arises from the action-outcome relationships which occur through interaction. Given a task, it is these semantics that enable a robot to select an object from those available in the environment with which to interact.

A robot's sensorimotor coordination (SMC) links perception, action and outcome [1]. The basic object semantics are therefore grounded in the SMC system of the robot. If a robot is to autonomously acquire understanding of object semantics over its lifetime it must be able to autonomously develop its SMC.

Object semantics become important when acting in an environment in which multiple objects are present. However, in such environments, the high dimensionality which occurs due to combinations of different objects being simultaneously detected in the sensory field makes the task of developing the SMC non-trivial, even for a low degree of freedom robot with the task of navigating to useful objects.

The proposed Experience Network (EN) is a type of Markov Network which continually develops the robot's SMC over its lifetime. The EN captures sensory experiences in the nodes of the network, and temporal and motor information in the inter-nodal links. While network dynamics are similar to that of typical reinforcement learning [2], the research focus is on how the continual stream of sensorimotor data is efficiently organised to produce the SMC representations which capture the required object semantics.

Previous work demonstrates the development of SMC representations which allow various goal states to be achieved, when interacting with only a single object [3]. The work is extended into a domain in which the robot simultaneously detects multiple features from both foreground and background entities, requiring the formation of basic semantics in order to achieve a goal state. Three problems are considered: (1) how unchecked growth can be minimised when state-space size increases exponentially with state dimensionality, (2) how emergent semantic information about feature relevance can be captured and used to better inform action, and (3) how learning speed can be boosted through inferring actions from of novel states.

This paper presents an alternative to developing a network which stores the entire sensory state within a single node. Instead, each node is created with only a single sensory feature, and thereby distributes the state across multiple nodes in the network. This has three benefits: (i) the state space size becomes $O(N)$ with respect to the dimensions of the sensory features, rather than $O(c^N)$, (ii) the probabilistic dynamics of the Markov network can perform pattern generalisation and separation to efficiently generate more semantically driven sensorimotor coordination, (iii) inference of action from nodes can be more easily calculated as there is no ambiguity in the credit assignment resulting from groupings of features.

The paper proceeds by outlining related work in section II, before describing the details of the EN in section III. Studies using a real mobile robot are presented in section IV. Object detection is colour-based; however features are extracted from both the foreground and background entities. Results and discussions are presented in Section V and VI respectively, and Section VII describes future work.

II. Background

Autonomously building object models is usually performed by predefining the foreground [4-6], or the background [7], so the semantically interesting information is available to the robot. The problem of autonomously learning the relevance of a feature (which is task contextual, and may not be static) is less often considered.

Learning appropriate actions to achieve the desired goal

Manuscript received February 16, 2011. This work was supported in part by the Australian Research Council under a Discovery Project Grant DP0987078.

A. J. Glover and G. F. Wyeth are with the Queensland University of Technology, QLD Australia. Phone: +61731389974 Email: {aj.glover, gordon.wyeth}@qut.edu.au.

is often performed by finding object, action, outcome relationships in a second phase, after object representations have been formed [5]. Outcome learning has been performed by statistically averaging the change in object state, after an applied action, over multiple runs [7],[8] More continuous methods generally apply an uphill learning algorithm to optimise the performance of a single task [4]. To acquire relevant semantics over the lifetime of the robot the learning must occur simultaneously to actions being performed. The representations learnt must also be flexible in allowing different goals to be achieved given a change in task.

While Markov networks have successfully been employed in robot navigation scenarios in the past [9], research in reinforcement learning has shown that network structures have issues with computational complexities when confronted with large state spaces and high dimensionality [2]. However, other recent trends in object/affordance learning have shown that the dimensionality can be reduced by employing Bayesian network learning to capture the conditional dependence relationships [5], allowing the full state distribution to be estimated by modelling only the most causal relationships.

III. THE EXPERIENCE NETWORK

The Experience Network is a Markov network of sensory states which have been experienced by the agent and which are linked together through the actuation commands that were performed when the state changed. The sensory state at time t is referred to as an agent's experience e_t:

$$e_t = < V_t, H_t >$$

where V_t is a set of visual features and H_t is a set of haptic features. Each visual feature v_i in features V_t is extracted from the raw data and defined as:

$$v_i = < f_i, c_i >$$

where $f_i \in F_V$ is a description or label component of the feature and c_i is a component describing the position in the visual field. Similarly haptic features are defined by a label, selected from set F_h, and position in the haptic field.

The robot's actuation command is referred to as an action. The action a at time t is:

$$a_t \in$$

where the set A defines the set of all possible actions.

A. A Network of Experience and Action

The EN is realised using a graph structure in which the sensory experiences are stored within the nodes, N, and the actions are stored within the set of all links, L, that connect nodes (Fig. 1). The graph structure can then be exploited by forming an agent state from one or more nodes and following links towards a node with a desirable experience.

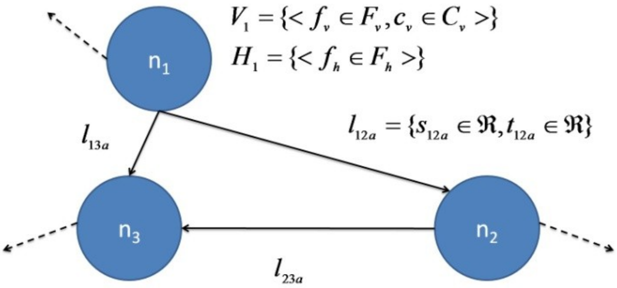

Fig. 1. The multi-dimensional EN projected in 2 dimensions. Each node stores sensory data while each link store transitional data. The dotted lines indicate links to experiences not shown.

Each node $n_i \in$ is representative of an experience:

$$n_i = < V_i, H_i >$$

where V_i are the features which describe the node visually and H_i are the features which describe the node haptically. A link connecting node i to node j by performing action a is defined as:

$$l_{ija} = < s_{ija}, t_{ija} >$$

where s_{ija} is the strength or repeatability of the link, and t_{ija} is the time to traverse between nodes.

B. Measures of Node Similarity

Nodes are added to the network by considering the information currently stored in the network and comparing it to the experience at time t. To perform a comparison a measure of the similarity of two nodes is required. The probability that node n_i is the same as node n_j is calculated as:

$$P(n_i | n_j) = P(V_i | V_j) P(H_i | H_j) \tag{1}$$

where the probability based on the visual element is calculated as:

$$P(V_i | V_j) = \prod_{y=}^{V} \ _{\in j} x[P(f_y | f_z) P(c_y | c_z)] \tag{2}$$

which makes the assumption that the feature label and position are independent. Independence is also assumed between feature labels; the label similarity is calculated as:

$$P(f_y | f_z) = \begin{cases} 0, f_y \neq f_z \\ 1, f_y = f_z \end{cases} \tag{3}$$

The probability due to position in sensory field is defined by a closeness measure which assumes that areas in the field are dependent on neighbouring areas:

$$P(c_y | c_z) = \frac{1}{1 + c_y - \ _z|} \tag{4}$$

The haptic component can be calculated in a similar fashion depending on the sensor arrangement. Due to simplistic haptic sensors of the current robotic setup, the haptic probability was simplified to:

$$P(H_i | H_j) = \begin{cases} 0, f_i \neq f_j \\ 1, f_i = f_j \end{cases} \tag{5}$$

C. Node Addition

The two methods for developing nodes and adding them to the EN, combination of features and distributed features, are now defined. In general both methods form the current robot state S_t, which is a subset of all nodes N, by selecting nodes already in the network or by adding new ones to it. New nodes are added when no current node similarity is above the threshold T_n.

Combining features in a single node is the direct extension to previous work, a single node is required to represent the current state and all features within the node must match to 'revisit' this node in the future.

Algorithm 1: Combination of Features

1. $S_t \leftarrow \ ; n_p \leftarrow \ ; n_{\max} \leftarrow \underset{n_i \in}{\text{ax}} P(n_i \mid n_p)$
2. **if** $P(n_{\max} \mid n_p) >$ $_n$
3. $\quad S_t \leftarrow$ $_{\text{ax}}$
4. **else**
5. $\quad N \leftarrow \ , n_p\} ; \ S_t \leftarrow$

The distributed method adds nodes to the network by forming individual nodes for each feature in the experience. Many nodes are then required to represent the current state; however a change in a single component feature does not shift the entire state to a new node, only the specific feature that changed.

Algorithm 2: Distributed Features

1. $S_t \leftarrow$
2. **for** $v_m \in$
3. $\quad n_p \leftarrow < v_m, H_t >; \ n_{\max} \leftarrow \underset{n_i \in}{\text{ax}} P(n_i \mid n_p)$
4. $\quad$**If** $P(n_{\max} \mid n_p) >$ $_n$
5. $\quad\quad S_t \leftarrow \ _t, n_{\max}\}$
6. $\quad$**Else**
7. $\quad\quad N \leftarrow \ , n_p\} ; \ S_t \leftarrow \ _t, n_p\}$

D. Link Addition

Links are added between nodes by considering the change in robot state from time t-1 to time t. For each node in S_{t-1} no longer present in S_t, a link is added to the most probable node in S_t.

Due to the closeness measure, $P(c_y|c_z)$, and the fact that the label probability, $P(f_y|f_z)$, is a binary measure, often the most probable node will be a nearby node with the same feature. This behaviour is designed to allow the modelling of feature motion through the visual field, however this is not an absolute 'rule' as noise in feature detection, as well as perceptual aliasing (the same feature can be in scene multiple times), will introduce uncertainty. Over time correct trends in motion should emerge. In cases in which no node has the same feature all other nodes become equally probable (at 0) and thus links are made to all nodes. Over time emergent behaviour, such as a feature changing label due to viewpoint change, can be captured.

Once links are chosen the current network links are updated by increasing the link strength, s_{ija}, by 1 and averaging the time component, t_{ija}, over all traversals.

E. Network Navigation

The graph representations formed in the EN are exploited to direct future agent action, closing the sensorimotor loop. Given the current agent state, S_t, the next action is selected so as to minimise the probable time it takes to arrive at a node that complies with a set goal criteria.

The probable time to a goal node given an action, $T(a_k)$, is calculated for every node in the network, given each action, using dynamic programming techniques [2]. Nodes which meet goal criteria are set to have a T value of 0 and every other node is updated according to:

$$T_i(a_k) = \sum_{n_j \in N} P_{ij}(a_k)(t_{ija} + \min_{a_r \in A}[T_j(a_r)]) \tag{6}$$

where $P_{ij}(a_k)$ is the probability of action a_k changing the state from node n_i to node n_j. $Pij(a_k)$ is calculated as:

$$P_{ij}(a_k) = P(n_j \mid n_i, a_k, L) = \frac{s_{ija}}{\sum_{n_k \in N} s_{ika}} \tag{7}$$

If action a_k has never been performed from node n_i (i.e. the denominator of (7) is 0) the time to goal is set as:

$$T_i(a_k) = \tag{8}$$

which instigates the exploration of unperformed links to new nodes. Alternatively the time can be inferred from other nodes as described in the following sub-section.

In the distributed feature method the current state is formed by multiple nodes; a greedy method is used to perform action selection in order to minimise the probable time to goal given all possible actions and all nodes in S_t:

$$a_t = \underset{n_i \in}{\text{nin}}(\min_{a_k \in}[T_i(a_k)]) \tag{9}$$

F. Inference

Many objects in the world have similar SMC behaviours, and hence will have similar network connections within the EN. To remove the need for complete exploration of a previously un-encountered feature's state-space, the time to goal, $T_i(a_k)$, can be inferred from already established nodes. If a successful match, based on partial link similarity, can be made between two different features, the entire state-space of that feature can be inferred resulting in reduced learning times. Feature similarities are more causally calculated in a distributed network as the links from a given features only represent the change in the change in a single feature, as opposed to many.

Two algorithms are introduced to perform inference. To recognise when two features exhibit similar behaviour, the inference measure I between the features f_u and f_v is continuously calculated as the network is developed:

$$I_{uv} = \frac{\beta}{\alpha} \tag{10}$$

where the values of α and β are updated each time a link is added. Given a link between node n_i and n_j is added with the action a_{t-1} the values of α and β (initially zero) are updated by finding similar links within the network based on the feature position c in pre and post nodes (Algorithm 3). The resulting cross-correlation matrix I defines the similarity in sensorimotor behaviour for all features in F.

Algorithm 3: Inference Calculation

1. **for** $n_g \in$
2. **if** $P(c_g \mid c_i) > T_n$
3. $\alpha \leftarrow \alpha +$
4. **for** $n_h \in$
5. **if** $P_{gh}(a_{t-}) > 0.5$ **&** $f_h = f_g$ **&** $P(c_h \mid c_j) > T_n$
6. $\beta \leftarrow \beta +$

The similarity between features is used when calculating probable times of each node for network navigation. If an appropriate inferred node can be found when action a has never been performed from node n_i the time is, instead, inferred.

Algorithm 4: Inference Usage

1. **for** $n_i \in$, $a \in$
2. **if** $\sum_{n_k \in N} s_{ika} = 0$
3. **for** $n_j \in$
4. **if** $I_{ij} > T_I$ **&** $P(c_j \mid c_i) > T_n$
5. $T_i(a) \leftarrow (a)$

IV. Experiments

A. Robot and Environment

Experiments were performed on a Pioneer3-DX mobile robot with a 2-DOF gripper including an IR break beam between the paddles for haptic sensing. A forward facing Logitech Webcam Pro 9000 was used as visual input (Figure 2).

Fig. 2. The robot in the experimental 3x3m walled environment. The red, and other (green) markers, can be detected using the gripper while the wall and floor cannot.

The robot's experience consisted of visual features from the camera and haptic data from the gripper. Colour-based image segmentation was employed as illustrated in Figure 3. Although colour segmentation is a simple way to extract useable features from the environment the experiments are only initial studies in which no distinction between foreground and background is made, the robot had to develop its own semantics about which features where important to achieve its goals.

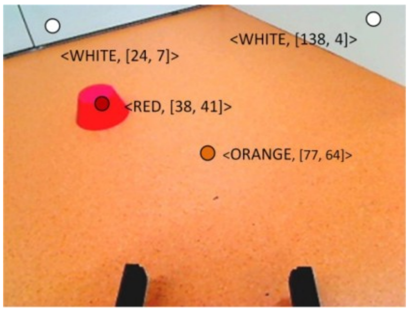

Fig. 3. The visual features extracted from a typical scene. The image is colour segmented allowing features to be generated by the walls, the floor, and the red and green markers. Each visual experience consists of the visual features f_v and the centroid c_v.

The elementary actions available to the robot were forward, left, and right and selected at a rate of 10 Hz. Laser range data was used to detect when robot actions would lead to a collision, any detection would stall the robot before collision occurred. The network was built with the node similarity threshold, T_n, set so nodes cover a region with a ~20 pixel radius (in a qqVGA image) and the inference threshold, T_I, was set to 30%.

B. Experimental Procedure

Each run was conducted with the robot in an initial position and the markers randomly placed within the robot's field of view. The robot began with an empty EN, hence no understanding of the sensory-motor mappings. The goal criterion was set to be nodes with the IR break beam triggered. Robot and marker positions were reset upon reaching a goal state. Two EN's were developed based on both Algorithm 1 and Algorithm 2, however only the network formed using Algorithm 2 was used to control the action of the robot.

Study 1: 20 runs were performed with a single red marker in the arena. The study aim was to demonstrate that the distributed EN could be successfully exploited to solve the target acquisition problem in the face of multiple different background/foreground sensory input.

Study 2: A further 20 runs were performed, continuing the use of the EN from Study 1. An additional red marker was added and the aim was to investigate performance and representation size when two distinctive (goal) features were present.

Study 3: A final 10 runs were performed with only a single green marker in the arena. The study aim was to investigate the utility of inferring novel feature behaviour from known feature representations.

V. Results

A. Node Addition Method Comparison

The distributed network (DN) led to a smaller network size than the combined features network (CFN) as can be seen in Figure 4. The CFN size grew significantly larger than the DN due to the state-space dimensionality; each feature in each position was an orthogonal axis in the state space. It was from this evidence that the CFN was disregarded in further analysis; the time and physical exploration required to develop worthwhile representations was exponentially larger than the DN.

The DN growth supports the method's advantages: it slowed in the first study as a reasonable amount of the state space was physically explored, was not required to grow when a second marker was added, and only required new nodes to represent the green marker in the third study. The CFN continually increases in size over all three studies as the state-space is exponentially larger. It is only in the third study when growth slowed, due to the traversal of similar paths by the (action selecting) DN.

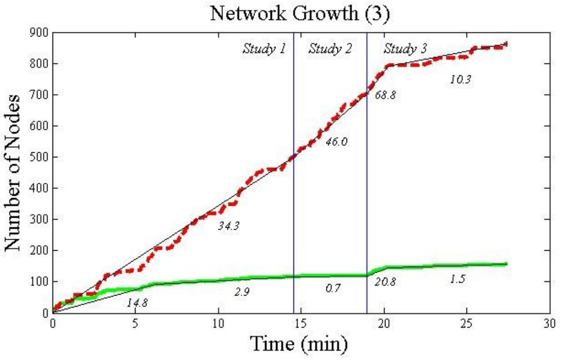

Fig. 4. The state-space size comparison between the CF network (dotted) and IF network (solid). The slope is reported in units of nodes/minute.

B. Study 1: Grounding Multiple Sensory Features

The initial trial took over 5 minutes as the robot filled out the empty EN with newly acquired sensorimotor experience. Subsequent runs, even with the marker in a random position, showed the effectiveness of the DN to acquire the task specific semantic information required to complete the task, as the average time dropped to 28s. It learnt that the marker feature is the most important feature to use given the haptic goal and in most cases the wall and floor features are not helpful.

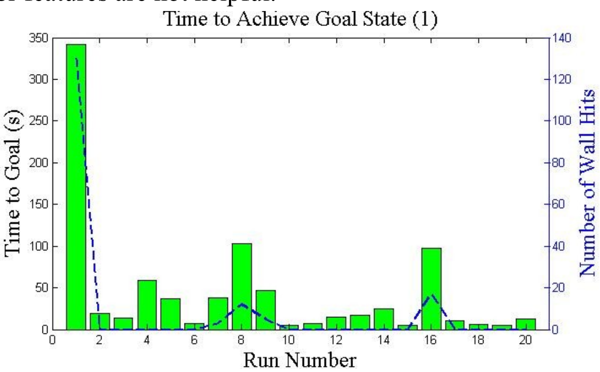

Fig. 5. The time to goal for each run in study 1. The dotted line indicates the number of wall 'hits' as indicated on the right axis.

When the robot cannot see the informative marker feature, goal directed behaviour is not possible. There is no egocentric information available to the robot to make a prediction about the relative location of the target unless it is in the field of view. However the robot would learn to turn rather than drive straight at the wall, although at times it would prevaricate between turning left and right without gaining sight of the target, as evidenced in runs 8 and 16. These runs also show an increased number of wall hits as the robot learnt more about the semantics of the wall as a feature. The ability to avoid walls is learnt as can be seen in the decreasing number of wall hits.

C. Study 2: Navigating with Multiple Goal States

The already grounded network was able to continue to achieve a goal state when a second marker was added to the environment (Figure 6). Very few new nodes were added to the EN in this study (Figure 4), hence the SMC state representations were already adequate for correct behaviour with links continually being reinforced. Due to the greedy algorithm the closest node could be selected as the goal target only further reducing the time to goal. In contrast the CFN network would have to develop new representations to represent the combination of 2 marker features.

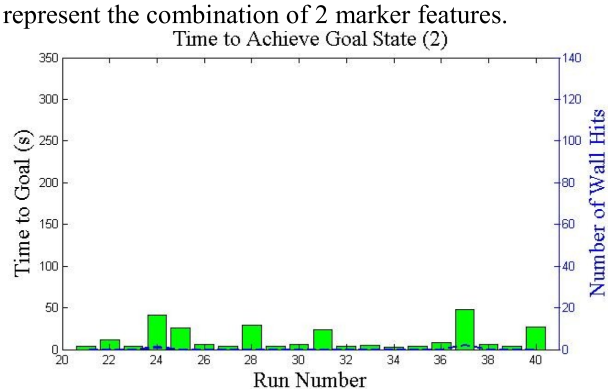

Fig. 6. The time to goal and wall hits for each run in study 2.

D. Study 3: Inferring Novel Feature Utility

While inference between features was being calculated and, if necessary, invoked throughout all studies it was only in the third study in which they came into effect. The walls floor and marker all had unique movement behaviour with respect to their centroids and hence no inference was performed. When the green marker was introduced (producing a previously unseen feature) it was closely matched with the already developed red marker nodes (Table 1), allowing correct actions to almost immediately be performed to achieve the goal state (Figure 7).

The long run duration in runs 45 and 48 were caused by the incorrect association of the green marker with the wall features. In both runs the marker was positioned towards the top of the visual image, where typical wall behaviour is to turn, rather than driving forwards. Action was then incorrectly inferred and hence direct motion to the marker was not performed. The red marker had a zero inference value from the green marker as the red marker was never reintroduced after the green marker was introduced.

TABLE 1
MOTION INFERENCE BETWEEN FEATURES

	Wall	Floor	Red Marker	Green Marker
Wall	48.45%	29.41%	17.31%	58.73%
Floor	4.35%	19.25%	10.20%	15.40%
Red Marker	14.81%	5.70%	47.26%	0.00%
Green Marker	32.79%	4.50%	33.21%	58.89%

The probability that a feature (rows) has the same sensorimotor behaviour as another (columns). Auto-inference is not 100% as the probability only increased when $P_{ij}(a) > 0.5$.

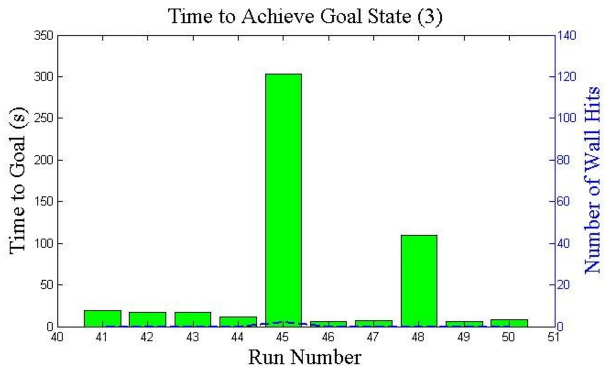

Fig. 7. The time to goal and wall hits for each run in study 3.

VI. DISCUSSION

The Distributed Experience Network algorithm differs from other methods of learning sensorimotor coordination in a number of important ways.

A. Learning is One-Shot and On-Line

While the EN develops an adaptable SMC from scratch, the network complexity is kept manageable by having each node deal only with a single sensory feature from an experience, representing the experience in a distributed fashion. The EN does not require separate learning and recall phases. All of the robot's interactions with the environment result in learning, allowing the robot to continually update its SMC over the lifetime of the robot

B. Attention is Intrinsic to the Network

Our experiments used colour segmentation to simplify the incoming visual information, but no specification was made as to which features were in the foreground, and which were in the background. There is no inherent attention operator to highlight features of interest – rather the EN develops task specific semantic information by noting which sensory changes occur consistently with motor action. Only the foreground features (i.e. from the object) are recognised as informative.

C. Learning can be Boot Strapped by Inference

Bootstrapping knowledge between features is important when using state-action representations especially in high dimensional space, as otherwise each state of each feature needs to be explored and grounded. The distributed network more easily allows for dependencies to be learnt which are then exploited to reduce the amount of learning the network has to do before appropriate actions are emergent.

VII. FUTURE WORK

Adapting these studies to a real-world environment requires the introduction of appearance-based features, such as SURF [10]. The large increase in the number of features which must be processed forms the key challenge of doing so. While the individual features can be picked out in the DN, a single object can produce multiple features in this scenario. It would then seem viable to allow small groupings of features to autonomously form in nodes thereby reducing the number of features to be processed, while also demonstrating emergent 'object recognition'.

Bayesian network theory could also be employed to increase the robustness of inference between variables in the system, such as in [5]. Not only does this speed learning but also allows reduction of the dimensionality of the system if new variables were added, such as global X, Y coordinates. This would allow the robot to then search for objects outside the immediate view, and also, in a similar method to visual features, develop task-specific spatial semantic information.

ACKNOWLEDGMENT

The authors would like to thank the other members of the Lingodroid group Dr Ruth Schulz and Prof. Janet Wiles.

REFERENCES

[1] R Pfeifer and C Scheier, "Sensory-motor coordination: The metaphor and beyond", Robotics and autonomous systems, 20(2-4):157-178, 1997

[2] RS Sutton and AG Barto, Reinforcement learning: An introduction. The MIT press, 1998

[3] A. Glover, R. Schulz, G. Wyeth and J. Wiles, "Grounding Action in Visuo-Haptic Space using Experience Networks", In The Australasian Conference on Robotics and Automation, Brisbane, Australia, 2010

[4] Christian Scheier and Dimitrios Lambrinos, "Categorization in a real-world agent using haptic exploration and active perception", Simulation of Adative Behaviour, pages 65-75, 1996

[5] L. Montesano, M. Lopes, A. Bernardino and J. Santos-Victor, "Learning Object Affordances: From Sensory--Motor Coordination to Imitation", IEEE Transactions on Robotics, 24(1):15-26, 2008

[6] J. Sun, J.L. Moore, A. Bobick and J.M. Rehg, "Learning visual object categories for robot affordance prediction", The International Journal of Robotics Research, 29(2-3):174, 2010

[7] J Modayil and B Kuipers, "The initial development of object knowledge by a learning robot", Robotics and autonomous systems, 56(11):879-890, 2008

[8] P Fitzpatrick and G Metta, "Grounding vision through experimental manipulation", Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 361(1811):2165, 2003

[9] S. Koenig and R. Simmons, "Xavier: A robot navigation architecture based on partially observable markov decision process models", Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems, 91–122, 1998

[10] H. Bay, T. Tuytelaars and L. Van Gool, "Surf: Speeded up robust features", Computer Vision–ECCV 2006, 404-417, 2006