Learning Posture Invariant Spatial Representations Through Temporal Correlations

Michael W. Spratling

Abstract—**A hierarchical neural network model is used to learn, without supervision, sensory-sensory coordinate transformations like those believed to be encoded in the dorsal pathway of the cerebral cortex. The resulting representations of visual space are invariant to eye orientation, neck orientation, or posture in general. These posture invariant spatial representations are learned using the same mechanisms that have previously been proposed to operate in the cortical ventral pathway to learn object representation that are invariant to translation, scale, orientation, or viewpoint in general. This model thus suggests that the same mechanisms of learning and development operate across multiple cortical hierarchies.**

Index Terms—**Cognitive science, computational models of vision, coordinate transformations, neural networks for development, visual system and development.**

I. Introduction

Visually guided behavior is most naturally defined relative to a number of distinct reference frames or spatial coordinate systems. For example, a saccade takes place within a retina-centered reference frame, reaching for an object takes place in a body-centered frame, while manipulating one object relative to a reference object is performed in a coordinate system centered on the reference object. To support such visually-guided behavior, spatial information is represented in multiple coordinate systems along the dorsal pathway of the cortical visual system [10]: neural representations are arranged in a retinotopic map in primary visual cortex (V1), while regions of the parietal cortex contain representations of space in head-centered [2], [24], body-centered [12], object-centered [14], and world-centered [79], coordinates. A head-centered reference frame could be considered as a representation of retinal position that is *invariant* to eye movements. Similarly, a body-centered representation of visual space is one that is invariant to both eye and neck movements, and an object or world centered reference frame is invariant to eye, neck, and body movements. This insight suggests a mechanism by which a hierarchy of coordinate systems (see Fig. 1) could be learned in the dorsal pathway in a manner analogous to that believed to underlie the learning of object representations invariant to viewpoint in the ventral pathway.

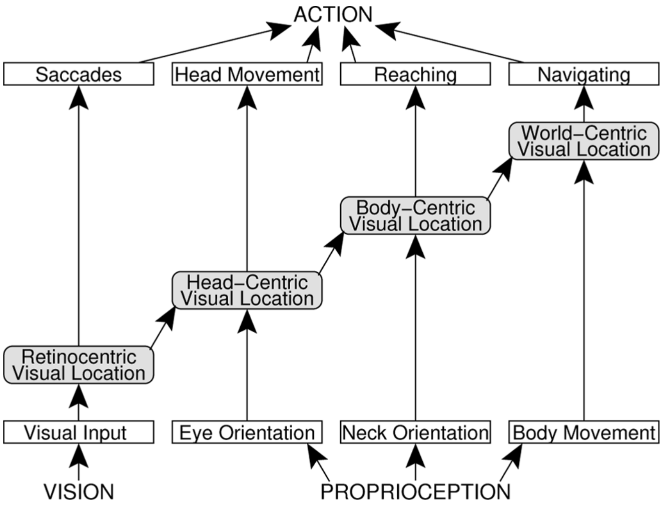

Fig. 1. A hierarchy of coordinate systems. It is proposed that such a hierarchy is learned by the cortical dorsal pathway and that each level in the hierarchy is learned by increasing the invariance to posture, e.g., the head-centric reference frame is learned by forming spatial representations invariant to eye orientation, and the body-centric reference frame is learned by forming spatial representations invariant to neck orientation. Each coordinate system is most appropriate for controlling different types of action, however, this article only considers the learning of sensory-sensory coordinate transformations, and *not* how to map those sensory representations to actions.

Along the cortical ventral pathway, neurons learn to respond to recurring or behaviorally relevant patterns of presynaptic activity generated by the neurons in more peripheral cortical regions from which they receive their inputs [8], [42], [43], [48], [61], [75], [76], [100], [102]. Higher level perceptual representations are thus learned from lower level ones, and this process can be repeated hierarchically, such that at each stage neurons learn increasing *specialization* together with increasing *invariance*. The viewpoint invariance of these representations results from learning associations across time [20], [70], [77], [89], [92], [96], [98], [99]. By learning to associate images whose appearance is closely temporally correlated, cortical neurons exploit the fact that objects are generally observed for periods of time, during which they may undergo a number of transformations or be observed from a number of viewpoints. By learning to associate images of an object, seen from different viewpoints, an invariant representation can be formed. This mechanism has formed the basis for a large number of algorithms that learn viewpoint invariance from sequences of images [9], [11], [27], [30], [44], [60], [69], [80], [90], [91], [93], [97], [101], [107]. Hence, a standard method for learning perceptual representations that are invariant to viewpoint is to form temporal associations across image sequences showing object transformations.

Manuscript received April 21, 2009; revised October 13, 2009. First version published December 15, 2009; current version published February 05, 2010. This work was funded by EPSRC Research Grant EP/D062225/1.

The author is with the Division of Engineering, King's College London, London WC2R 2LS U.K. (e-mail: michael.spratling@kcl.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAMD.2009.2038494

The same mechanism of learning temporal correlations between sequences of images can also be used to learn spatial representations invariant to posture. For example, retinocentric visual information can be combined with information about eye orientation to generate a representation of visual space in head-centric coordinates. Such a recoding can be learned since as the eye moves, stationary objects will move across the retina. By recording temporal correlations across sequences of retinal images, it is possible to learn all the combinations of eye orientation and retinal location that correspond to the same point in space (i.e., to learn a spatial representation that is *invariant* to eye orientation). Using the same process during head movements to learn associations between corresponding head-centered visual locations and neck orientations, results in a representation of visual space in body-centric coordinates.

Rather than learning such sensory-sensory coordinate transformations, they could be hard-coded using standard mathematical equations for describing kinematics or using neural networks with predefined weight values (see Discussion). The motivation for learning sensory-sensory transformations is twofold. First, to contribute to research in developmental robotics which seeks mechanisms for creating more adaptive and more autonomous robots, as well as mechanisms for creating complex robot control systems that are too complex to design by hand [82], [105]. Secondly, to provide a model that may be consistent with the developmental process that occurs during infancy, and hence, which could potentially offer insights into biological development.

II. Methods

Different mathematical processes are required to learn more specialized representations and to learn more invariant representations [67], [80]. A more specialized representation results from a node responding to a combination of coactive lower level features. A node must thus, learn to represent a *conjunction* of presynaptic inputs. To respond to a conjunction of inputs, a standard weighted sum of presynaptic activation values can be used. Such a function will cause the output of the node to be a maximum when all the lower level features to which it responds are simultaneously active. In contrast, a more invariant representation results from a node responding to multiple, noncoactive, lower level features. A node must thus learn to represent a *disjunction* of presynaptic inputs. To respond to a disjunction of inputs, the maximum of the weighted presynaptic activation values can be used. Such a function enables a node to respond invariantly across a number of inputs while maintaining the feature specificity of its response. Hence, several existing architectures for invariant object recognition [32], [33], [45], [46], [67], [73], [74], [80], consist of alternating layers of neurons that perform these two operations in order to form more specialized representations in one layer, and more invariant representations in the next layer. It has been proposed [33], [67], [74], that these two forms of processing correspond to the functionality of simple and complex cells observed in the primary visual cortex [41].

The architecture proposed here for learning a hierarchy of spatial coordinate systems also consists of alternating layers of neurons that learn conjunctions and disjunctions (see Fig. 2). One pair of conjunctive and disjunctive layers learns to transform retinocentric coordinates into head-centered coordinates. A second pair of conjunctive and disjunctive layers then learns to transform these head-centered coordinates into body-centered coordinates. The inputs to the hierarchy are presumed to come from more peripheral thalamic regions that are not explicitly modeled. Details of the algorithms employed in the conjunctive and disjunctive layers are provided in the following subsections. However, in brief, these learning methods are as follows. Each conjunctive layer employs a form of competitive learning that causes a distinct node to learn to represent each distinct input pattern (e.g., for the first conjunctive layer, each distinct combination of eye pan value, eye tilt value, and the location of an active pixel in the retinotopic input). Each disjunctive layer employs a form of temporal associative learning that causes a node to learn to represent sets of nodes in the conjunctive layer that are frequently active in sequence, but not simultaneously. If we consider a world containing a single, stationary object, then as the eyes move distinct combinations of eye pan/tilt and retinal input will be generated activating different conjunctive nodes. This sequence of activity in the first conjunctive layer will be learned by a single node in the first disjunctive layer resulting in a single disjunctive node representing all the conjunctive nodes that represent the location occupied by the object. The performance of this algorithm has been tested, as detailed in the Results section, on a simple task in which objects are represented as single pixels in the retinal input. Future work aims at determining if this entirely unsupervised learning method will scale-up to learning sensory-sensory coordinate transformations in more realistic tasks.

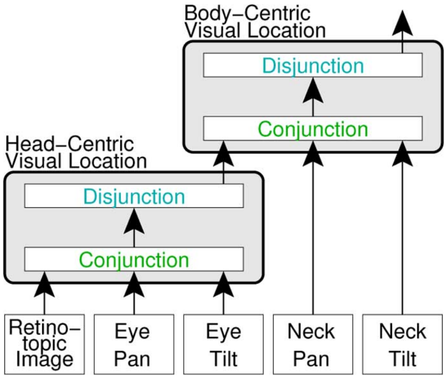

Fig. 2. The neural network architecture used to learn a hierarchy of coordinate systems. Two layers of neurons, one learning conjunctions and the other learning disjunctions, are used to perform each sensory-sensory coordinate transform.

A. Conjunctive Learning

Each conjunctive layer employs the algorithm proposed in [88]. This is an unsupervised, competitive learning algorithm in which nodes compete to represent unique combinations of inputs. This learning algorithm has been shown to reliably and accurately learn distinct input patterns or image components [88]. In this algorithm, nodes compete to respond to the current pattern of input activity in a manner that is closely related to a number of other algorithms. This mechanism of competition can be interpreted as: a sequential version of non-negative matrix factorization [40], [47], as a divisive form of lateral inhibition that targets the inputs to a population of competing nodes [36], [37], [81], [85], [86], as a neural implementation of Bayes' theorem [88], as a form of divisive normalization like that proposed by Heeger [38], [39], and as a hierarchy of such networks can be interpreted in terms of both biased competition and predictive coding [83].

For each conjunctive layer, the output $(\mathbf{y})$ generated in response to an input stimulus $(\mathbf{x})$ was calculated by iteratively updating the following equations

$$\mathbf{e} = \mathbf{x} \oslash \left(\epsilon + \hat{\mathbf{W}}^T \mathbf{y}\right) \tag{1}$$

$$\mathbf{y} \leftarrow (\epsilon + \mathbf{y}) \otimes \mathbf{W}\mathbf{e}. \tag{2}$$

where $\mathbf{y} = [y_1, \ldots, y_n]^T$ is an n-element vector of output activations, $\mathbf{x} = [x_1, \ldots, x_m]^T$ is an m-element vector of input activations, $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_n]^T$ is an n by m matrix of weight values, each row of which contains the weights received by a single node, $\hat{\mathbf{W}} = [\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_n]^T$ is a matrix representing the same synaptic weight values as $\mathbf{W}$, but such that the rows of $\hat{\mathbf{W}}$ are normalized to have a maximum value of one, $\mathbf{e}$ is the inhibited value of the input (or equivalently, the reconstruction error; see below), and $\oslash$ and $\otimes$ indicate element-wise division and multiplication, respectively. The parameter ϵ is a small constant (i.e., 1×10^{-9}) that has a negligible effect on the calculation of $\mathbf{e}$ and $\mathbf{y}$ except to prevent division-by-zero errors when the values of $\mathbf{y}$ are zero. The steady-state values of the node outputs were calculated using 100 iterations of the above equations while the input was held constant.

The following learning rule was applied to the steady-state node activations

$$\mathbf{W} \leftarrow \mathbf{W} \otimes \left\{1 + \beta \mathbf{y}\left(\mathbf{e}^T - 1\right)\right\}. \tag{3}$$

where β is a positive constant which controls the learning rate. A value of $\beta = 0.025$ was used in the experiments described in this article. Following learning, weights were clipped at zero to ensure that they were non-negative. Weights were initialized to random values chosen from a Gaussian distribution with mean 0.5 and standard deviation 0.125.

This learning algorithm for the conjunctive layers operates by minimizing the error between the input stimulus $(\mathbf{x})$ and the input that is reconstructed from the node outputs $(\hat{\mathbf{W}}^T\mathbf{y})$. The values of $\mathbf{e}$ indicate the degree of mismatch between the top-down reconstruction of the input and the actual input. When a value within $\mathbf{e}$ is greater than unity, indicating that a particular element of the input is under-represented in the reconstruction, the responses of all output nodes receiving nonzero weights from this under-represented input are increased [via (2)] and the values of weights connecting the under-represented input with active output nodes are also increased [via (3)]. Both these changes will lead to an increase in the strength with which that element is represented in the reconstructed input, and hence, reduce the value of that element of $\mathbf{e}$ towards one [via (1)]. Similarly, when a value within $\mathbf{e}$ is less than unity, indicating that a particular element of the input is over-represented in the reconstruction, the responses of all output nodes receiving nonzero weights from this over-represented input are reduced [via (2)] and the values of weights connecting the over-represented input with active output nodes are also reduced [via (3)]. Both these changes will lead to a decrease in the strength with which that element is represented in the reconstructed input, and hence, increase the value of that element of $\mathbf{e}$ towards one [via (1)]. When the value of $\mathbf{e}$ is equal to unity the reconstruction of that element is perfect and the weights stop changing due to the term $\left(\mathbf{e}^T - 1\right)$ in (3). For elements that are not active in the input vector, the corresponding elements of $\mathbf{e}$ will be zero and the corresponding weights (for active nodes) will stop changing once they have reached a value of zero. Hence, a weight stops changing value when the top-down reconstruction is perfect (i.e., when $\hat{\mathbf{W}}^T\mathbf{y} = \mathbf{x}$) or when the weight is zero.

This algorithm thus finds non-negative, elementary components of the training data, and uses these components to encode each stimulus with the minimal loss of information (i.e., with the minimal reconstruction error). Unlike many other competitive learning algorithms see [86], [87], and the references therein, the proposed algorithm does not prespecify the number of nodes that are required to encode each stimulus. Instead, the number of active nodes is determined by the number of elementary components that are required to accurately represent the input. Hence, each conjunctive layer can represent multiple, co-occurring objects.

B. *Disjunctive Learning*

Each disjunctive layer implements an improved version of the algorithm proposed in [80]. This algorithm is similar to previous methods that exploit temporal correlations to learn invariant representations. However, it differs from these previous methods in: (a) learning correlations between consecutive image pairs, rather than across longer sequences of images; (b) biasing learning so that coincident inputs will be represented by distinct nodes. These modifications enable learning to succeed when the training environment contains multiple, co-occurring stimuli [80]. Multiple stimuli might occur when the visual input contains multiple objects or background clutter.

For each disjunctive layer, an n-element vector of output values $(\mathbf{y})$ generated in response to an m-element vector of inputs $(\mathbf{x})$ was calculated using the following equation

$$\mathbf{y} = \max\left\{\hat{\mathbf{W}} \otimes \check{\mathbf{W}} \otimes \mathbf{X}\right\} \tag{4}$$

where $\hat{\mathbf{W}} = [\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_n]^T$ is an n by m matrix representing the synaptic weight values such that the rows of $\hat{\mathbf{W}}$ are normalized to have a maximum value of one, $\check{\mathbf{W}} = [\check{\mathbf{w}}_1, \ldots, \check{\mathbf{w}}_n]^T$ is an n by m matrix representing the same synaptic weight values such that the columns of $\check{\mathbf{W}}$ are normalized to have a maximum value of one, $\mathbf{X} = [\mathbf{x}^T, \ldots, \mathbf{x}^T]^T$ is a n by m matrix each row of which is a copy of the input vector $\mathbf{x}$, and $\max$ is a function which returns the maximum value in each row.

The activation function, described above, is identical to that used in [80]. The learning rule, described below, is different from that used previously, although it operates using the same principles. The following learning rule was applied

$$\mathbf{W} \leftarrow \mathbf{W} \otimes \{1 + \gamma \mathbf{y} (\mathbf{x}^* - \mathbf{x})\} \tag{5}$$

where $\mathbf{x}^*$ is the vector of input values for the *previous* input pattern, and γ is a positive constant which controls the learning rate. A value of $\gamma = 0.25$ was used in the experiments described in this article. Importantly, for each node, the single synapse that produced the maximum value in (4) did not have its weights modified. Following learning, synaptic weights were clipped at zero (i.e., negative weight values were made equal to zero) and were normalized such that the sum of the weights in each column of $\mathbf{W}$ was equal to one. Weights were initialized to random values chosen from a Gaussian distribution with mean $0.5/n$ and standard deviation $0.125/n$.

This learning algorithm for the disjunctive layers operates by: (a) finding correlations between successive input patterns; (b) ensuring that coactive inputs (which must be generated by the coactivation of distinct retinal locations) are not represented by the same node. Objective (a) is achieved by nodes which are active in response to the current input pattern increasing their synaptic weights to inputs which were active in the previous stimulus (in proportion to $\mathbf{y}\mathbf{x}^*$). A strongly active node has a strong weight to an input that is active in the current stimulus. Hence, by increasing this node's synaptic weights to inputs that were active in the previous stimulus the algorithm is learning temporal correlations between successive stimuli, in a manner consistent with Hebbian learning. Objective (b) is achieved by nodes which are active in response to the current input pattern decreasing their synaptic weights to inputs which are active in the current stimulus in an anti-Hebbian manner (i.e., in proportion to $-\mathbf{y}\mathbf{x}$). Note, only those synapses that did not generate the maximum input to the node are modified, so a single active input remains unchanged, but weights to all other active inputs are deceased. Furthermore, the weights are normalized so that the sum of the weights emanating from a single input are equal to one. This normalization process provides an implicit form of competition between nodes, since if one node strengthens its connection to a particular input, then connections from that input to all other nodes are weakened.

III. Results

A two stage hierarchy (as illustrated in Fig. 2) was used to learn to encode sensory input in both head-centric and body-centric coordinates. The inputs to this hierarchy were a two-dimensional image, and four one-dimensional arrays encoding the four motor degrees of freedom: eye pan/tilt and neck pan/tilt. The axis of the motor actions were coincident with the x and y axis of the retinal array, hence, both the eye and neck pan/tilt values served to slide the retina around within a larger world coordinate system, as illustrated in Fig. 3. This visual world consisted of a grid of pixels large enough to accommodate the retina at the extremes of the pan/tilt ranges. The world was populated by "objects," represented by pixels with contrast values randomly selected within the range 0 to 1. Since real objects have an extent, representing objects as pixels is a limitation of the current algorithm, but one that it shares with other methods for performing sensory-sensory transformation (see Discussion).

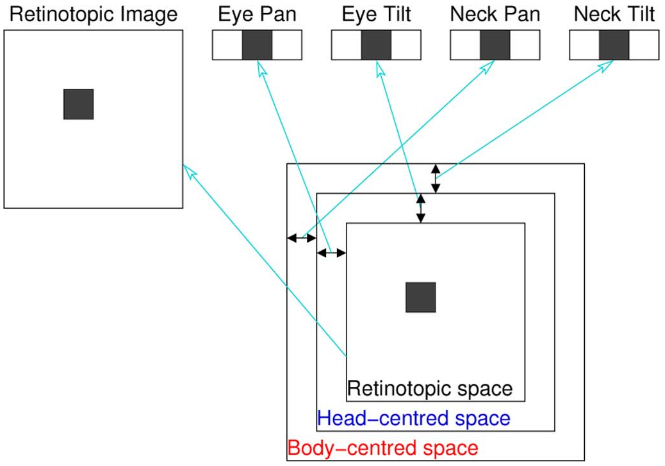

Fig. 3. Training data, for presentation to the inputs of the neural network architecture shown in Fig. 2, was generated by populating a body centric world with objects represented by pixels with nonzero values (one object is illustrated near the center of the retinocentric space). The retina was slid around within a world by both eye and neck movements. The portion of the world visible on the retina at the extremes of eye pan and tilt (for a fixed neck pan and tilt) defines a head-centric world. The world visible on the retina at the extremes of both eye and neck pan and tilt defines a body-centric world. Values of pan and tilt were represented to the network as vectors. Each pan/tilt vector had one element equal to a value of one, representing the pan/tilt value, and all other elements were zero. The retinotopic image presented to the network was simply the array of world pixels visible to the retina given the current pan and tilt values. The strength of the pixel contrasts in the retina were normalized so that they summed to one.

Experiments were carried out using a small 3-by-3 pixel retina, and pan and tilt values that could each take one of three values. The world was thus 7-by-7 pixels. For these experiments the first conjunctive layer contained 180 nodes, the first disjunctive layer contained 50 nodes, the second conjunctive layer contained 550 nodes, and the second disjunctive layer contained 100 nodes. Hence, each layer contained approximately twice as many nodes than the minimum number required for the task. While this is a small scale problem, it is still more complex than the task used to test many existing methods (see Discussion) and the network is still quite complex containing 97 500 synaptic weights.

The initial world was randomly created with each pixel independently selected to contain an object with probability P_s (P_s sets the sparsity of the visual input). Values of P_s ranging from 0.05 to 0.3 were used in the different experiments reported below. Each selected pixel then had its contrast set to a value chosen uniformly from the range $[0, 1]$. In order to generate a sequence of data in which objects tended to remain stationary for an extended period of time, each subsequent visual world was created by removing, with probability P_{off}, existing objects (by setting the corresponding pixel value to zero). Similarly, new objects could appear at empty pixel locations with probability P_{on}. New objects were assigned a random contrast in the range $[0, 1]$. In order to keep the sparsity of the world constant (at P_s), P_{on} was related to P_{off} by $P_{on} = P_{\text{off}} P_s/(1 - P_s)$. Values of

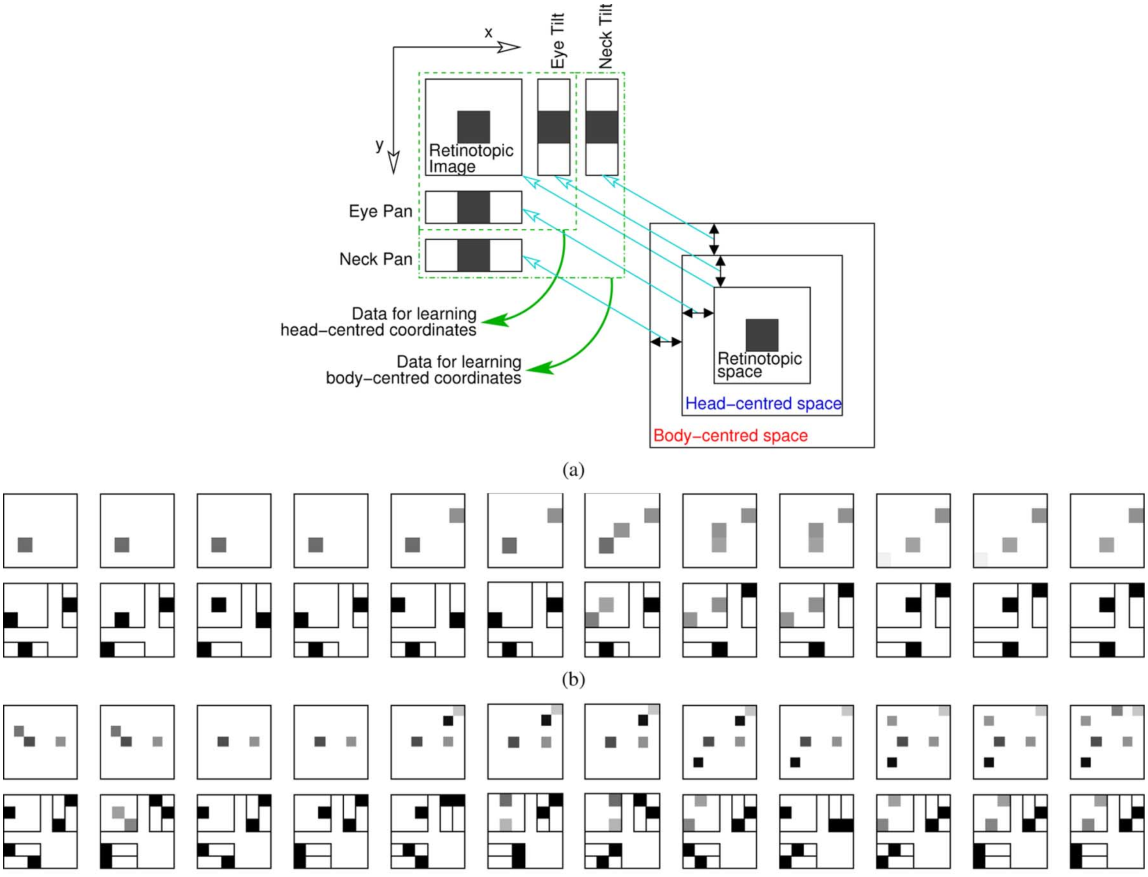

Fig. 4. Example training data. (a) shows Fig. 3 rearranged to illustrate the format in which the training data is presented in (b) for learning head-centered coordinates, and (c) for learning body-centered coordinates. In both (b) and (c) the top rows show a sequence of 12 images of a visual world created using $P_s = 0.1$ and $P_{\text{off}} = 0.1$. The bottom rows show the corresponding 12 sets of input data presented to the neural network. For learning head-centric coordinates it is assumed that the neck pan and tilt values are constant. Hence, the visual world, shown on the top row of (b), is the head-centered space (fixed relative to the body-centered space) as shown on the right of (a). Furthermore, since the neck pan/tilt values are not used as inputs to the stage of the neural network hierarchy learning the head-centric coordinates (see Fig. 2), these values are ignored. Hence, the sequence of training data shown on the bottom row of (b) consists of the retinotopic image and the eye pan and tilt vectors shown in the configuration illustrated within the dashed box on the top-left of (a). For learning body-centric coordinates, the visual world, shown on the top row of (c), is the body-centered space shown on the right of (a). The sequence of training data shown on the bottom row of (c) consists of the retinotopic image, and both the eye and neck pan and the eye and neck tilt vectors, shown in the configuration illustrated within the dash-dot box on the top-left of (a). In both sets of training data, the retina is smaller than the visual world, and hence, the retinal inputs [top-left square in the bottom row of each subfigure in (b) and (c)] contains only a subset of the pixels shown in the corresponding visual world (top row of the same column). The retina is positioned on the visual world at the location determined by the pan and tilt values, and hence, the subset of pixels contained in the retinotopic image is determined by the pan and tilt vectors. For example, for the left most column in (b) both the pan and tilt values are in the middle of their ranges ($pan = 2, tilt = 2$) and so the top and left edges of the retina are positioned one pixel from the top and left edges of the world (and the bottom and right edges of the retina are positioned one pixel from the bottom and right edges of the world). Hence, the only object in this world appears at the bottom left of the retina [at coordinates (1,3)]. In the next image in the sequence, the eye pan value has moved one position to the left ($pan = 1, tilt = 2$) so that the left edge of the retina is now positioned at the left edge of the world and the object now appears in the retina at the bottom center pixel (at coordinates (2,3)). If for brevity we denote each input pattern using a vector of four values $(pan, tilt, x, y)$, then the first five sets of input data for learning head-centered coordinates shown on the bottom row of (b) can be written as (2,2,1,3), (1,2,2,3), (1,3,2,2), (2,2,1,3), and (2,3,1,2). Note that the first and fourth patterns are the same, so we have four distinct training patterns. Each of these patterns represents the same location in the visual world [the location of the pixel at coordinates (2,4) in the visual world shown on the top row of (b)]. These four patterns form a "disjunctive set" representing that location.

P_{off} ranging from 0.05 to 0.3 were used in the different experiments reported below. Worlds that contained no objects were removed from the training sequence.

The retinal input corresponding to each world array was generated by randomly selecting pan and tilt values with the restriction that the highest contrast pixel in the world remained visible on the retina. Each retinal input was then normalized by the total sum of the pixel values in the retina. To generate the data for training the head-centric representation, the neck pan/tilt values were kept constant while the eye pan/tilt values were chosen randomly. To generate the data for training the body-centric representation, both the eye pan/tilt and neck pan/tilt values were chosen randomly. Examples of sequences of training data generate by this method are shown in Fig. 4. The four layers of neurons making up the model were trained one after the other. Specifically, a sequence of 100 000 input patterns were used to train the first conjunctive layer. These weights were then held constant while another 100 000 input patterns were presented and the weights into the first disjunctive layer were learned. This procedure was then repeated to train the second conjunctive and disjunctive layers using 10 0000 input patterns each.

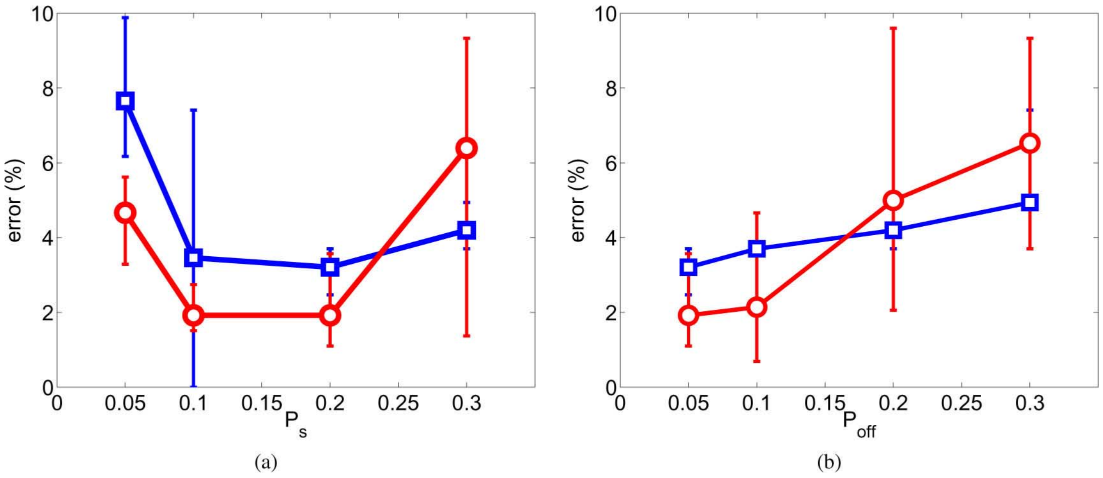

Fig. 5. Accuracy of head-centric representation (square markers) and body-centric representation (circular markers), for (a) changing the sparsity of objects in the world (while $P_{\rm off}$ was fixed at 0.05), and (b) changing the average time an object stayed static in the world (while P_s was fixed at 0.2). Each plot shows the mean percentage error with error bars indicating the best and worst performance over five trials using different randomly generated sequences of input data and different randomly initialized synaptic weights.

To test the accuracy of the sensory coordinate mappings learned by the network, the following procedure was used. A single object was placed in the visual world at each possible location in turn. For each world location, a set of input patterns were created containing every possible combination of the retinal location and the pan/tilt values corresponding to this world location. This set of input patterns corresponding to a single world location constitutes a "disjunctive set" of patterns, and the entire test set consisted of disjunctive sets for every possible world location. For each individual input pattern the node in the disjunctive layer under test that generated the strongest response was identified. This node can be said to represent that input pattern. Ideally, all the patterns in a disjunctive set (i.e., all the combinations of pan/tilt and retinal coordinates corresponding to a single world location) should be represented by a single node, and patterns in different disjunctive sets (i.e., patterns corresponding to different world locations) should be represented by distinct nodes. Hence, a pattern was considered misrepresented if it was not represented by the node representing the majority of patterns in that disjunctive set. Furthermore, if a single node represented patterns corresponding to more than one distinct world location, those patterns not forming part of the largest disjunctive set represented by that node were also considered misrepresented. A percentage error was then calculated as the ratio of misrepresented patterns to all patterns in the test set (i.e., all possible combinations of retinal location and pan and tilt values for all possible world locations). Note that error was *not* calculated in terms of the distance between the actual location of the object and the location represented on the map. Instead, the error values measured the percentage of patterns mismapped by the network. In other words, for the percentage of patterns not mismapped (100-error) the mapping produced was 100% accurate.

Experiments were carried out to assess the robustness of the learning algorithm to changes in the sparsity of the world (P_s), and the probability that successive images contained the same object $(P_{\rm off})$. For each condition tested, the neural architecture was trained five times using different randomly generated sequences of input data and different randomly initialized synaptic weights. Fig. 5 shows the mean percentage error (and the maximum and minimum error) over these five trials for each combination of P_s and $P_{\rm off}$ tested. It can be seen from Fig. 5(a) that performance was best for P_s values between 0.1 and 0.2. As the world becomes sparser than 0.1, the retina will rarely contain multiple objects. Since, the disjunctive learning rule exploits the presence of multiple objects to divide its inputs into separate disjunctive sets, it is not surprising that performance gets worse as P_s becomes very small. As the world becomes denser than 0.2, the retina will typically contain several objects. It is then difficult to learn which of several objects in one image corresponds to which of several objects in the next image. Hence, it is to be expected that learning posture invariance from temporal correlations becomes less accurate as the average number of objects in each input pattern increases. However, it is easy to imagine how the early stages of visual processing could control the sparseness of the retinocentric representation of space, to enable successful learning of subsequent sensory-sensory transformations. Fig. 5(b) shows that the performance of the learning algorithm improves as $P_{\rm off}$ decreases. This is expected since the smaller $P_{\rm off}$, the longer the sequence of training data containing the same object, and hence, the more opportunity the algorithm has to learn all combinations of retinal position and pan/tilt value that correspond to the same spatial location.

It can be seen from both Fig. 5(a) and (b) that the body-centric representation is often more accurate than the head-centric representation. Since the former is built on the latter, this result seem inconsistent. However, it is due to the rather strict criteria used to quantify the accuracy of the representations learned. The assessment method requires that each disjunctive set is represented by a distinct node, and that a single node produces the strongest response to all members of a disjunctive set. The second criteria of success is responsible for many of the recorded errors, but such "errors" in the head-centric representation do not prevent the body-centric representation being learned successfully. For a small proportion of disjunctive sets two or more nodes in the disjunctive layer learn strong weights to all the members of this disjunctive set. Unless one of these nodes produces the strongest response to all members of the disjunctive set, we classify some of the patterns making up the disjunctive set as being misrepresented. However, if multiple nodes all respond to a disjunctive set, the conjunctive layer in the next stage of the learning hierarchy can learn strong connections to all these nodes and still form an accurate representation of all possible conjunctions at the next level. Hence, some of the errors being counted by the assessment method, when applied to the head-centric representation, do not actually have an effect on subsequent learning of the body-centric representation. Such "errors" seem to be most common at the corners of the space being encoded. Of all the combinations of retinal location and pan/tilt values a smaller proportion are near corners in the body-centric space than in the head-centric space, hence, the accuracy of the body-centric representation is better.

IV. Discussion

This article has shown that it is possible to learn sensory-sensory coordinate transformations using a completely unsupervised learning algorithm. Specifically, it has been shown that a sequence of retinal images produced using random eye movements and a slowly changing world can be used to learn a representation of visual space that is invariant to eye movements, i.e., a head-centric reference frame. Similarly, this representation of the visual world combined with random head movements can be used to learn a representation of visual space that is invariant to eye and neck movements, i.e., a body-centric reference frame. Other recent work has shown that similar methods which exploit temporal correlations can be used to learn spatial representations invariant to posture similar to those encoded by "place cells" in the rat hippocampus [13], [31]. The current results complement this work by showing that similar mechanisms can account for learning spatial reference frames in parietal cortex.

The reported results demonstrate that the proposed algorithm works on a simple test problem. Future work will be required to determine if this algorithm can scale-up to more realistic tasks. Existing methods for generating sensory-sensory mappings use hard-coded solutions or supervised learning, and hence, are likely to outperform the method proposed here. The advantages of the current method are not in terms of a quantitative improvement in performance, but rather in demonstrating that a completely unsupervised method can learn sensory-sensory coordinate transformations. Hence, in the following discussion the comparison with other methods is concerned primarily in discussing the shortcomings of these existing methods in terms of their plausibility as models of development in infants and/or as algorithms that can advance work in epigenetic robotics.

The proposed algorithm has been tested with a simple task using a two-dimensional retinal image and four degrees of freedom for motor action: eye pan/tilt and neck pan/tilt. While this task is simple, it is more challenging and realistic than the task that has typically been used to assess previous models of cortical spatial transformations. This previous task is illustrated in Fig. 6(a). In this task, the retinotopic map is one-dimensional and there is one degree of freedom for motor action which causes the retinal input to translate along its length [63], [103]. This is equivalent to calculating the sum of two scalar values represented by peaks of activity in two one dimensional input maps [95]. For ease of comparison with previous methods, the architecture of the proposed algorithm when applied to this simple one-dimensional problem is shown in Fig. 6(b).

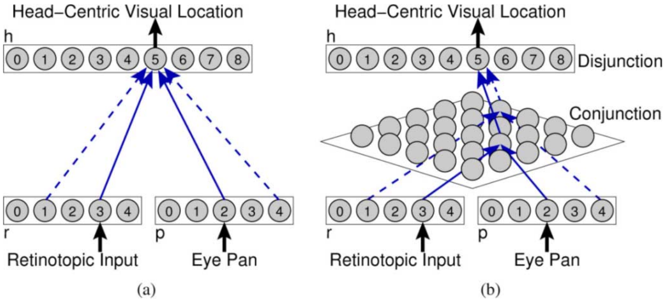

Fig. 6. (a) A simple task used to test previous models of sensory coordinate transformations. Two scalar values r_i and p_j, encoded by the activation pattern of two populations of neurons, are combined to calculate a third scalar value h_k, encoded by the firing of a third population of neurons. This general task can be interpreted in terms of simulating a mapping from retinocentric coordinates to head-centered coordinates, if r_i is interpreted as representing the position of a target on a one-dimensional retina, and p_i is interpreted as a representation of eye position. In the simple case where the eye position generates a horizontal shift of the retina along its axis, the correct value of h is simply the sum of the values r and p. Hence, two pairs of retinocentric coordinates and eye position values that should produce the same head-centric output are indicated by solid and dashed lines respectively. (b) The architecture of the proposed model when applied to this simple task. The head-centered representation is generated by a population of disjunctive nodes. These nodes learn strong weights from a set of conjunctive nodes (all representing the same spatial location). The conjunctive nodes learn each possible combination of retinotopic and eye position inputs. The conjunctive layer thus learns a set of basis functions, and the disjunctive layer learns the mapping from these basis functions to a coordinate system in which object position is invariant to posture.

Basis function networks [Fig. 7(a)] are an influential model of cortical spatial transformations [22], [63]–[65]. In such models, a layer of basis function nodes encode every possible combination of sensory input signals. This layer is functionally equivalent to the conjunctive layer in the proposed model [compare Fig. 7(a) with Fig. 6(b)]. The outputs of the basis functions can be combined linearly, to produce a spatial map in a new reference frame (which is equivalent to the operation performed by the disjunctive layer in the proposed model). The need to assign one node to represent each combination of sensory signals results in the basis function network size increasing exponentially with problem size [22], [63]. To resolve this issue it is possible to decompose sensory-sensory transformations into several steps. Such a solution results in a model containing a hierarchy of coordinate systems [65] similar to the one proposed in this article. However, basis function networks differ in two key respects from the proposed model. First, because the response of a neuron in a basis function network is inversely proportional to the Euclidean distance between the input and the weights of that node, basis function networks require that all the sensory information originates from a single object [65], and hence, unlike the proposed model cannot represent the spatial locations of multiple objects that are present simultaneously. Secondly, the synaptic weights of each basis function need to be predefined, hence, unlike the

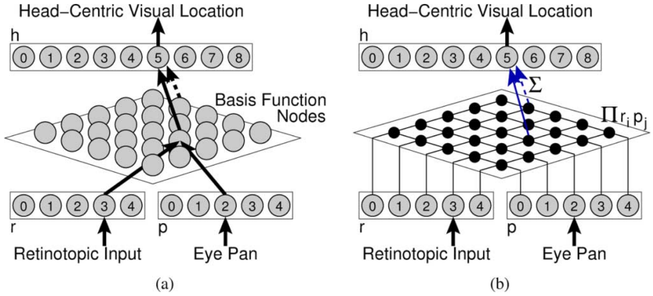

Fig. 7. Architectures of alternative methods of performing sensory reference frame transformations for the one-dimensional task described in Fig. 6(a). (a) A basis function network as used in [22], [63]–[65], (b) A sigma-pi network as used in [103] and [104].

proposed model spatial representations are not learned in an unsupervised way, and such a model cannot be used to explore the development of spatial representations in cortex.

Weber and Wermter [103], and Weber *et al.* [104] proposed a method for learning sensory reference frame transformations. The nodes in their network were sigma-pi units (Fig. 7(b)). The response of each node was determined by calculating the products of pairs of signals, and then calculated the sum over these products. The two parts of this activation function correspond to the two layers in the proposed model and a basis function network [compare Fig. 7(b) to Figs. 6(b) and 7(a)]. As with the proposed model, the weights connecting products to outputs are learned using temporal correlations between successive input patterns. However, unlike the proposed model, the combinations of inputs that form the products are predefined. This is equivalent to predefining the weights in a basis function network, or predefining the weights in the conjunctive layers of the proposed model. Hence, only part of the mapping is learned in this algorithm, while half the problem is solved by hard wiring the solution rather than learning it.

Hence, in contrast to the algorithm proposed here (and to similar models of hippocampal spatial representations [13], [31]), previous models have failed to learn the sensory coordinate transformation, but have had the solution hard wired into the weights of the network to some extent. Furthermore, none have demonstrated that the resulting spatial representation can be used as the input to a subsequent spatial reference frame mapping (i.e., that the model can form part of a larger hierarchy of spatial transformations).

The proposed model, and those discussed above, suggest that when performing a task such as reaching for an object, the retinocentric coordinates of the target object are transformed into body-centered coordinates via a series of intermediate reference frames. Hence, sensory information is recoded into a coordinate system which is more likely to be consistent with the coordinate system of the motor effectors. The implicit assumption is that the new coordinate systems is more appropriate for learning and controlling arm movements. However, an alternative strategy, embodied in many existing models, is to directly learn the mapping between sensory inputs and motor outputs without any recoding via intermediate reference frames e.g., [1], [5], [6], [15], [16], [18], [19], [34], [53]–[55], [68], [71], and [72]. Unlike many previous algorithms for sensory-sensory mappings, these sensory-motor mappings are usually learned rather than hard-wired. This is made possible by associating the sensory consequences of random motor movements (motor "babbling") with the outputs that generated those motor movements. In other words, the motor outputs provide a supervisory signal for learning the sensory-motor mapping that is not available for learning a sensory-sensory mapping.

While the ability to exploit the correlation between motor outputs and sensory inputs to learn the sensory-motor mapping gives these methods an advantage over hard-coded methods of learning sensory-sensory mappings, the existing algorithms for learning direct sensory-motor mappings typically suffer from other issues.

- These existing algorithms typically require that the sensory input contain only one item which always corresponds exactly to the position of the end-effector that the algorithm is learning to control, which is a serious limitation.
- During training, the output of the motor region is determined by a random number generator, whereas it is controlled by the sensory input once learning is complete. Such a distinction between the training phase and operational phase is biologically implausible as it requires neurons to switch behaviors [84].
- There are a large number of possible combinations of sensory inputs for which a mapping needs to be learned [94], this is equivalent to the problem faced by basis function networks where the size of the network needs to increase exponentially with the size of the task.
- The required motor configuration may depend on nonlinear relationships between the different sensory inputs. There may be little similarity in the required output for similar input patterns, hence, little opportunity for generalization.

At least some of these difficulties can be solved by recoding the sensory data into a more abstract representation such as a body-centric reference frame. Recoding via intermediate coordinate systems can overcome the exponential increase in problem size, the more direct correspondence between the recoded sensor and motor coordinate systems provides generalization between similar situations, and the abstract sensory representation can be reused for learning other sensory-motor mappings.

As well as providing a mechanism for learning sensory-sensory coordinate transforms (and subsequently, sensory-motor control), the proposed hierarchical neural network algorithm may have application to modeling infant development or implementing developmental processes in robots. Development is a constructivist process through which progressive improvements in ability are achieved by using simpler skills as a basis for learning more complex ones, and so on hierarchically [4], [29], [50], [66], [78]. A neural network model of such a process thus requires nodes to be available to exploit the learning that has been achieved by other nodes. One obvious method by which this can be achieved is to allow the outputs of certain nodes to provide (some of) the inputs to other nodes, in a hierarchical arrangement. This second set of nodes is then in a position to make use of, and build upon, the results of learning in first set of nodes. In such a hierarchical neural architecture, more complex representations can be learned in higher level networks based on simpler representations learned in lower level networks. In addition, the simpler representations constrain the search space for learning in subsequent layers, and hence, make tractable the task of learning more complex representations [17], [29]. The organization of cortical regions into functional hierarchies, such as those found in the dorsal and ventral streams, suggest that hierarchical neural architectures, like that proposed here, can provide the basis for a model of cortical development that is more biologically plausible than many other neural network algorithms, like those reviewed in [106], that have been proposed as models of development. Neural hierarchies capable of learning complex perceptual representations which are appropriate for controlling complex actions are likely to be essential to the developmental process in humans, and algorithms capable of learning a hierarchy of representations are therefore likely to be essential if the field of developmental robotics [82], [105], is to make further progress, which in turn, is imperative for the creation of more intelligent and adaptive machines.

In the proposed model, initial learning of a head-centric representation simplifies the subsequent task of learning a body-centric representation. In turn, the learning of the head-centric spatial representation requires visual inputs generated by eye movements in the absence of neck movements. Hence, the model proposes that the lack of head control in young infants is an advantage for learning spatial representations, in the same way that it has previously been proposed that limitations in motor control or cognitive ability may provide an advantage to learning complex tasks [3], [28], [29], [49], [52]. Another prediction of the proposed model is that learning invariant representations will be aided by the presence of multiple stimuli. This is in contrast to other algorithms for learning invariance which require stimuli to be presented in isolation e.g., [30], [59], [91], [97], and [101].

V. Conclusions

The structural uniformity of the neocortex [21], [25], [56] has led many theorists to suggest that the cortex is also computationally uniform [7], [23], [25], [26], [35], [51], [57], [58], [62]. This article adds to these arguments by demonstrating that the same mechanisms which are widely believed to underlie the learning of object representations with invariance to viewpoint in the ventral pathway can also give rise to spatial representations invariant to eye and neck movements that are believed to exist in the dorsal pathway. A major advantage of the proposed algorithm over previous models of cortical sensory-sensory transformations is that these coordinate transformations are learned using an entirely unsupervised learning algorithm.

References

[1] J. S. Albus, *Brains, Behaviour and Robotics*. Peterborough, NH: BYTE Books, 1981.

[2] R. A. Andersen, G. K. Essick, and R. M. Siegel, "Encoding of spatial location by posterior parietal neurons," *Science*, vol. 230, no. 4724, pp. 456–458, 1985.

[3] E. G. Araujo and R. A. Grupen, P. Maes, M. Mataric, J. Meyer, J. Pollack, and S. W. Wilson, Eds., "Learning control composition in a complex environment," in *Proc. Int. Conf. Simulation Adapt. Behav.*, Cambridge, MA, 1996, pp. 333–342.

[4] M. A. Arbib, E. J. Conklin, and J. C. Hill, *From Schema Theory to Language*. Oxford, U.K.: Oxford Univ. Press, 1987.

[5] C. Balkenius, "Multi-modal sensing for robot control," in *Current trends in connectionism*, L. F. Niklasson and M. B. Bodén, Eds. Hillsdale, NJ: Erlbaum, 1995, pp. 203–216.

[6] P. Baraduc, E. Guigon, and Y. Burnod, "Recodning arm position to learn visuomotor transformations," *Cereb. Cortex*, vol. 11, pp. 906–917, 2001.

[7] H. B. Barlow, "What is the computational goal of the neocortex?," in *Large-Scale Neuronal Theories of the Brain*, C. Koch and J. L. Davis, Eds. Cambridge, MA: MIT Press, 1994, ch. 1.

[8] H. B. Barlow, "The neuron doctrine in perception," in *The Cognitive Neurosciences*, M. S. Gazzaniga, Ed. Cambridge, MA: MIT Press, 1995, ch. 26.

[9] M. S. Bartlett and T. J. Sejnowski, "Learning viewpoint invariant face representations from visual experience in an attractor network," *Netw.: Comput. Neural Syst.*, vol. 9, no. 3, pp. 1–19, 1998.

[10] A. Battaglia-Mayer, R. Caminiti, F. Lacquaniti, and M. Zago, "Multiple levels of representation of reaching in the parieto-frontal network," *Cereb. Cortex*, vol. 13, no. 10, pp. 1009–1022, 2003.

[11] S. Becker, "Implicit learning in 3D object recognition: The importance of temporal context," *Neural Comput.*, vol. 11, no. 2, pp. 347–374, 1999.

[12] P. R. Brotchie, R. A. Andersen, L. H. Snyder, and S. J. Goodman, "Head position signals used by parietal neurons to encode locations of visual stimuli," *Nature*, vol. 375, no. 6528, pp. 232–235, 1995.

[13] P. Byrne and S. Becker, "A principle for learning egocentric-allocentric transformations," *Neural Comput.*, vol. 20, no. 3, pp. 709–737, 2008.

[14] M. V. Chafee, B. B. Averbeck, and D. A. Crowe, "Representing spatial relationships in posterior parietal cortex: Single neurons code object-referenced position," *Cereb. Cortex*, vol. 17, no. 12, pp. 2914–2932, 2007.

[15] P. M. Churchland, "Some reductive strategies in cognitive neurobiology," in *The Philosophy of Artificial Intelligence*, M. A. Boden, Ed. Oxford, U.K.: Oxford Univ. Press, 1990, ch. 14.

[16] P. S. Churchland, *Neurophilosophy*. Cambridge, MA: MIT Press, 1986, ch. 10.

[17] A. Clark and C. Thornton, "Trading spaces: Computation, representation and the limits of uninformed learning," *Behav. Brain Sci.*, vol. 20, no. 1, pp. 57–66, 1997.

[18] P. R. Cohen, M. S. Atkin, T. Oates, and C. R. Beal, "Neo: Learning conceptual knowledge by sensorimotor interaction with an environment," in *Proc. 1st Int. Conf. Autonom. Agents*, Marina Del Ray, CA, 1997, pp. 170–177.

[19] Y. Coiton, J. C. Gilhodes, J. L. Velay, and J. P. Roll, "A neural network model for the intersensory coordination involved in goal-directed movements," *Biolog. Cybern.*, vol. 66, no. 2, pp. 167–176, 1991.

[20] D. D. Cox, P. Meier, N. Oertelt, and J. J. DiCarlo, "'Breaking' position-invariant object recognition," *Nature Neurosci.*, vol. 8, no. 9, pp. 1145–1147, 2005.

[21] F. Crick and C. Asanuma, "Certain aspects of the anatomy and physiology of the cerebral cortex," in *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Volume 2: Psychological and Biological Models*, D. E. Rumelhart and J. L. McClell, Eds. Cambridge, MA: MIT Press, 1986, pp. 333–371.

[22] S. Deneve and A. Pouget, "Basis functions for object-centered representations," *Neuron*, vol. 37, pp. 347–359, 2003.

[23] R. J. Douglas, K. A. C. Martin, and D. Whitteridge, "A canonical microcircuit for neocortex," *Neural Comput.*, vol. 1, no. 4, pp. 480–488, 1989.

[24] J.-R. Duhamel, F. Bremmer, S. BenHamed, and W. Graf, "Spatial invariance of visual receptive fields in parietal cortex neurons," *Nature*, vol. 389, 1997.

[25] M. Ebdon, The Uniformity of Cerebral Neocortex and its Implications for Cognitive Science School of Cogn. Comput. Sci., Univ. of Sussex, 1992, Tech. Rep. CSRP-228.

[26] M. Ebdon, "Towards a General Theory of Cerebral Neocortex," Ph.D. dissertation, Univ. of Sussex, Sussex, U.K., 1996.

[27] W. Einhäuser, J. Hipp, J. Eggert, E. Körner, and P. König, "Learning viewpoint invariant object representations using a temporal coherence principle," *Biol. Cybern.*, vol. 93, pp. 79–90, 2005.

[28] J. L. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, vol. 48, pp. 71–99, 1993.

[29] J. L. Elman, E. A. Bates, M. H. Johnson, A. Karmiloff-Smith, D. Parisi, and K. Plunkett, *Rethinking Innateness: A Connectionist Perspective on Development*. Cambridge, MA: MIT Press, 1996.

[30] P. Földiák, "Learning invariance from transformation sequences," *Neural Comput.*, vol. 3, pp. 194–200, 1991.

[31] M. Franzius, H. Sprekeler, and L. Wiskott, "Slowness and sparseness lead to place, head-direction, and spatial-view cells," *PLoS Comput. Biol.*, vol. 3, no. 8, p. e166, 2007.

[32] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980.

[33] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Netw.*, vol. 1, no. 2, pp. 119–130, 1988.

[34] P. Gaudiano and S. Grossberg, "Vector associative maps: Unsupervised real-time error-based learning and control of movement trajectories," *Neural Netw.*, vol. 4, pp. 147–183, 1991.

[35] S. Grossberg, "How does the cerebral cortex work? learning, attention, and grouping by the laminar circuits of visual cortex," *Spatial Vis.*, vol. 12, pp. 163–187, 1999.

[36] G. F. Harpur, "Low Entropy Coding with Unsupervised Neural Networks," Ph.D. dissertation, Dept. of Engineering, Univ. of Cambridge, Cambridge, U.K., 1997.

[37] G. F. Harpur and R. W. Prager, "A fast method for activating competitive self-organising neural networks," in *Proc. Int. Symp. Artif. Neural Netw.*, 1994, pp. 412–418.

[38] D. J. Heeger, "Nonlinear model of neural responses in cat visual cortex," in *Computational Models of Visual Processing*, M. S. Ly and J. A. Movshon, Eds. Cambridge, MA: MIT Press, 1991, pp. 119–133.

[39] D. J. Heeger, "Normalization of cell responses in cat striate cortex," *Vis. Neurosci.*, vol. 9, pp. 181–197, 1992.

[40] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, 2004.

[41] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, pp. 106–154, 1962.

[42] A. Karni, "The acquisition of perceptual and motor skills: A memory system in the adult human cortex," *Cogn. Brain Res.*, vol. 5, pp. 39–48, 1996.

[43] A. Karni, G. Meyer, P. Jezzard, M. M. Adams, R. Turner, and L. G. Ungerleider, "Functional MRI evidence for adult motor cortex plasticity during motor skill learning," *Nature*, vol. 377, no. 6545, p. 155, 1995.

[44] K. P. Körding and P. König, "Neurons with two sites of synaptic integration learn invariant representations," *Neural Comput.*, vol. 13, no. 12, pp. 2823–2849, 2001.

[45] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, pp. 2278–2324, 1998.

[46] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2004.

[47] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[48] N. Logothetis, "Object vision and visual awareness," *Current Opinion Neurobiol.*, vol. 8, no. 4, pp. 536–44, 1998.

[49] M. Lungarella and L. Berthouze, "On the interplay between morphological, neural and environmental dynamics," *Adapt. Behav.*, vol. 10, no. 3–4, pp. 223–241, 2002.

[50] D. Mareschal, M. H. Johnson, S. Siros, M. W. Spratling, M. S. C. Thomas, and G. Westermann, *Neuroconstructivism: How the Brain Constructs Cognition*. Oxford, UK: Oxford Univ. Press, 2007.

[51] D. Marr, "A theory for cerebral neocortex," *Proc. Roy. Soc. London. Series B., Biol. Sci.*, vol. 176, pp. 161–234, 1970.

[52] J. L. McClelland and D. C. Plaut, "Computational approaches to cognition: Top-down approaches," *Current Opinion Neurobiol.*, vol. 3, pp. 209–216, 1993.

[53] B. W. Mel, *Connectionist Robot Motion Planning : A Neurally-Inspired Approach to Visually-Guided Reaching*. New York: Academic, 1990, vol. 7.

[54] Q. Meng and M. H. Lee, "Automated cross-modal mapping in robotic eye/hand systems using plastic radial basis function networks," *Connect. Sci.*, vol. 19, no. 1, pp. 25–52, 2007.

[55] G. Metta, G. Sandini, and J. Konczak, "A developmental approach to visually-guided reaching in artificial systems," *Neural Netw.*, vol. 12, pp. 1413–1427, 1999.

[56] V. B. Mountcastle, *Perceptual Neuroscience: The Cerebral Cortex*. Cambridge, MA: Harvard Univ. Press, 1998.

[57] D. Mumford, "On the computational architecture of the neocortex II: The role of cortico-cortical loops," *Biol. Cybern.*, vol. 66, pp. 241–251, 1992.

[58] D. Mumford, "Neuronal architectures for pattern-theoretic problems," in *Large-Scale Neuronal Theories of the Brain*, C. Koch and J. L. Davis, Eds. Cambridge, MA: MIT Press, 1994, pp. 125–152.

[59] M. W. Oram and P. Földiák, "Learning generalisation and localisation: Competition for stimulus type and receptive field," *Neurocomputing*, vol. 11, no. 2–4, pp. 297–321, 1996.

[60] R. C. O'Reilly and J. L. McClelland, The Self-Organization of Spatially Invariant Representations Dept. of Psychology, Carnegie Mellon Univ., 1992, Tech. Rep. PDP.CNS.92.5.

[61] S. E. Petersen, H. Van Mier, J. A. Fiez, and M. E. Raichle, "The effects of practice on the functional anatomy of task performance," *in Proc. Nat. Acad. Sci. USA*, vol. 95, pp. 853–860, 1998.

[62] W. A. Phillips and W. Singer, "In search of common foundations for cortical computation," *Behav. Brain Sci.*, vol. 20, no. 4, pp. 657–722, 1997.

[63] A. Pouget and T. J. Sejnowski, "Spatial transformations in the parietal cortex using basis functions," *J. Cogn. Neurosci.*, vol. 9, no. 2, pp. 222–237, 1997.

[64] A. Pouget and L. Snyder, "Computational approaches to sensorimotor transformations," *Nature Neurosci.*, vol. 3, pp. 1192–1198, 2000.

[65] A. Pouget, S. Deneve, and J. R. Duhamel, "A computational perspective on the neural basis of multisensory spatial representations," *Nature Rev. Neurosci.*, vol. 3, pp. 741–747, 2002.

[66] S. R. Quartz, "The constructivist brain," *Trends Cogn. Sci.*, vol. 3, no. 2, pp. 48–57, 1999.

[67] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neurosci.*, vol. 2, no. 11, pp. 1019–1025, 1999.

[68] H. Ritter, T. Martinetz, and K. Schulten*, Neural Computation and Self-Organizing Maps. An Introduction*. Reading, MA: Addison-Wesley, 1992.

[69] E. T. Rolls and T. Milward, "A model of invariant object recognition in the visual system: Learning rules, activation functions, lateral inhibition, and information-based performance measures," *Neural Comput.*, vol. 12, no. 11, pp. 2547–2572, 2000.

[70] K. Sakai and Y. Miyashita, "Neural organization for the long-term memory of paired associates," *Nature*, vol. 354, pp. 152–155, 1991.

[71] E. Salinas and L. F. Abbott, "Transfer of coded information from sensory to motor networks," *J. Neurosci.*, vol. 15, pp. 6461–6474, 1995.

[72] K. Schulten and M. Zeller, "Topology representing maps and brain function," *Nova Acta Leopoldina*, vol. 72, no. 294, pp. 133–157, 1996.

[73] T. Serre, J. Louie, M. Riesenhuber, and T. Poggio, "On the role of object-specific features for real-world object recognition in biological vision," in *Proc. 2nd Workshop Biol. Motivated Comput. Vis.*, Tubingen, Germany, 2002.

[74] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 411–426, Mar. 2007.

[75] N. Sigala and N. K. Logothetis, "Visual categorization shapes feature selectivity in the primate temporal cortex," *Nature*, vol. 415, pp. 318–320, 2001.

[76] M. Sigman and C. D. Gilbert, "Learning to find a shape," *Nature Neurosci.*, vol. 3, no. 3, pp. 264–269, 2000.

[77] P. Sinha and T. Poggio, "Role of learning in three-dimensional form perception," *Nature*, vol. 384, pp. 460–463, 1996.

[78] S. Siros, M. W. Spratling, M. S. C. Thomas, G. Westermann, D. Mareschal, and M. H. Johnson, "Précis of neuroconstructivism: How the brain constructs cognition," *Behav. Brain Sci.*, vol. 31, no. 3, pp. 321–331, 2008.

[79] L. H. Snyder, K. L. Grieve, P. Brotchie, and R. A. Andersen, "Separate body- and world-referenced representations of visual space in parietal cortex," *Nature*, vol. 394, pp. 887–891, 1998.

[80] M. W. Spratling, "Learning viewpoint invariant perceptual representations from cluttered images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 753–761, May 2005.

[81] M. W. Spratling, "Pre-synaptic lateral inhibition provides a better architecture for self-organising neural networks," *Netw.: Comput. Neural Syst.*, vol. 10, no. 4, pp. 285–301, 1999.

[82] M. W. Spratling, "Artificial Ontogenesis: A Connectionist Model of Development," Ph.D. dissertation, Univ. of Edinburgh, Edinburgh, U.K., 1999.

[83] M. W. Spratling, "Predictive coding as a model of biased competition in visual selective attention," *Vis. Res.*, vol. 48, no. 12, pp. 1391–1408, 2008.

[84] M. W. Spratling and G. M. Hayes, M. Verleysen, Ed., "Learning sensory-motor cortical mappings without training," in *Proc. 6th Euro. Symp. Artif. Neural Netw.*, Brussels, Belgium, 1998, pp. 339–344.

[85] M. W. Spratling and M. H. Johnson, "Dendritic inhibition enhances neural coding properties," *Cereb. Cortex*, vol. 11, no. 12, pp. 1144–1149, 2001.

[86] M. W. Spratling and M. H. Johnson, "Pre-integration lateral inhibition enhances unsupervised learning," *Neural Comput.*, vol. 14, no. 9, pp. 2157–2179, 2002.

[87] M. W. Spratling and M. H. Johnson, "Neural coding strategies and mechanisms of competition," *Cogn. Syst. Res.*, vol. 5, no. 2, pp. 93–117, 2004.

[88] M. W. Spratling, K. D. Meyer, and R. Kompass, "Unsupervised learning of overlapping image components using divisive input modulation," *Comput. Intell. Neurosci.*, vol. 2009, no. 381457, pp. 1–19, 2009.

[89] J. Stone, "Object recognition using spatio- temporal signatures," *Vis. Res.*, vol. 38, no. 7, pp. 947–951, 1998.

[90] J. Stone and A. Bray, "A learning rule for extracting spatio-temporal invariances," *Netw.: Comput. Neural Syst.*, vol. 6, no. 3, pp. 429–436, 1995.

[91] S. M. Stringer and E. T. Rolls, "Position invariant recognition in the visual system with cluttered environments," *Neural Netw.*, vol. 13, pp. 305–315, 2000.

[92] M. P. Stryker, "Temporal associations," *Nature*, vol. 354, pp. 108–109, 1991.

[93] J. N. Templeman and M. H. Loew, "Staged assimilation: A system for detecting invariant features in temporally coherent visual stimuli," in *Proc. Int. Joint Conf. Neural Netw.*, New York, NY, 1989, vol. 1, pp. 731–738.

[94] J. K. Tsotsos, "Behaviourist intelligence and the scaling problem," *Artif. Intell.*, vol. 75, pp. 135–160, 1995.

[95] M. C. W. van Rossum and A. Renart, "Computation with populations codes in layered networks of integrate-and-fire neurons," *Neurocomputing*, vol. 58–60, pp. 265–270, 2004.

[96] G. Wallis, "The role of object motion in forging long-term representations of objects," *Vis. Cogn.*, vol. 9, pp. 233–247, 2002.

[97] G. Wallis, "Using spatio-temporal correlations to learn invariant object recognition," *Neural Netw.*, vol. 9, no. 9, pp. 1513–1519, 1996.

[98] G. Wallis, "Temporal order in human object recognition," *J. Biol. Syst.*, vol. 6, no. 3, pp. 299–313, 1998.

[99] G. Wallis and H. Bülthoff, "Role of temporal association in establishing recognition memory," *in Proc. Nat. Acad. Sci. USA*, vol. 98, no. 8, pp. 4800–4804, 2001.

[100] G. Wallis and H. Bülthoff, "Learning to recognize objects," *Trends Cogn. Sci.*, vol. 3, no. 1, pp. 22–31, 1999.

[101] G. Wallis and E. T. Rolls, "Invariant face and object recognition in the visual system," *Progr. Neurobiol.*, vol. 51, no. 2, pp. 167–194, 1997.

[102] V. Walsh, E. Ashbridge, and A. Cowey, "Cortical plasticity in perceptual learning demonstrated by transcranial magnetic stimulation," *Neuropsychologia*, vol. 36, no. 4, pp. 363–367, 1998.

[103] C. Weber and S. Wermter, "A self-organizing map of sigma-pi units," *Neurocomputing*, vol. 70, no. 13–15, pp. 2552–2560, 2007.

[104] C. Weber, M. Elshaw, J. Triesch, and S. Wermter, "Neural control of actions involving different coordinate systems," in *Humanoid Robots: Human-like Machines*, M. Hackel, Ed. Vienna, Austria: I-Tech Education, 2007.

[105] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, "Autonomous mental development by robots and animals," *Science*, vol. 291, pp. 599–600, 2001.

[106] G. Westermann, S. Sirois, T. Shultz, and D. Mareschal, "Modeling developmental cognitive neuroscience," *Trends Cogn. Sci.*, vol. 10, pp. 227–233, 2006.

[107] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural Comput.*, vol. 14, no. 4, pp. 715–70, 2002.

Michael Spratling received the B.Eng. degree in engineering science from Loughborough University, Loughborough, U.K., in 1991, and the M.Sc. and Ph.D. degrees in artificial intelligence and neural computation from the University of Edinburgh, Edinburgh, U.K., in 1994 and 1999, respectively.

He has held research positions both in psychology and in engineering and is currently a Lecturer in the Division of Engineering at King's College London, U.K. His research is concerned with creating biologically plausible computational models to investigate the function and development of the cerebral cortex and specifically the neural mechanisms underlying visual perception.