Proto-Symbol Emergence*

Karl F. MacDorman, Koji Tatani, Yoji Miyazaki,† and Masanao Koeda‡
Department of Systems and Human Science
Graduate School of Engineering Science
Osaka University
Toyonaka, Osaka 560-8531 JAPAN
http://robotics.me.es.osaka-u.ac.jp/~kfm

Abstract

Robotics can serve as a testbed for cognitive theories. One behavioral criterion for comparing theories is the extent to which their implementations can learn to exploit new environmental opportunities. Furthermore, a robotics testbed forces researcher to confront fundamental issues concerning how internal representations are grounded in activity.

In our approach, a mobile robot takes the role of a creature that must survive in an unknown environment. The robot has no *a priori* knowledge about what constitutes a suitable goal — what is edible, inedible, or dangerous — or even its shape or how its body works. Nevertheless, the robot learns how to survive. The robot does this by tracking segmented regions of its camera image while moving. The robot projects these regions into a canonical wavelet domain that highlights color and intensity changes at various scales. This reveals sensory invariance that is readily extracted with Bayesian statistics. The robot simultaneously learns an adaptable sensorimotor mapping by recording how motor signals transform the locations of regions on its camera image. The robot learn about its own physical extension when it touches an object. But it also undergoes an internal state change analogous to the thirst quenching or nausea producing effects of intake in animals. This allows the robot to learn what an object affords — is it edible or poisonous? — by relating these effects to learned clusters of invariance. In this way primitive symbols emerge. These proto-symbols provide the robot with goals that it can achieve by using its sensorimotor mapping to navigate, for example, toward food and away from danger.

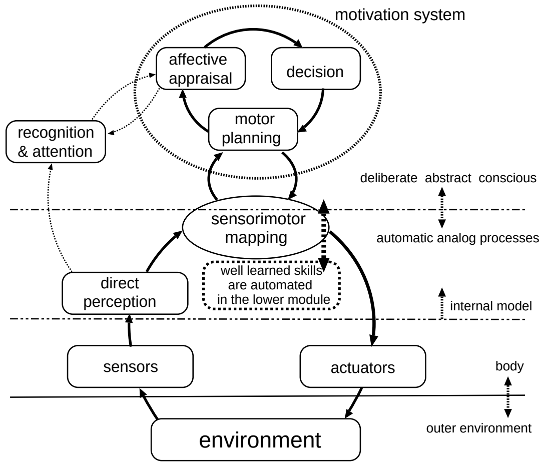

Figure 1: The design for Ψro (from *psy*che + *ro*bot), a robot that exploits affordances.

1 The Motivation for Symbol Emergence

Even one-celled animals are able to make distinctions, detecting the presence or absence of light or chemicals so as to climb a gradient to a food source. But whereas every other species lives in its niche, *Homo sapiens* alone live in a world of their own making. They adapt to the environment by adapting the environment to themselves. They can *dis*embed themselves, shift perspectives, stand back from the here and now, and ponder futures unseen.

These abilities are often associated with language and especially the power of a word or symbol to stand in for something that is absent. Though the warning call of a vervet monkey can stand in for the perception of a predator [4], language is more than that. It is a kind of *system* of distinctions or differences [29]. While as practised it has aspects that are messy and probabilistic, its syntax is for the most part productive, systematic, and inferentially coherent [7].

Inferential coherence refers to the fact that operations can be performed on sentences in such a way that, if the original sentences correspond to true states of affairs (e.g., Socrates is a man; and all men are mortal), the resulting sentences or conclusions also correspond to true states of affairs (Socrates is mortal). Of course, since Aristotle invented the syllogism, it has been known that a representation's syntax can encode its role in inference. What is different today is that we have computers capable of automating this process. This has strengthened the view that the mind — like language — is a kind of symbol system [26].

The traditional AI approach to constructing a symbol system involves a programmer determining a set elementary symbols and rules for combining and manipulating them [19, 30]. The symbols may be manipulated deductively [20] or procedurally [6]. In the lat-

*A grant from CREST of JST and a grant from the Ministry of Education of Japan (No. 12750215) help support this research. Java is a trademark of Sun Microsystems.

†Yoji Miyazaki is now at NEC, Corporation.

‡Masanao Koeda is now a Master's student at Nara Institute of Science and Technology.

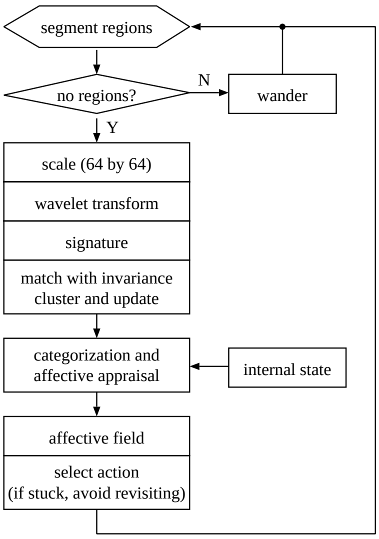

Figure 2: Ψro produces compact wavelet signatures that correspond to segmented regions. The robot learns invariance clusters that match against these regions. Making contact with a region that corresponds to a certain kind of object (e.g., edible, poisonous, dangerous) results in an internal state change. The robot identifies this change with its matching clusters. Thus, proto-symbols emerge as the robot learns to categorize objects by their internal effects (and visual similarities). The robot appraises recognized objects in terms of its current internal state (e.g., hungry, horny, bored). In this sense they correspond to different kinds of affordances. Their relative location and hedonic value (given the robot's current internal state) determine a hydrodynamic potential field in the robot's sensorimotor phase space, which the robot attempts to minimize by moving toward desirable objects while avoiding obstacles and dangerous objects (e.g., a predator or a cliff).

ter case, operators transform state descriptions. Thus, the application of an operator is meant to be analogous to taking an imagined action during planning.

Although it is possible to embody a symbol system in a robot so that objects instantiate symbols and symbolic operations cause the robot to act on those objects, this is a rather weak form of embodiment. This is because a symbol system's symbol manipulation obeys symbol-rule relations that are *a priori* and internal to the system. The "system of distinctions" may set up an infinite space of possibilities, but the distinctions and the possibilities they circumscribe are fixed. But it has become increasingly clear to even proponents of symbol systems that symbol-object relations must be brought to bear on symbol manipulation [8, 11, 12, 16]. These relations are sensorimotor; they depend on having a particular body, and they change as that body or its environment changes. Sensorimotor relations influence what the environment affords — what Gibson called *affordances* [9] — and the kinds of sensory invariance on which any distinction can be made. Thus, they constrain not only what can be done but what can be perceived. Taken together

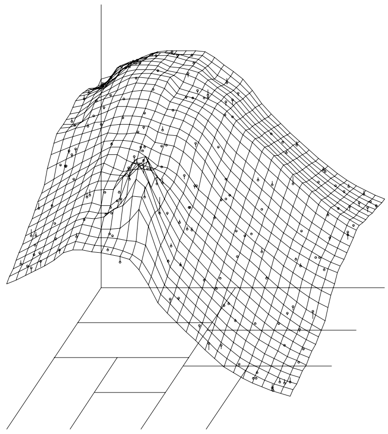

Figure 3: A 2D-to-1D mapping with partition nets after only a few second of learning. The coordinates vary from 0 to 1. Circles represent data points. All points lie on f. The length of the line above or below a circle gives the prediction error $f - g$.

with cognitive and perceptual limitations, sensorimotor relations delimit the space of potential distinctions, and it is within this space that even the most abstract forms of reasoning must occur.

To take account of these facts in developing reasoning machines, we propose an approach to emerging symbols from the bottom up. A mobile robot learns a sensorimotor model from experience so that it can make predictions about the consequences of its actions (§2). This involves learning to recognize affordances (§3) and the actions needed to exploit them. For a fairly low-level task like robot navigation, this may work out to knowing *what* is *where*. But notions of object and distance must be discovered by the robot itself based on how its motor signals transform both internal indicators of well-being and the information that external objects project on to its camera image planes and other sensors [15]. Once a robot has learned to recognize affordances, it can use its sensorimotor model to obtain goals.

2 Sensorimotor Learning: Partition Nets

A robot may need to respond immediately to unexpected changes in its sensorimotor relations. Unlike neural networks, closest point methods are capable of learning immediately from single instances. However, they may give slower predictions and poorer generalizations, especially in high dimensional phase spaces. Since thousands of predictions may be required in the course of making a single plan and prediction errors can easily accumulate, we need an algorithm that can combine the strengths of closest point algorithms and neural networks.

Partition nets [17] are an efficient on-line learning algorithm that can make fast predictions about well-practised movements while quickly adapting to changes in sensorimotor relations. The algorithm is interesting from a cognitive standpoint because it mirrors certain aspects of how humans learn. For example, in learning to ride a bike, we progress from reason-

function $\mathrm{g}(\vec{\imath}, \Re)$ {
  if ($\Re$ is a *leaf node*) {
    if ($\mathcal{E} < \epsilon_1$) return $\mathtt{network}(\vec{\imath})$;
    else return $\mathtt{gblend}(\vec{\imath}, (\vec{m}_1, \ldots \vec{m}_N))$;
  } else { /* $\Re$ is an *internal node* */
    $dp \leftarrow \vec{\imath}_d - p$; /* distance to partition plane */
    $blendzone_L \leftarrow \kappa\ \mathtt{width}(\Re_L, \vec{\imath}, d)$;
    if ($dp \leq -blendzone_L$) return $\mathrm{g}(\vec{\imath}, \Re_L)$;
    $blendzone_H \leftarrow \kappa\ \mathtt{width}(\Re_H, \vec{\imath}, d)$;
    if ($dp \geq blendzone_H$) return $\mathrm{g}(\vec{\imath}, \Re_H)$;
    else { /* $\vec{\imath}$ is in the blend zone, so blend */
      $\vec{o}_L \leftarrow \mathrm{g}(\vec{\imath}, \Re_L)$;
      $\vec{o}_H \leftarrow \mathrm{g}(\vec{\imath}, \Re_H)$;
      $\omega \leftarrow \frac{(dp + blendzone_L)}{(blendzone_H + blendzone_L)}$;
      $\forall k \in \{1 \ldots K\}, (\vec{o})_k \leftarrow (1 - \omega)(\vec{o}_L)_k + \omega(\vec{o}_H)_k$;
      return $\vec{o}$;
    }
  }
}

Listing 1: Pseudocode for function $g(\vec{\imath})$.

ing from memories of particular events or instructions to a kind of automatic response that demands little conscious effort or attention.

For simplicity we may view sensorimotor relations as a function from an input state $\vec{\imath}_t$ at time t to an output state $\vec{o}_{t+1}$ at time $t+1$: $\vec{o}_{t+1} = f(\vec{\imath}_t)$, where the robot's perceived state $\vec{s_t}$ and motor signals $\vec{a_t}$ at time t determine $\vec{\imath}_t$, and its perceived outcome state determines $\vec{o}_{t+1}$. To give a concrete example, the robot Ψro sees an object at position (x_t, y_t) on its camera image planes, and it needs to predict the next location of the object (x_{t+1}, y_{t+1}) after it has turned its right and left wheel by (l, r) (see Figure 5). Therefore, it needs to learn a mapping $g(\vec{\imath})$ that approximates f.

As Ψro move about, partition nets memorize input-output pairs $(\vec{\imath}_t, \vec{o}_{t+1})$, unless they are able to predict $\vec{o}_{t+1}$ from $\vec{\imath}_t$ with enough accuracy. At first, a prediction about the consequence of an action in some new state is based on a Gaussian blending of range values for pairs that are near to the vector $\vec{\imath}_t$ in the domain. Meanwhile, a neural network trained during spare CPU cycles using the backpropagation learning algorithm [28]. When the predictive accuracy of the network eclipses that of the blending functions, partition nets switch over to using the network for prediction.

If the number of memorized input-output pairs reaches a threshold, the input space is partitioned and two networks are formed — one whose receptive field covers the subspace on the low side of the partition, and the other whose receptive field covers the subspace on the high side. The input-output pairs are also partitioned into two matching subspaces. This partitioning concentrates network weights in more complex or often explored areas of the mapping where they are needed most. It also facilitates the rapid look-up of nearby points when using Gaussian blending [27, 33].

Partition nets set the partition at the mean value of the dimension with highest mean absolute deviation:

$$mean_k = \frac{\sum_{i=1}^{B} (\vec{\imath}_i)_k}{B}$$
$$\max_k \sum_{i=1}^{B} |(\vec{\imath}_i)_k - mean_k|$$

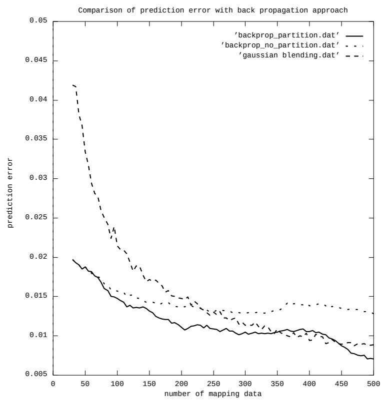

Figure 4: After learning converges, backprop with partitioning gives excellent results.

The mean absolute deviation is usually more robust than the variance and, unlike the variance, requires no multiplications. This method requires on the order of B operations, where B is the size of the bucket.

Since input values vary from 0 to 1, if the input space has been partitioned many times, the neural networks will only receive a small range of input values. This will greatly slow learning. The input to the network is scaled to match its shrunken receptive field and high and low subspace networks are adjusted so that their weights remain applicable.

The prediction algorithm. Multidimensional search trees [1, 33] provide an ideal data structure for hierarchically partitioning subspaces. Each node in the tree corresponds to a subspace. Internal nodes have a discriminator dimension d, partition p (which indicates where dimension d is partitioned), and a pair of pointers, one that point at the node's low child $\Re_L$ and the other that points at its high child $\Re_H$ (see [33]). Partition nets extend the KD tree structure so that, in addition to a pointer that points a subspace's bucket of points, each leaf node also a pointer that points at its associated neural network.

If for any input point $\vec{\imath}$ only the network or closest points in one leaf node's subspace are used for approximating the output, the function $g(\vec{\imath}_t, \Re)$ will be discontinuous at subspace boundaries. Regardless of whether the subspace is approximated by a neural network or Gaussian blending of M closest points, values at these boundaries will generally be less accurate than in the central area of the subspace because these values are extrapolated. Therefore, the output $\vec{o}$ near boundaries is calculated from a linear blending of the outputs of the two neighboring subspaces.

If $\Re$ is the subspace of a leaf node, $g(\vec{\imath}, \Re)$ returns an estimate of the perceived outcome state $\vec{o}_{t+1}$ (see Listing 1). The estimate of $\Re$'s neural network is returned if it is accurate enough ($\mathcal{E} < \epsilon_1$). Otherwise, the estimate is determined by $\mathtt{gblend}(\vec{\imath}, (\vec{m}_1, \ldots \vec{m}_N))$, which collects in a heap the M closest points in $\Re$'s bucket and blends them using the Gaussian function, as explained above. If $\Re$ is the subspace of a leaf node, its distance to the partition plane dp is calculated. The function width, which may recursively call itself, will find $\vec{\imath}$'s leaf node subspace and return its width along the dimension d. So a blend zone is calculated which begins in the low subspace $\Re_L$ some fraction κ (e.g., 0.2) of $\Re$'s width below the partition plane and ends in the high subspace $\Re_H$ some fraction κ of $\Re$'s width above the partition plane (κ must fall in the range $[0, 1]$). If $\vec{\imath}$ is on the low side of the blend zone, no blending at this partition is required, and $g(\vec{\imath}, \Re_L)$ is called recursively for $\Re_L$. Likewise, if $\vec{\imath}$ is on the high side of the blend zone, $g(\vec{\imath}, \Re_H)$ is called recursively for $\Re_H$. Otherwise, it is necessary to obtain the results of both $g(\vec{\imath}, \Re_L)$ and $g(\vec{\imath}, \Re_H)$ and blend them. The blending is linear across the width of the blend zone (i.e., an affine combination with linear influence weights).

The reader may explore partition nets further with the aid of a Java™ applet. It is available at http://robotics.me.es.osaka-u.ac.jp/~kfm/#tutorials

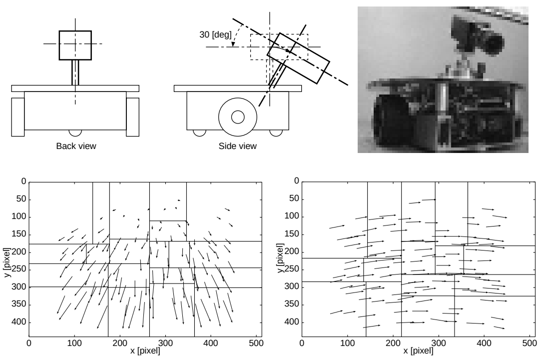

Figure 5: The camera geometry of the robot and the corresponding learned flow vectors of objects for moving forward or turning leftward in place.

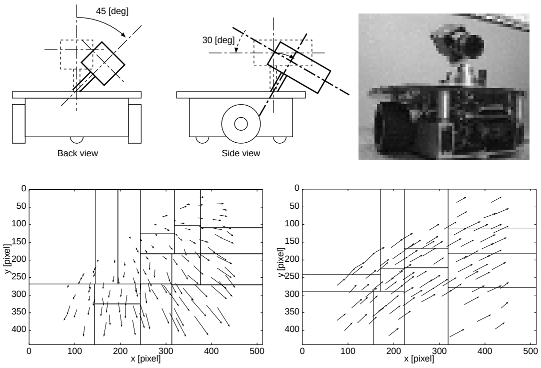

Figure 6: The camera geometry and the relearned flow vectors after tilting the camera 45 deg. in the roll axis.

2.1 Results

Since partition networks can immediately give a good approximation based on Gaussian blending, learning converged orders of magnitude faster than for ordinary backprop. Partition nets gave better predictions than either Gaussian blending alone or backprop with or without a partitioning of the input space in the simulation in Figure 3. After a few seconds backprop was able to replace Gaussian blending in most subspaces (viz., the smoother ones).

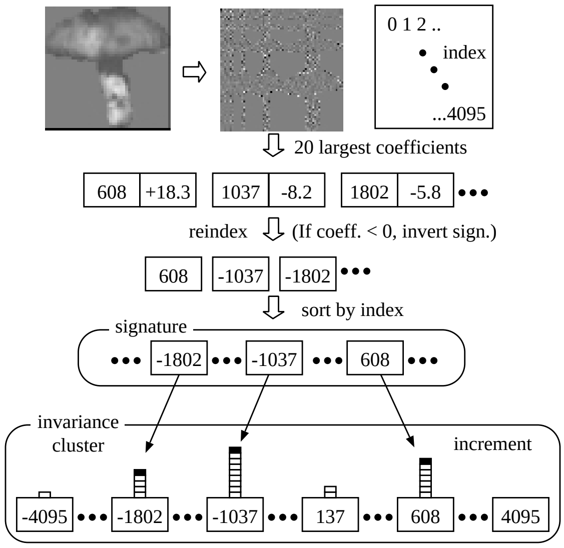

Figure 7: Y, U, and V images of segmented regions are scaled and projected into the normalized Haar wavelet domain. This highlights invariance, since wavelet coefficients are sensitive to differences in the lightness and color of the image at varying resolutions. A compact signature is formed from wavelet coefficients that are highest in absolute value. These signatures are used to learn invariance clusters, which serve as naive Bayesian classifiers (see text).

It was useful to evaluate the component processes of partition nets individually. Results from 200 new trials show that on average Gaussian blending uniformly gave better predictions than linear blending regardless of the number of learned data points. Another figure (see website) shows that output from *eight* closest points provided the best prediction for Gaussian blending if the sample noise was low, but more points gave a better result as the noise increased. Figure 4 shows that after 20,000 learning cycles, backprop with partitioning uniformly gave better predictions than ordinary backprop and that it almost uniformly outperformed Gaussian blending. (Of course, partition nets give better performance still, because they can choose whether to use backprop or Gaussian blending for each subspace.) Partition nets gave good predictions on real (noisy) image data from Ψro.

2.2 A Sensorimotor Adaptation Experiment

The sensorimotor mapping relates motor signals that control two independently driven wheels to flow vectors that signify changes in the visual projections of objects in the robot's camera image. Presently, we use the Hitachi IP5005 color image processing board to remove the background and calculate the subcentroid of an object (its lowest point under the centroid) because this offers a reasonable distance estimate for many kinds of objects placed on a flat surface, given that our robot is only monocular.

The input to the sensorimotor mapping is the location of the i'th object in the camera image (x_t^i, y_t^i) and a motor signal (l_t, r_t) for turning the wheels. The output $\vec{o} = (x^i_{t+1}, y^i_{t+1})$ is the location of the i'th object in the camera image after the robot has taken an action. If the movement vector of an object resulting from a motor signal was inaccurate, the robot replaces nearby flow vectors with the newly experienced flow vector. Thus, it adapts to sensorimotor change only where necessary by relearning.

We verified the usefulness of these method in an experiment. The robot first learned a sensorimotor map by recording the movement vectors of the subcentroids of several objects (Figure 5). We then tilted the robot's camera 45 degrees in the roll axis. Although the robot could not initially predict the objects' movements accurately, it quickly relearned the mapping so that it could again make accurate predictions (Figure 6). This demonstrates the flexibility of the learning method, and how it addresses issues concerning embodiment such as a change in sensor geometry.

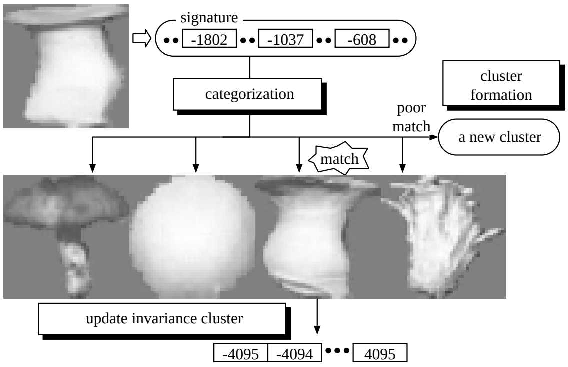

Figure 8: The row showing four species of mushrooms — *shiitake*, closed cup, *eringi*, and *enoki* — symbolize four invariance clusters. Each cluster acts as a naive Bayesian classifier, and the signature of a newly segmented and scaled region is matched against the classifier with the highest probability estimate. If the probability is low that the signature matches against any of the clusters, or if a signature is miscategorized, a new cluster is tentatively formed.

3 Learning to Recognize Affordances

Segmentation and tracking. Ψro's goal is survival. It must intercept tasty objects and avoid poisonous and dangerous ones in a dynamic environment. The robot learns to do this through a process of feature extraction and classifier induction (cf. [2, 3]). A Hitachi IP5005 image processing board is used to segment regions by removing the wall and floor using color information.

Image preprocessing and feature extraction. The processing board next converts segmented regions to a canonical form. This highlights invariance and facilitates comparison between different segmented images. The process involves *(1)* separating segmented regions into separate Y, U, and V images; *(2)* scaling the regions to fit on a 64-by-64 grid, *(3)* decomposing the scaled region into a set of wavelet coefficients, and *(4)* quantizing the coefficients, retaining only those that are largest in absolute magnitude, to form a Y, U, and V signature for each segmented region.

The wavelet transform and other multiresolution techniques are useful because it is often hard to find invariant features at a single characteristic scale. We chose the Haar discrete wavelet transform for its speed, and normalized the basis in relative, but not absolute, terms. The normalized Haar basis proved to be more robust because it does not over-emphasize detailed coefficients, which tend to be sensitive to the sharp discontinuities along region boundaries created by the segmentation process.

To generate the signature, the indices for each image array of the N wavelet coefficients that are largest in absolute magnitude are collected in a heap data structure (e.g., $N = 20$). The indices are given the sign of their corresponding coefficient and, once sorted, form a compact signature (Figure 7).

Learning invariance clusters and categorical representations. As Ψro tracks multiple regions, invariance clusters count the signed indices of matching wavelet signatures. Low-level miscategorization is detected based on localization constraints on the position of a region in successive camera images. A new cluster is tentatively formed if a signature is miscategorized or if it matches poorly against all existing invariance clusters (i.e., results in a low probability estimate, Figure 8). The new cluster will be removed if it cannot pick up enduring invariance. This typically happens during segmentation errors when a poor match results from two objects being segmented as if they were a single object because of occlusion.

An invariance cluster is used to categorize regions by linking them with internal changes (e.g., the effect of eating a poisonous mushroom). They can do this, for example, by serving as the basis for estimating naive Bayesian probabilities. The robot may need to develop several invariance clusters in order to classify the same kind of object (e.g., edible). Different clusters, for example, may also correspond to the same species of mushroom if it produces visual projections that are very different in the wavelet domain.

Internal feedback gives Ψro the affordance when it makes contact with an object. The robot then creates a *categorical representation* [10] or proto-symbol that links invariance clusters to the internal and external consequences of actions. Ψro uses categorical representations to predict the affordance from the representation that best matches the image signatures. If Ψro miscategorizes, it refines its categorical representations accordingly.

4 Primitive Navigation under Sensorimotor Adaptation

In this section, we introduce a method of primitive navigation that is able to exploit a sensorimotor mapping while it is adapting to changes in a mobile robot's body (e.g., its camera geometry, wheel radius) [21, 13]. The robot's internal state and perceived affordances determine what its current goals are and what objects are unimportant or dangerous and to be avoided. To reach a goal, the robot needs not only to predict the movement of objects but also to recognize its own body. Furthermore, the robot must predict the objects' movement out of the camera's view. This is because objects often go out of view during navigation when using a perspective camera and because

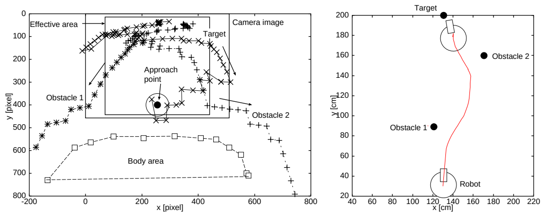

Figure 9: The figure on the left shows the trajectories of two stationary obstacles and a target object (e.g., an edible mushroom when the robot is hungry). The trajectories within the camera image are accurate, but those outside it must be predicted using learned vectors in the sensorimotor mapping that are near the edges of the camera image. Note from the body area that the robot has learned about its physical extension from collisions with objects, shown as squares. The image on the right provides an overhead view, estimated by dead reckoning. (It is given only for illustration since the robot does not use dead reckoning.)

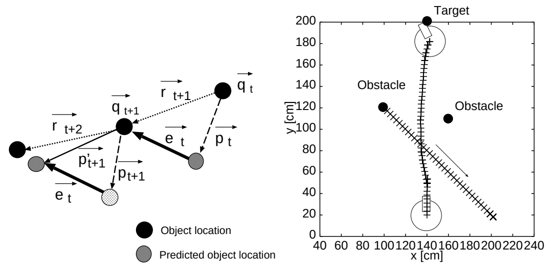

Figure 10: For a given pair of motor signals, Ψro's sensorimotor mapping predicts the movement of stationary objects in its camera image. Moving objects (e.g., a predator) result in prediction error $\vec{e}_t$: the difference between the predicted $\vec{p}_t$ and real $\vec{r}_t$ new location q_{t+1} of the moving object. This error vector is added to the predicted movement vector in the next time step to compensate for the movement and, thus, improve accuracy. The figure on the right shows that Ψro is able to navigate, even when its camera is tilted and another object is moving. The robot stops, backs up slightly to avoid collision, veers left, and moves ahead.

the robot's body is not usually observable. The robot predicts the movement of the unseen objects using the learned sensorimotor mapping based on the information near the edges of the camera image (Figure 9). Simultaneously, the robot learns to recognize its own body in a sensorimotor space that extends beyond the camera's view based on collisions.

Using the learned sensorimotor mapping and body image, the robot chooses an action in the sensorimotor space to circumnavigate obstacles and reach goals. We apply a hydrodynamic potential field in the sensorimotor space to choose an action (cf. [5]). Given each action's predicted movement, the robot calculates the potential value for the next location. Then it takes the action that corresponds to the location with the lowest potential value. If that location is very close to its recognized body, the robot investigates the actions with the next lowest potential values, successively, until it finds one that is expected to bring it into a position that is not too close.

The potential field technique usually has the problem of local minima. To overcome it, the robot avoids revisiting the same location in the sensorimotor space. If the next location is very close to the visited location, the robot investigates an action corresponding to the next lowest potential value. If all the actions are examined, the robot chooses the action that corresponds to the highest potential value (e.g., to back out). Using this algorithm, the robot can react to the immediate state of its sensorimotor space without making a plan that could quickly become out-of-date in a rapidly changing environment. As a result of applying this algorithms to the robot, we confirmed that it could avoid obstacles and reach goals as determined by its current internal state (e.g., hungry, thirsty, see Figure 9). Furthermore, the robot could perform the task after the camera had been tilted, even in the presence of moving obstacles, once it had relearned the mapping (Figure 10). These results demonstrate the generality of the learning method in the mobile robot domain.

5 Conclusion

We examined a computational architecture for responding to affordances. The robot learns a sensorimotor mapping and affordance categorizations or proto-symbols and uses the mapping for primitive navigation to exploit affordances. The robot learns the mapping and categorizations entirely within its sensorimotor space, thus avoiding the issue of how to ground *a priori* internal representations.

Partition nets provide a fast way to learn the sensorimotor mapping. They allocate network weights and partition receptive fields according to the local complexity and sample density of the mapping. They are resilient to kinematic and environmental change and combine neural networks' generalizing ability with the immediate learning of closest point techniques. This makes them interesting as a cognitive model.

References

[1] Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the* ACM, *18*(9), 509-517.

[2] Bhandaru, M. K., Draper, B. A., & Lesser, V. R. (1993). Learning image to symbol conversion. AAAI *Technical Report FS-93-04.*

[3] Bhandaru, M. K., & Draper, B. A. (1998). Discrepancy directed model acquisition for adaptive perceptual systems. In D. F. Rosental & H. G. Okuno (Eds.), *Computational Auditory Scene Analysis.* Mahwah, New Jersey: Lawrence Erlbaum.

[4] Cheney, D. L. & Seyfarth, R. M. (1990). *How monkeys see the world: Inside the mind of another species.* Chicago: University of Chicago Press.

[5] Endo, K., Inaba, M., & Inoue, H. (1992). Viewpoint control based on the potential method,

ISMCR-*1992: Proceedings of the Second International Symposium on Measurement and Control in Robotics.*

[6] Fikes, R. & Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence, 2*(3-4), 189-208.

[7] Fodor, J. A. & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition, 28*(1-2), 3-71.

[8] Fodor, J. A. (1994). J. A. Fodor. In S. Guttenplan (Ed.), *A companion to the philosophy of mind.* Oxford: Blackwell.

[9] Gibson, J. J. (1979). *The ecological approach to visual perception.* Boston, MA: Houghton Mifflin.

[10] Harnad, S. (1987). Category induction and representation. In S. Harnad (Ed.), *Categorical perception: The groundwork of cognition.* Cambridge: Cambridge University Press.

[11] Harnad, S. (1990a). The symbol grounding problem. *Physica D, 42*(1-3), 335-346.

[12] Harnad, S. (1993). Problems, problems: The frame problem as a symptom of the symbol grounding problem. *Psycholoquy, 4*(34). frame-problem.11.

[13] Koeda, M. (2000). *Embodiment in perspective: Sensorimotor adaptation for primitive navigation in a changing body.* Unpublished bachelor's thesis. Faculty of Engineering Science, Osaka University.

[14] MacDorman, K. F. (1992). *First year report and thesis proposal.* Computer Laboratory, Cambridge.

[15] MacDorman, K. F. (1997). *Symbol grounding: Learning categorical and sensorimotor predictions for coordination in autonomous robots.* Technical Report No. 423. Computer Laboratory, Cambridge (e-mail librarian@cl.cam.ac.uk for a copy).

[16] MacDorman, K. F. (1999). Grounding symbols through sensorimotor integration. *Journal of the Robotics Society of Japan, 17*(1), 20-24.

[17] MacDorman, K. F. (1999). Partition Nets: An efficient on-line learning algorithm. ICAR *99: Ninth International Conference on Advanced Robotics*, Tokyo.

[18] Malcolm, C. M. (1995). The SOMASS system: A hybrid symbolic and behaviour-based system to plan and execute assemblies by robot. In J. Hallam, et al. (Eds.), *Hybrid Problems, Hybrid Solutions*, pp. 157-168. Oxford: ISO Press.

[19] Masterman, M. (1961). Semantic message detection for machine translation using an interlingua. In *Proceedings of the International Conference on Machine Translation,* pp. 438-475.

[20] McCarthy, J. & Hayes, P. J. (1969). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer & D. Michie (Eds.), *Machine Intelligence* (vol. 4, pp. 463-502). Edinburgh: University of Edinburgh Press.

[21] Miyazaki, Y. (2000). *Embodied mobile robot navigation using a learned sensorimotor mapping.* Unpublished master's thesis. Graduate School of Engineering Science, Osaka University.

[22] Moore, A. W. (1991). Variable resolution dynamic programming: Efficiently learning action maps in multivariate real-valued state-spaces. In L. Birnbaum & G. Collins (Eds.), *Machine Learning: Proceedings of the Eighth International Workshop.* San Mateo, CA: Morgan Kaufmann.

[23] Moore, A. W. & Atkeson, C. G. (1995). The partigame algorithm for variable resolution reinforcement learning in multidimensional state-spaces. *Machine Learning, 21*(3), 199-233.

[24] Nakawaki, D., Cisek, R., MacDorman, K. F., Joo, S., & Miyazaki, F. (1998). Coaching information determined from dynamic modeling based on a total energy analysis. *16th Annual Conference of the Robotics Society of Japan* (vol. 1), September 18-20, 1998, Hokkaido University, pp. 45-6.

[25] Newell, A. & Simon, H. A. (1972). Human problem solving. Englewood Cliffs, NJ: Prentice-Hall.

[26] Newell, A. (1980). Physical symbol systems. *Cognitive Science, 4*, 135-183.

[27] Omohundro, S. M. (1991). Bumptrees for efficient function, constraint, and classification learning. In R. P. Lippmann, J. E. Moody, and D. S. Touretzky, *Advances in Neural Information Processing Systems 3.* San Mateo, CA: Morgan Kaufmann.

[28] Rummelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representation by error propagation. In D. E. Rummelhart and J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (vol. 1, ch. 8). Cambridge, MA: MIT Press.

[29] Saussure, F. de (1916/1959). Course in General Linguistics (trans.). London: Peter Owen.

[30] Schank, R. C. (1972). Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology, 3*(4), 552-631.

[31] Schyns, P. G., Goldstone, R. L., & Thibaut, J.-P. (1998). The development of features in object concepts. *Behavioral and Brain Sciences, 21*(1), 1-17.

[32] Sidgwick, R. (1998). *Algorithms in C* (3rd ed.). Reading, MA: Addison-Wesley.

[33] Sproull, R. F. (1991). Refinements to nearest-neighbor searching in *K*D trees. *Algorithmica, 6*(4), 579-589.