Learning to recognize objects on the fly: A neurally based dynamic field approach

Christian Faubel *, Gregor Schöner

Institut für Neuroinformatik, Ruhr-Universität Bochum, Germany

ARTICLE INFO

Article history:
Received 6 December 2006
Received in revised form
7 March 2008
Accepted 7 March 2008

Keywords:
Object recognition
Computer vision
Robotics
Online learning
Human–machine interaction
Scene representation
Dynamic field theory
Binding

ABSTRACT

Autonomous robots interacting with human users need to build and continuously update scene representations. This entails the problem of rapidly learning to recognize new objects under user guidance. Based on analogies with human visual working memory, we propose a dynamical field architecture, in which localized peaks of activation represent objects over a small number of simple feature dimensions. Learning consists of laying down memory traces of such peaks. We implement the dynamical field model on a service robot and demonstrate how it learns 30 objects from a very small number of views (about 5 per object are sufficient). We also illustrate how properties of feature binding emerge from this framework.

1. Introduction

Autonomous robots generate flexible behavior based on their own sensory information. One defining feature of autonomous robots is that they operate in "natural environments", that is, environments that are not specifically designed for the robots' operation, are not metrically calibrated, and may change over time. Natural environments are difficult to model, requiring non-conventional engineering approaches such as learning from experience. One reason why autonomous robots need to be able to deal with such environments is that they are often expected to share environments with human users. Typical scenarios involving robot–human interaction are service robots, production assistants, support robotics in care or clinical settings, as well as entertainment robotics and robotic interfaces to information services. In many of these cases, the human user will be directing the attention and action of the robot onto objects in the shared environment. This requires the robot to perceive these objects and to be able to understand commands referring to the objects as well as to communicate about the objects.

Building scene representations is thus a central task that must be solved in most autonomous robotics scenarios, and certainly in all robotic scenarios involving goal-directed interaction with human users. To build scene representations that support interaction with human users, relevant objects must be segmented and recognized, and pose parameters must be estimated. Recognition involves associating objects with labels that can be used in communication with the human user.

Recognition in the context of scene representation differs from the general object recognition problem of computer vision. That latter problem is still largely unsolved, in particular, when objects are embedded in natural environments. Recognizing objects within robotic scenes is both a more difficult and a much simpler problem than general object recognition in computer vision. The added difficulty comes from the requirement that new objects are learned from a very small number of views, ideally a single view, typically a handful of views. The human user will teach new objects to the robotic system by directing attention to a part of the scene (e.g., through pointing) and labeling the object. The user may also give corrective input when the robot system tries to recognize the object in new poses. The number of times a human user is willing to provide such teaching signals is very limited, however. Another aspect of this difficulty is the required flexibility in which a single label may refer to different objects at different moments in time, such as when a new exemplar of a category of objects is being referred to. The robotic recognition system must be able to update object representations, replacing no longer relevant information with new input on the same fast timescale that human users find tolerable.

These difficulties are off-set by the fact that in typical robotic scenarios only a quite limited number of objects is relevant for meaningful interaction and object-directed action within a scene. All action within the scene can be used to update
object representations. User feedback may be available throughout operation, so that a small error rate is tolerable. Moreover, many robotic scenarios provide relatively simple viewing conditions. Objects to which a robot’s attention or action is drawn may typically be placed so that the robot has them in view. Scenes will not be excessively cluttered with objects, occlusions will be limited in extent, recognition does not necessarily have to be successful from only limited component views of the object. The robotic system may also be endowed with a priori knowledge about learned objects. For instance, the approximate visual size of objects may be inferred from the learning trial and the position of the segmented object in the visual scene.

Humans themselves are, of course, extremely efficient in this category of visual tasks. In fact, our visual perception of our surroundings is, in large part, a form of scene representation in which an inventory of action-relevant items or items which had been closely examined are represented in detail, while visual details of the larger environment are much less reliably retained. A dramatic demonstration of this cognitive nature of scene representation comes from failures to detect change when visual transients are masked and the task directs attention at non-changing parts of the visual array (O’Regan, Rensink, & Clark, 1999).

This human capacity may be one motivation to search for neurally inspired solutions to the problem of building scene representations. A look at the psychophysics of scene perception is, in fact, helpful in more precisely defining the problem. There are two limit cases that touch upon the problem of building scene representations and have been extensively studied both experimentally and theoretically. The first limit case involves visual working memory as a basis for making judgments about visual scenes (Henderson & Hollingworth, 1999). Such judgments may involve the detection of a particular target object in visual search, the detection of change in discrimination paradigms, or the estimation of object features. Some of these operations involve visual working memory characterized by limited capacity and a temporal contiguity constraint, so that new objects interfere with objects previously held in working memory. Set effects, the influence of context and, relatedly, the role of reference frames point, however, to the longer-term factors (Baddeley, 1986; Fuster, 1995).

Theoretical accounts for this form of scene representation is based on the notion of feature dimensions, a small number of which is required to characterize each object (Treisman, 1998). Cortical feature maps form the neurophysiological backdrop for this conception. These maps are assumed to separately represent different feature dimensions such as orientation, spatial frequency, or color. When an object is segmented and brought into the foreground (in the language of this field, when attention is focused on the object), the feature values along the different feature dimensions are bound into an “object file”. This idea accounts for how tasks involving conjunctions of different feature dimensions differ through “feature binding” from tasks that can be solved on the basis of any individual feature dimension. How such binding in an object file would occur in neural terms is not quite clear. An alternative account postulates that there are specific neuronal mechanisms for binding, involving, for instance, correlation between neural spike trains (Malsburg, 1981; Rafal & Wolters, 2001). The neurophysiological reality of such a binding mechanism as well as its functional effectiveness are debated (see, e.g., the special issue of Neuron in 1999 (Volume 24, pages 7–125)).

The second limit case is object recognition on the basis of object categories learned over much longer timescales. In this work, the fundamental tension is between two requirements (Riesenhuber & Poggio, 1999; Serre, Wolf, & Poggio, 2005). The first, selectivity, is the capacity to discriminate between objects whose images are highly correlated. The most dramatic example is probably human face recognition: the images formed by faces are very similar across different faces, especially if compared to other kinds of visual objects, but humans are particularly astute at discriminating different faces. The second requirement is invariance of recognition under a broad set of image transformations that enables humans to recognize objects from different viewing angles, under partial occlusion, and at variable visual distances. A neurophysiologically based approach to this problem (Riesenhuber & Poggio, 2000) treats this form of object recognition as a largely feedforward computational problem, in which complex features are extracted by neurons with relatively low spatial resolution. Pooling and a hierarchical organization of such feature detectors leads to inputs from which the winner category can be determined. In this framework, the problem of binding different features belonging to the same object does not arise as a separate problem. Complex features, in a sense, already bind values along any elementary feature dimension (Riesenhuber & Poggio, 1999). Alternatively, neuronal interaction may contribute to binding within such a feedforward architecture (Wersing, Stel, & Ritter, 2001).

When humans perceive and operate in a scene, they perform both forms of object recognition. The particular exemplars of object categories are perceived and memorized in their particular rendering and pose (Henderson & Hollingworth, 1999). But humans also recognize object categories, and may use such categorical perception to structure discourse about the objects in a scene. In fact, their perceptual categories influence the representations on which such visual working memory and visual discrimination is based (Schyns, Goldstone, & Thibaut, 1998).

The problem an autonomous robot must solve when it interacts with a human user in a given environment is somewhat halfway between these two limit cases. Working memory is a first step toward building scene representations, but is too fragile and short term to achieve that building by itself. Some longer-term maintenance of acquired knowledge about objects is required. That may still fall short of the extensive learning involved in acquiring new object categories for invariant recognition.

We propose that Dynamical Field Theory is a framework, within which a neural approach to working memory may be extended to endow representations with the longer-term stability required for scene representations, while at the same time remaining close to ongoing sensory input and providing the flexibility and fast learning capabilities needed to maintain and update scene representations. Dynamic Field Theory is a neurally based theoretical approach to understanding embodied cognition. Originally developed to understand how movements are prepared (Bastian, Schöner, & Richle, 2003; Erhagen & Schöner, 2002), the ideas have been applied to a wide range of behaviors ranging from perception (Giese, 1999; Hock, Schöner, & Giese, 2003) to spatial memory (Schutte, Spencer, & Schöner, 2003).

Dynamic fields are distributions of neuronal activation defined directly over relevant perceptual or motor parameters (e.g. feature dimensions or movement parameters) rather than over the cortical surface. Conceptually, they are neuronal networks, in which the discrete sampling by individual neurons is replaced by a continuous neural field that represents the metric structure of the represented dimensions. Localized peaks of activation are units of representation. When the activation level in the peaks exceed a threshold (conventionally chosen to be zero), such peaks represent perceptual or motor decisions, both in the sense of detection decisions and in the sense of selection among competing inputs. The location of such peaks along the feature or motor dimension represents the outcome of estimation processes and encodes metric information about stimuli or motor states. The neuronal dynamics of such activation fields is governed both by inputs and by neuronal interaction, which stabilizes localized
peaks of activation through mutual excitation of neighboring field sites and global inhibition among all field sites (Amari, 1977). These peaks are attractor states of the neuronal dynamics, which may coexist bistably with subthreshold distributions of activation and may go through instabilities. Localized peaks lay down a dynamic memory trace, which in turn provides input preactivating the field when new sensory inputs are supplied (Erthagen & Schöner, 2002; Thelen, Schöner, Scheier, & Smith, 2001). The memory trace thus preshapes the field, making it easier to induce localized peaks in those locations where such peaks had previously been generated. The dynamic field concept was first applied to robotics by Engels and Schöner (1995) and Schöner, Dose, and Engels (1995) as a way to implement a subsymbolic form of memory and later as a method to stabilize decisions about movement targets and stabilize these in the face of fluctuating sensory input (Bicho, Mallet, & Schöner, 2000).

In this paper, we show how the framework of dynamic field theory can be used to develop solutions for parts of the scene representation problem. Specifically, we build a system in which a small number of very simple feature dimensions are represented by dynamic fields. These fields interact to build a simple feature representation of visual objects. In an ongoing learning process, memory traces are accumulated for different objects under interactive guidance by the human user. The system is capable of recognizing objects in new poses after a very small number of views have been acquired. The system lends itself to integration into a user-centered service robotics scenario as we demonstrate through an implementation on the service robot CoRA, which is equipped with an active stereo camera system and shares a workspace with the human user (Iossifidis, Theis, Grote, Faubel, & Schöner, 2003).

2. Methods

2.1. The cooperative robotic assistant CoRA

The anthropomorphic service robot CoRA, depicted in Fig. 1, was designed to facilitate interaction with human users. The robotic arm has the same seven degrees of freedom as the human arm and is controlled by a neurally inspired attractor dynamics which generates human-like trajectories that are easily predictable by the human user (Iossifidis & Schöner, 2004). CoRA has multiple sensory channels including artificial skin, moment sensing, gesture recognition, and gaze tracking (Iossifidis et al., 2003), that provide an intuitive user interface. Speech recognition (implemented through a custom keyword recognition approach (Fink, 1999)) and speech synthesis enable discourse between user and robot. The present work aims to extend this interface by providing the capability to associate user keywords with objects to which the user refers and to recognize these objects by activating the keyword label when the object appears in new locations and poses in the scene. For testing purposes, the speech interface was, however, typically replaced by a keyboard interface as this speeds up long test series.

2.2. Segmentation

The robotic arm is mounted on a white table, which is the workspace shared between CoRA and the human user. Objects on the table are segmented in two steps. First, all parts of the image that do not belong to the table are masked based on the known current geometry of the camera system relative to the table. Second, pixels are categorized as belonging to objects vs. to the table. Based on the assumption that the majority of pixels belongs to the table, the maximum of the grayscale distribution is tracked and a Gaussian distribution of fixed width is centered around it. All pixels within this Gaussian distribution are marked as pixels belonging to the table. The tracking mechanism enables correct object-table segmentation even under changing lighting conditions. Having segmented the image into regions belonging and not belonging to the table, we apply a cluster algorithm that fuses connected components into object hypotheses, each consisting of a blob of pixels (Born & Voelpel, 1995). In the current implementation, the user selects an individual cluster through a graphical user interface. More advanced interfaces will make user of human gesture (Theis, Iossifidis, & Steinhage, 2001).

2.3. Feature extraction

From the color image in the segmented blob we extract estimates of three feature dimension that describe the segment as a whole: (a) color; (b) size; (c) aspect ratio (see Fig. 2). Color is represented by the color histogram over the segment, which provides directly the input to the label-color dynamic field. (a) Color is represented in HSV space and only pixels with saturation and intensity values above a threshold contribute. (b) The size of the segment is computed from the number of pixels within the segment. As the apparent size of objects in the image depends the position of the object on the table, we transform the estimated image size into an approximated object size based on an estimate of the distance between the object and the focal plane of the camera. This can be done by estimating the location of the object (in terms of the center of the binary blob) on the table based on the known camera-table geometry. Finally, (c) aspect ratio is a simple form measure that is estimated by computing the ratio of the first two eigenvalues. The major axes also provide information about the orientation of the object on the table. We use this information to approximately correct for perspective foreshortening. Note that both the size and the shape estimates neglect the 3D form of the object, so that the extracted measures are only approximately invariant under view changes. Similarly, the color histogram is not a true invariant because parts of the object surface will typically occluded and may differ in color composition from the visible portion. This leads to variance in all feature values as view points are changed. The color histogram represents a form of filtering. For the two scalar features, size and aspect ratio, we compute 20 consecutive estimations in time, from which histograms are generated, effectively suppressing outliers. These histograms provide input to the corresponding label-feature fields.
Fig. 2. The object recognition architecture: Objects lying on worktable of CoRA are segmented. Whole-segment features are extracted and fed in a label-feature field layer consisting of a set of dynamic activation fields. These are coupled to a decision layer consisting of competing dynamical neurons. Interaction within the dynamic fields, mutual coupling between the dynamic fields as well as top-down feedback from the decision layer to the label-feature fields make that this is not a feedforward architecture.

Fig. 3. A label-color field: On the left, the fields are at their negative resting level, although shared input from the camera, depicted along the hue axis, drives up a ridge at the dominant hue value. One can also see small ripples reflecting input from memory traces that are individual for each label value. On the right, a homogeneous boosting input has led to the generation of a self-stabilized peak at label number 1, which suppresses alternatives.

2.4. Label-feature fields

For each of the three feature dimensions, a set of label-feature fields are set up (Figs. 2 and 3). These are collections of dynamic fields, all receiving the same input from the associated feature dimension and all endowed with the typical interaction profile of local excitation, global inhibition (Amari, 1977). For each label value, the associated dynamic field is coupled to a specific memory trace that preshapes the field by tuning it to a particular object. Across label values, the fields compete.

2.4.1. Input to the label-feature fields

The feature dimension of all label-feature fields is defined by how the input from the associated feature extraction algorithm is mapped onto the field. Given that all fields are eventually numerically integrated, we refer here to their discrete sampling by a fixed number of field sites. The color histogram contains 90 bins that sample hue space. These are mapped one-to-one onto 90 neurons sampling the label-color fields. The fields have periodic boundary conditions reflecting the periodic nature of the hue dimension. The input to the label-size field spans a range from 20 to 120, which is mapped linearly onto 72 field sites. Input to the label-aspect-ratio field ranges between 1 to 50 and is mapped onto 80 field sites. The input distributions are normalized to a total area of one and then rescaled with a different strength factor for each feature dimension. For size and aspect ratio, the input activation varies between 1/20 and 1 (when all 20 measurements lead to the same estimate). Typically, measurements are quite stable so
values close to one are most common. The input strength for size and aspect ratio is 1.27. As typical blob sizes are about 3000 pixels, the color histogram has a much larger dynamic range and is typically more broadly distributed. Its strength factor is set to 1.5.

In the dynamic equations below we denote feature input by $S(x, t)$, where $f$ = color, size, or aspect ratio, $x$ is the feature dimension, and $t$ is time.

2.4.2. Interactions within and across label-feature fields

For the feature dimensions size and aspect ratio, we use a Gaussian excitatory interaction kernel and constant inhibitory interaction

$$w_r(x - x') = w_{\text{exc}, 1} \exp[-(x - x')^2 / 2\sigma_{\text{exc}, 1}^2] - w_{\text{inh}, 1}. \tag{1}$$

(All parameters are described and their values are listed in the Appendix.) This ensures that along these feature dimensions a single localized peak is selected, which makes sense given that an object should only have one size and aspect ratio at a time. As objects can have multiple colors, we enable multi-peak solutions by using a difference of Gaussian interaction kernel (“mexican hat”) together with a constant inhibitory component.

$$w_r(x - x') = w_{\text{exc}, 1} \exp[-(x - x')^2 / 2\sigma_{\text{exc}, 2}^2] - w_{\text{inh}, 2} \exp[-(x - x')^2 / 2\sigma_{\text{inh}, 2}^2] - w_{\text{inh}, 3}. \tag{2}$$

Only field sites with sufficient levels of activation contribute to interaction. This is mediated by applying a sigmoidal threshold function to neural activation level, $u$,

$$\sigma_i(u) = \frac{1}{2} \left( 1 + \frac{\beta u}{1 + |\beta u|} \right) \tag{3}$$

where $\beta$ determines the degree of nonlinearity and was chosen differently for different dimensions, $i$. Throughout the paper, we denote different values for these parameters with the index, $i$, of the sigmoidal function. Mathematically, the intra-field interactions take this form:

$$W_f(x, t) = \int dx' w_r(x - x') \sigma_i[u_{f, i}(x', t)]. \tag{4}$$

Across labels, all dynamic fields compete. This is mediated by inhibitory interaction, scaled to the total thresholded activation, with a relatively soft sigmoid, $\beta_2$, which allows for interaction among subthreshold peaks:

$$W_{\text{across labels}, f}(x, t) = w_{\text{inh}, 4} \sum_{l \neq f} \int dx' \sigma_2[u_{f, l}(x', t)] \tag{5}$$

where the sum goes over all competing label fields, $l$, different from the current field, $f$.

Across the different feature dimensions, dynamic fields representing the same label value are mutually excitatory, boosting their respective competitive advantage.

$$W_{\text{across features}, f}(x, t) = w_{\text{exc}, 2} \sum_{l \neq f} \int dx' \sigma_2[u_{l, f}(x', t)]. \tag{6}$$

2.4.3. Preshaping label-feature fields

Peaks in each label-feature field, $u_{f, i}(x, t)$, leave a memory trace in an associated preshape field, $p_f(x, t)$. These memory traces, in turn, provide input to the activation field.

The memory trace is implemented by a dynamical system

$$\tau_{p_f}(x, t) = \alpha_{\text{dec}} \sigma_2(l_{\text{event}, f}(t)) \lambda_{\text{build}} \sigma_1[u_{f, i}(x, t)]$$

$$+ \lambda_{\text{decay}} (1 - \sigma_1[u_{f, i}(x, t)])$$

$$\times \left( -p_f(x, t) + \sigma_1[u_{f, i}(x, t)] \right) \tag{7}$$

which looks complicated, but is based on a simple logic. The memory dynamics is a simple relaxation to the sigmoided input driven by $-p_f - \sigma_1(u)$. The rate depends on a number of factors. The user has the possibility of suppressing any update of the memory trace ($\alpha_{\text{user}} = 0$) if the recognition is erroneous. The rate is also zero if a peak-detector, $\sigma_{\text{peak}} \in [0, 1]$, does not detect a peak in the associated activation field. This function $\sigma_{\text{peak}}$ is built by integrating over the sigmoided activation field and normalizing the response to either 0 or 1. The memory trace is updated only during a short time interval after a peak has appeared in the activation field.

This is implemented by the time-varying parameter, $\lambda_{\text{event}, f}(t)$:

$$\tau_{\text{event}, f}(t) = -\lambda_{\text{event}, f}(t) + \Delta(\alpha_f) \tag{8}$$

which is triggered when the value of the peak detector $\sigma_{\text{peak}}$ changes from 0 to 1, and decays within $\tau_{\text{event}}$. Here, $\Delta$ is a discrete form of the derivative returning a one after a switch from 0 to 1 for a short time interval (the Euler step of the numerics). The rate, $\lambda_{\text{build}}$, at which a memory trace is built at a location, $x$, that matches the location of the activation peak is larger than the rate of decay, $\lambda_{\text{decay}}$, at other locations.

2.5. The decision layer

The decision layer is constituted by a number of competitive dynamic neurons, $u_{\text{dec}, l}(t)$, representing all object labels.

$$\tau_{\text{dec}}(l, t) = -u_{\text{dec}}(l, t) + h_2 + \sum_{f \neq l} \sigma_3(l, t)$$

$$- w_{\text{inh}, 5} \sum_{f \neq l} \sigma_4[u_{\text{dec}, f}(t)] + w_{\text{exc}, 3} \sigma_4[u_{\text{dec}, l}(t)]. \tag{9}$$

In the absence of input, the decision neurons are at the resting level $h_2$. The suprathreshold activity of the label-feature field, normalized for each label by the excitatory part of the interaction kernel, provides excitatory input into the decision layer: $S_l(t) = w_{\text{exc}, 4} \int dx \sigma_2[u_{f, i}(x, t)]$.

Each decision neuron has self-excitatory feedback loop, $w_{\text{exc}, 3} \sigma_4[u_{\text{dec}, l}(t)]$, and receives inhibition $\sum_{f \neq l} \sigma_4[u_{\text{dec}, f}(t)]$ from all other neurons. This connectivity enforces the stable selection of one neuron when input is provided. Once the activity of a decision neuron passes threshold, inhibition is spread to all non-matching labels in the label-feature fields: $w_{\text{inh}, 6} \sum_{f \neq l} \sigma_4[u_{\text{dec}, f}(t)]$. This helps to suppress input that votes for other labels.

2.6. Complete label-feature dynamics

Together, these contributions determine the dynamics of the label-feature fields:

$$\tau_{\text{u}_{\text{f}, i}}(x, t) = -u_{\text{f}, i}(x, t) + h_1 + h_{\text{recog}} p_{\text{f}}(x, t) + h_{\text{learn}, f} \sigma_5(\alpha_{\text{learn}, f})$$

$$+ \omega_{\text{input}} S_f(x, t) + W_f(x, t) - W_{\text{across labels}, f}(x, t)$$

$$+ W_{\text{across features}, f}(x, t)$$

$$+ w_{\text{preshape}} p_f(x, t) (1 - \sigma_1[u_{\text{dec}, l}(t)]) (1 - \alpha_{\text{learn}, f})$$

$$- W_{\text{dec}}(l, t). \tag{10}$$

The $h$-parameters determine the resting level of the field in the absence of any inputs. These differ in the different modes of operation of the system, which will be discussed next. The modes are represented by the switches $\alpha_{\text{learn}, f} \in [0, 1]$ and $\alpha_{\text{recog}} \in [0, 1]$. The input from the memory trace is turned off when the decision layer has reached a decision $(1 - \sigma_1[u_{\text{dec}, l}(t)] = 0)$ as well as in the teaching mode $(1 - \alpha_{\text{learn}, f} = 0$, see immediately below).
2.7. Operational modes of teaching, recognition, and learning

Which of the three modes of teaching, recognition, or learning the dynamics operates in is controlled by the user.

2.7.1. Teaching

In the teaching mode, $\alpha_{\text{learn}}$ is set to 1, otherwise to 0. The user provides a cue that an object must be recognized, the label-feature fields are boosted homogeneously across all labels by $h_{\text{recog},f}$. This shifts all fields into a bistable regime, in which they may form self-stabilized peaks if stimulated. Input from the cameras is slowly ramped up, driving activation levels up. As the fields begin to build peaks, they compete with each other. Those label fields in which current input best matches the memory trace stored on previous trials are most likely to generate a self-stabilized peak. The longer selection takes, reflecting multiple similar matches, the stronger the relative weight of current input. This is controlled by a dynamics of the strength of that input, $\omega_{\text{input},f}$, which has an attractor at $\omega_{\text{base},f}$. If the system is not in recognition mode ($\alpha_{\text{recog}} = 0$) or if the system has already recognized an object in recognition mode ($1 - \alpha_{\text{recog}}$ is $\sum \omega_{\text{dec}(l,t)} = 1$). The system instead has an attractor at larger level, $\omega_{\text{recog}}$, if it is in recognition mode ($\alpha_{\text{recog}} = 1$) and has not yet recognized the object ($1 - \sum \omega_{\text{dec}(l,t)} = 1$).

$$
\tau_u \omega_{\text{input},f} = -\omega_{\text{input},f} + \omega_{\text{base},f} \left( 1 - \alpha_{\text{recog}} + \sum \omega_{\text{dec}(l,t)} \right) + \omega_{\text{recog}} \left( 1 - \sum \omega_{\text{dec}(l,t)} \right). \tag{11}
$$

If a peak has been formed along a particular feature dimension at a label that does not match the outcome at the decision layer, $W_{\text{dec},l}$, then this peak is suppressed through inhibition projected down from the decision layer (see last term in Eq. (10)). This makes it possible to rebuild a peak along this feature dimension at a matching label based on its current input.

2.7.2. Recognition

The default mode is recognition, represented by $\alpha_{\text{recog}} = 1$. When the user provides a cue that an object must be recognized, the label-feature fields are boosted homogeneously across all labels by $h_{\text{recog},f}$. This will lift the corresponding field into a regime in which self-stabilized peaks become possible once input from the cameras is provided. Such self-stabilized peaks may leave a memory trace, which provide a competitive advantage to this specific label during recognition if new inputs metrically match the peak locations. While the boost is on, input to the label-feature fields from previous memory traces is turned off ($1 - \alpha_{\text{learn}} = 0$ in Eq. (10)), so that the activation pattern reflects the current input and lays down a veridical memory trace that is not biased by past learning episodes.

$$
\tau_u \omega_{\text{input},f} = -\omega_{\text{input},f} + \omega_{\text{base},f} \left( 1 - \alpha_{\text{recog}} + \sum \omega_{\text{dec}(l,t)} \right) + \omega_{\text{recog}} \left( 1 - \sum \omega_{\text{dec}(l,t)} \right) \tag{11}
$$

If a peak has been formed along a particular feature dimension at a label that does not match the outcome at the decision layer, $W_{\text{dec},l}$, then this peak is suppressed through inhibition projected down from the decision layer (see last term in Eq. (10)). This makes it possible to rebuild a peak along this feature dimension at a matching label based on its current input.

2.7.3. Learning

Once the dynamics has run its course in the recognition mode, a single decision neuron is active and, through the inhibitory back-projection into the label-feature layer, peaks are stabilized there in the selected label fields only. The system responds by announcing the label name. If the user does not contradict the recognition, the system switches into the learning mode, in which the memory trace dynamics is activated and the memory trace of the recognized object is updated. The user reacts to an error by indicating the correct object label. This switches the system into the teaching mode, in which the correct label neuron and fields are boosted, leading to suppression of the erroneous activations, and to building of activation patterns at the correct label fields and updating of the associated memory trace. In either case, recognition is always linked to some form of learning.

3. Results

3.1. Evolution of generalization and error rate during learning

To assess the learning and recognition capabilities of the system, we mix a regular teaching scenario with tests of generalization to new views. The basic teaching scenario consists of two phases. First, a new object is presented in the middle of the table with its longer axis perpendicular to the viewing axis. The system is in teaching mode. All of the 30 objects used in these tests (Fig. 4) were taught once that way. Second, all objects are again presented in the 9 different poses illustrated in Fig. 5 in the recognition mode. These sweeps begin with the same central position used during the initial teaching trial. Then the rotated position is used for all objects, followed by another rotation, followed by a translation to a closer location in the same three rotations and finally a shift to a location further back in the three rotations, all this in the order is illustrated in Fig. 4. Note that in response to any erroneous recognition, re-teaching occurs automatically, not requiring, of course, a new stimulus presentation. This makes for a total of 300 learning trials.

After each learning sweep through all objects in one of the 9 poses, we assess the capacity to generalize recognition to new views. This makes use of the 9 locations and orientations illustrated in Fig. 6, which were not used during learning. These include new views from a more lateral angle. This series of tests evaluates the system at a particular stage of learning. Unlike in the usual mode of operation, learning is therefore deactivated during these tests. At each learning step, this makes for a total of 270 tests.

The performance is illustrated in Fig. 7. Note that after learning only 5 views, the recognition rate in new poses is already above 80%. The recognition rate saturates after the 8th view at around 88% correct.

On the first learning set all trials are, by definition, relearning trials. The rate at which objects are incorrectly recognized and thus relearned decreases over learning sweeps (Fig. 7). After the 4th learning pose, in which the objects are first moved to the front position, generalization improves leading to a significant drop in relearning episodes. For most objects, the variance of their feature values has been captured by then. In total, of the 270 learning trials (excluding here the first 30 pure teaching trials), rejection of the returned response by the user occurs 85 times, an average of 2.8 user interventions per object (Fig. 8). For a third of the objects, a single episode of relearning or less is sufficient.
**Fig. 5.** The 9 poses used during learning. The poses used in the order indicated by the inset number. At each of the poses all objects are recognized one after the other.

**Fig. 6.** The 9 poses used to assess generalization after each of the 9 learning poses. Every object is presented in all 9 test poses while learning is deactivated.

**Fig. 7.** Evolution of recognition performance over the course of learning. For the initial teaching sweep (number 0) and each of the 9 learning poses, the first bar indicates the number of erroneous recognitions triggering relearning (left scale: in percent objects being relearned: $1 = 30$ objects, $0.1 = 3$ objects). The second bar indicates the system’s performance during the generalization tests (right scale: percent correct recognitions across all 270 test trials).

**Fig. 8.** Over the 9 learning trials for some objects re-teaching by the user is never necessary, some objects have to be re-taught on nearly every trial. This plot counts how many objects have to be re-taught how often.

A second scenario used much shorter learning: We used only 3 teaching views for each object, one view in the middle of the table, zero rotation, another at 45 deg rotation in the front location and a third at 90 deg rotation in the back of the table. After this learning phase, the same 270 test views were used to assess performance, while learning was deactivated. In the second scenario, the average recognition rate in the testing phase was 83%.
Analyzing performance after learning

Analyzing the performance after the learning phase has been terminated, reveals that more than half of the objects have a recognition rate of 100% (Fig. 9). Three quarters of the objects have a recognition rate higher than 85%. Four objects are largely responsible for the ceiling in performance. These have a recognition rate smaller than 60%.

Analysis reveals two mechanisms of recognition error. To discuss these, a confusion matrix is displayed in Fig. 10. This registers how often each response was made to each stimulus. Three of the objects are confused more frequently than the rest: The blue gripper was confused with the blue soldering iron, the red cutter with the glue stick and the can of fish with the honey. In all three cases, the source of the error is that the object with which confusion occurs has similar and overlapping feature values along all three feature dimensions. The fourth problematic object is the yellow stapler. It is confounded with different objects on different occasions. It turns out that size and aspect ratio are highly variant features for this object, which causes their contribution to recognition is not very significant (see also below). On the remaining, significant feature color, the yellow stapler overlaps with at least 4 other objects (see also Figs. 11–13). One remarkable observation is that the confusion matrix is asymmetric. Whether objects are vulnerable for confusion depends not only on the similarity of their feature encoding, but also on the inherent specifical strength of their memory trace (e.g., on it being narrower or broader). We shall now examine the factors affecting recognition more closely.

3.3. Mechanisms of recognition

3.3.1. Emergent categories

How do the memory traces specify the correct match? After learning, the memory traces within the different feature dimension are structured in a way that suggests that a limited number of categories has emerged. For instance, along the feature dimension color, the memory traces appear to be partitioned into 7 different colors (red, orange, yellow, green, turquoise, blue and violet, see Fig. 11). Along the feature dimension size there is more overlap, but some partitioning is still apparent. Four overlapping size categories (small, medium, large and very large) may likewise be postulated (Fig. 12).

We bring this up to ask this question: Is it possible, that the recognition capacity of the system is limited by this form of emergent categorization? For instance, if the system had in effect only stored which category an object is in with respect to the three feature dimensions, then the maximal number of discriminable objects would be $7 \times 4 \times 4 = 112$ objects. An answer can be obtained by searching for objects in our database that are identical in the categorical code. If these can be discriminated by the system, then we know that the fine grained metric information encoded in the memory traces is actually used. This turns out to be the case. In terms of categorical feature encoding it would be impossible, for instance, to discriminate the red beans from the box of slide frames (both are categorically red, big and square). This two objects are
very well discriminated, however. There are many more examples of this nature. The memory traces thus work by amplifying subtle differences in the match of input to memory traces. A slightly better match leads to a competitive advantage and to successful recognition.

3.3.2. Similarity matrix

A direct assessment of the degree of similarity between different objects can be obtained by looking at the activity in the decision layer before a decision has been made. Fig. 14 shows how active the different neurons are when the different objects are presented. The activation levels are read out when the first neuron crosses a threshold level. Comparing this matrix to the confusion matrix (Fig. 10), note that even objects that were never confused are facing active competition with other neurons activated before final recognition. There are only 5 objects that do not face any recognition.

One could be tempted to predict that similarity determines how much recognition is slowed down by inhibitory competition. Objects competing with many similar items may take longer to build up significant activation in the decision layer. This prediction does not bear out: Time to recognition is largely uncorrelated with the number of similar competitors (Fig. 15). So what does determine the rate of recognition for different objects?
3.4. Interaction among feature dimensions and binding

We have seen that the fully coupled system of label-feature fields and the associated dynamic decision layer can recognize objects based on metric information along very simple feature dimensions. Does the system combine information along the different feature dimensions? Does it, in effect, bind feature dimensions to disambiguate objects? How does the system control the relative weight of the different feature dimensions? To answer these questions we use a simpler setting in which only four different objects are learned (Fig. 16).

A first point is to understand that dimensions along which different objects compete are slowed down. This makes these dimensions in effect weaker contributors to the recognition decision: the other dimensions move ahead, contributing increasing inhibition of different choices and promoting through excitatory interaction like choices. Finally, the decision level dictates the selection in any undecided feature dimension.

Fig. 17 illustrates this fact. The banana is similar to the red pepper in size, but not in aspect ratio. Similarly, the banana competes in color but not in size with the lemon. The competition along size between the banana and the red pepper slows that dimension down (see the right-hand side of Fig. 17). Similarly,
Fig. 15. The dashed line is the number of similar objects as estimated from the similarity matrix for each object. The blue line is the time to recognition for the object. There is no correlation between these two measures.

Fig. 16. This set of four objects is used to examine interaction between feature dimensions and binding. These are chosen to create complex patterns of similarity along different feature dimensions. For example, the banana shares with the zucchini the aspect ratio, with the lemon the color and with the pepper the size. Some objects are similar on two feature dimension (e.g., size and aspect ratio for banana and zucchini).

Fig. 17. On the left, the memory traces of the red pepper (solid) and the banana (dashed) are shown along the feature dimensions size (top) and aspect ratio (bottom). Along size the two objects overlap strongly, while they differ in aspect ratio. On the right the time course of the maximal activation levels in the size feature field (green) and the aspect ratio feature field are shown (red). This is in response to being presented with the red pepper. Note how competition slows down the selection process in the size dimension (green), so that the correct (solid) and incorrect (dashed) choice separate more slowly and later for the aspect ratio dimension (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the competition along color between banana and lemon slows the color dimension down (not shown for simplicity). The aspect ratio becomes the dimension along which a separation of the banana from the other objects first begins to appear. Aspect ratio then becomes the dominant dimension, pulling the decisions within the other two feature dimensions along.
On this basis we can understand how the system deals with “binding”. To illustrate this, we include all four objects. Within each feature dimension there is at least one object that is similar to the banana (Fig. 18). The banana competes with the lemon on color, with the zucchini on aspect ratio and with both the zucchini and the red pepper on size. Unlike in the previous case, no single feature dimension can uniquely decide the competition. Only when the different feature dimensions of the banana “know of each other” or are bound, can the recognition take place.

Fig. 19 shows how this binding is achieved by spreading activity from one feature dimension to another. No single dimension is decisive. The activation of the banana overlaps with the activation of at least one object in each dimension. Only the banana is competitive on all dimensions. If a fluctuation pushes the banana activation higher in one feature dimension, the coupling of the feature dimensions is sufficient to give it the decisive competitive advantage in other fields.

4. Discussion

4.1. Summary

We have presented an approach to object recognition that is entirely phrased within neuronal dynamics. A set of label-feature dynamic activation fields represents objects by stabilizing peaks of activation located over the feature values that characterize a view of an object. We used very simple features characterizing the segmented visual object as a whole: its hue histogram, a size estimate and its aspect ratio, the latter two corrected for visual distance and perspective distortions. These feature dimensions are only approximately invariant under view changes.

These label-feature dynamics are interconnected to a decision layer of competing neurons. Each label-feature field is preshaped by activation patterns stored in memory traces of previous activation patterns held when the visual object was previously seen. Recognition is a dynamic process, started by homogeneously boosting activation in all fields. The previous neuronal dynamics stabilizes a coherent attractor state, a set of self-stabilized peaks that reflect the best match between current sensory input and patterns of preactivation.

A very simple memory trace mechanism supports the learning of the patterns of preactivation which control the selection of the appropriate label in response to new views. The suprathreshold self-stabilized peaks in the label-feature fields lay down a memory trace of their activation during a small transient window after the peaks are first generated. Thus, every instance of recognizing an object reinforces and updates its memory representation.

The user intervenes in two forms. During explicit teaching trials, the user indicates the label to be associated with an object. This boosts the fields linked to that label, in which self-stabilized peaks are then induced by current sensory input. A memory trace of these is then laid down. During regular operation, the user may reject the result of a recognition process, again specifying what the correct response would have been. This destabilizes the activation pattern realized at that point and prevents the laying down of its memory trace. As during teaching, a boost to the fields associated with the correct label reinstate an activation pattern reflecting the current input, which is used to update the memory trace associated with the correct label.

Given the rapid learning and updating of the memory trace, the approach lends itself to support scene representation. Here we have shown only the simplest function, learning a limited number of objects from a small number of views under user guidance and recognizing these objects in new poses within a limited workspace. Our tests involved the learning of 30 daily life objects. We found that only approximately 5 views were required to achieve recognition rates of above 80% for new views. Even training on only 3 views, which co-sampled 3 orientations and 3 visual distances yielded recognition rates of about 83% at other viewing locations in the workspace. Recognition is obtained in real time sufficiently rapid to satisfy the human user (faster than 1.5 s in our implementation).

Competitive as well as excitatory interaction among the label-feature fields as well as feedback from the decision layer enable the system to link the feature information across the different label-feature fields. Field dimensions on which an object has a clear competitive advantage take the lead in the dynamic recognition process when the other dimensions encounter close competition due to similarities with other objects. The architecture thus
automatically reweights the different dimensions through subtle shifts in the timestructure of recognition.

In a simple toy example, we showed how interaction between the feature dimensions may support recognition even when objects compete within all feature dimensions with different alternatives. In such cases, only the feature conjunctions, the conjoined ensemble of values along all feature dimensions are sufficient to identify the object. This form of binding emerges from the mutual neuronal interactions within the ensemble of label-feature fields and decision neurons.

4.2. Embedding in cognitive science

We are building of a line of work that has taken the ideas of dynamical field theory, which originated in the domain of motor planning (Erlhagen & Schöner, 2002; Kopeck & Schöner, 1995) and has moved these toward embodied cognition (Schutte & Spencer, 2002; Spencer & Schöner, 2003; Thelen et al., 2001). This involved the step of enabling the continuously valued dynamical fields to respond categorically when peaks are induced by homogeneous boosting inputs (Schneegans & Schöner, 2008). John Spencer and his colleagues have used this framework to develop accounts for how elementary cognitive functions like recall, discrimination and change detection emerge from dynamical fields defined over continuous metric feature dimensions (Simmering, Spencer, & Schöner, 2006). A model, supported by empirical data, of how people use spatial language to describe metric spatial information has developed a similar architecture as used here, an ensemble of metric dynamical fields in mutual interaction with a decision layer of competing neurons (Lipinski, Spencer, Samuelson, & Schöner, 2006). On a larger scale, this framework is consistent with Stephen Grossberg’s notion of neuronal dynamics as the basis of cognitive function going back over 30 years (Grossberg, 1973, 1980).

Our approach could be viewed as a neural dynamic version of Anne Treisman’s notion of an object file (reviewed in Treisman (1998)). In our architecture, only objects that are in the foreground, that is, that have been segmented and to which attention has been drawn, would be candidates for engaging the competitive neuronal dynamics of recognition. Under these circumstances, objects are characterized by the ensemble of simple feature descriptions, as in Treisman’s conception. Binding occurs in the sense that the conjunctions of feature values matter when this is required, that is, when recognition cannot be based on a distinctive individual feature dimension. Binding is emergent in our picture, that is, there is no explicit process responsible for binding itself nor is there a categorical distinction between “bound” and “not bound” forms of object representation.

The way binding arises in our object recognition framework points to an ecology of binding, that might make it less relevant in complex, real-life object recognition. Binding as a problem arises, essentially, because within every single feature dimensions there are one or multiple objects that are close competitors due to similarity within that dimension. Moreover, across dimensions these competitors differ. It appears, that this kind of configuration would become less and less likely as the number and complexity of feature dimensions is scaled up. In other words, binding as a problem may arise only when there is a detailed balance that prevents any sub-ensemble of feature dimensions from dominating the recognition process. When the number and complexity of features is scaled up, it remains important or even critical, that the strength of the contributions of individual feature dimensions are flexibly reweighted depending on how discriminative these dimensions are for the recognition task at hand.

4.3. Embedding in computer vision and robotics

This last point concurs with the criticism of the binding concept voiced from the perspective of object recognition in the larger sense (Riesenhuber & Poggio, 1998). The influential work of Riesenhuber and Poggio (2000) has provided a neural architecture of invariant visual object recognition, which is aimed at a much broader problem than that addressed here. In a feedforward, hierarchical network complex features are extracted over broad receptive fields that enable recognition of objects under varied viewing conditions, including recognition from key features that do not require previous segmentation of the object (Heisele, Serre, Mukherjee, & Poggio, 2001). In its most recent implementations, this approach achieves substantial recognition rates for general purpose object recognition under challenging invariance constraints (Serre et al., 2005). Introducing sparsification and lateral inhibition further boosts performance and speeds up recognition (Mutch & Lowe, 2006). Even so, these approaches are at this point quite far from real time.

Also, this form of recognition solves a slightly different problem from the one addressed by us. Views of objects in images are assigned into learned categories. This is more general because it makes no assumptions about placing the views in the context of a scene, which strongly restricts the range of candidate objects. It is more specific in the sense that the 3D structure of objects is irrelevant and categorically different views of objects, e.g., from the front vs. from the side are assigned to different categories. The learning of the object categories is allowed to take a substantial number of learning, and includes the learning of optimal features for these objects, although generalization across object categories is exploited. Online updating of categories on a fast timescale is not a theme.

Closer to the problem we try to solve is a recent expansion of the approach of Lowe (2004) in which a limited vocabulary of features is learned (Murphy-Chuitian, Aboutalib, & Triesch, 2005). This reduction in the feature representation enables performance in real time. The approach is used to recognize 50 objects in everyday scenes. Performance is similar to ours, including a very limited number of views required to recognize objects once the feature vocabulary has been learned. This work does not require prior segmentation, all the more strengthening the appeal of the method. While the features are inspired by the feature maps in the visual system of mammals, the mode of processing is not neurophysiologically based or plausible. By contrast, our approach is a faithful implementation of a process-model of visual recognition, in which algorithms only serve to numerically solve dynamical systems equations that may be directly embodied by the cortical neuro-dynamics of the visual system.

Quite similar to our approach to binding are the neurodynamic architecture of Ontrup, Wersing, and Ritter (2004) and Wersing et al. (2001). These orders combine feedforward feature extraction architecture, that is inspired by visual cortex, with neural interaction which mediates binding between feature maps. Their models are at a finer grain of representation than ours and are put to other uses, primarily to induce grouping of feature points that helps to improve segmentation, links contours and supports form perception. The models thus address a problem different from ours. In particular, they are not concerned with learning specific object representations.

Generic neuronal network approaches using supervised learning, in principle, be used to learn visual objects (for examples from a vast literature see Bishop (1995)). Note that without problem specific preprocessing such learning typically requires many presentations of stimuli in variable poses and does not come anywhere near the performance on small sets of objects with only a handful of views as reported here. To achieve performance likes
ours much more specific assumptions about the structure of a neuronal network must be made. Our model essentially is a dynamical neural network specifically prestructured for the task and learns in an unsupervised fashion.

Our instance on using very simple features rather than optimized ones suggests a relationship to the SEEMORE system by Mel (1997). It uses richer feature sets, which are weighted based on how variant and discriminative they are, and then applies nearest neighbor classification. This work was not aimed to minimize the number of training examples and does require much more learning than ours, but then also succeeds on larger numbers of categories. Again, online learning is not part of the SEEMORE approach.

A different theoretical frame work, primarily based on probabilistic methods was used by Fei-Fei, Fergus, and Perona (2003) to learn object categories from a very small number of views. The task solved by these authors is quite different, however. Each object category is separately tested against background, so the system returns a probability that a particular object class is present in the image. This particular model does not make the competitive decision among a set of objects (although Bayesian approaches, in general, can do that). Also, the object categories in this work are really view categories, in which larger view variations are assigned to separate categories.

Our approach based on dynamic fields with memory traces could be viewed as a specific implementation of Bayesian thinking. The memory traces essentially generate priors from experience. The field dynamics drives recognition by fusing current input with this priors. Following recognition, the memory trace is updated, generating the next prior. Although our biologically inspired approach does not use the formal mathematics of Bayesian decision making, it is thus conceptually consistent with Bayesian thinking (see Rao (2006) for an introduction to Bayesian thinking from a neural modelling perspective). One difference between Dynamic Fields and probabilistic decision making is that the activation distributions generated by dynamic fields are not normalized. In fact, the norm of activated portions of the fields may become zero, expressing the “off” states of the different systems. Probabilistic thinking does not always require normalization, but is based on the assumptions that distributions are normalizable.

4.4. Outlook

In the presented work we have demonstrated the isolated function of object recognition in a scenario involving human user interaction. One advantage of our radical neuro-dynamics approach is that integration with attractor dynamics based architectures for autonomous robotics (Erhlag & Bicho, 2006; Menzner, Steinhage, & Erhlag, 2000; Schöner et al., 1995; Steinhage & Schöner, 1998) is very simple. An immediate step we are making now is integrating this work into an ongoing process of building and maintaining scene representations. We are, for instance, using dynamical fields to autonomously sample the workspace for visually attractive segments, which are then sequentially analyzed and on which object recognition is performed. Moreover, we have implemented a first, simple system that organizes the interactive dialogue of a human user with the system (Sandamirskaya & Schöner, 2008).

To scale the method presented here beyond the limited sets of objects we will also need to enrich the feature set as well as go beyond the global features that are based on presegmented scenes. This can be done by representing objects as multi-peak patterns of activation, each peak representing characteristic feature points in an object. Recent work within the cognitive science domain on multi-peak representations as a basis for visual working memory and change detection (Johnson, Spencer, & Schöner, 2006) provides the setting within which we will build recognition systems of this kind.

Acknowledgements

Financial support from the BMBF DESIRE Project FKZ:01ME010G. We appreciate extensive discussion and sharing of ideas with John P. Spencer, Jeff Johnson, John Lipinski, and Vannessa Simmering from the University of Iowa and with Ioannis Liossidis and Yuliia Sandamirskaya from the Institut für Neuroinformatik.

Appendix. Parameter values

The values of the model parameters are primarily constrained by the different dynamical regimes in which the dynamic fields must operate. These regimes are demarcated from each other through dynamic instabilities (Schöner, 2007). For instance, the recognition mode requires that label-feature fields are in the selection regime, in which a single, self-stabilized peak is stable. The widths of inputs are dictated by the resolution along the feature dimensions. Once the constraints were satisfied, we did not perform extensive parameter tuning.

For the feature dimensions size and aspect ratio, the interaction kernel parameters were \(w_{exc,1} = 2.0, \sigma_{exc,1} = 3.0\), and \(w_{inh,1} = 1.0\). For the interaction kernel of the color-label fields we used \(w_{exc,1} = 2.0, \sigma_{exc,2} = 2.5, \sigma_{inh,2} = 1.0\), and \(w_{inh,2} = 6.0\) and \(w_{inh,1} = 0.2\). The parameters for sigmoidal threshold function along the feature dimensions were set to \(\beta_1 = 50\) for all feature dimensions. For the interaction across the labels we used \(w_{abh,4} = 0.1\) and \(\beta_2 = 28\). The excitatory weight between feature dimensions was set to \(w_{exc,2} = 2.3\). The preshape dynamics build and decay rates were the same for all feature label fields and set to \(\lambda_{build} = 0.2\), \(\lambda_{decay} = 0.005\) respectively. The decision layer’s resting level was set to \(h_2 = -2.7\), the inhibitory weight was \(w_{inh,5} = 2.9\). The excitatory weight was \(w_{exc,3} = 2.5\), and for the parameter for the sigmoidal function we chose \(\beta_3 = 800\). The weight factor for the inputs to the decision layer was set to \(w_{exc,4} = 2\) for all label-feature fields. The weights for the contributing memory traces were for all feature dimensions set to \(w_{preshape} = 0.28\). The inhibitory weight for suppressing peaks in other labels was \(w_{inh,6} = 0\). The parameters of the label feature fields where the following for all feature dimensions: The timescale of the dynamical equation was \(\tau = 10\), the resting level was \(h_7 = -5.5\), the homogeneous boost for the recognition mode was \(h_{base,f} = 9.1\), and the homogeneous boost for learning was \(h_{base,u} = 6.8\). For the scaling of the input we used a timescale \(\tau_{in} = 100\) for all feature dimensions. The color feature dimension’s base scale was set to \(\alpha_{base,f} = 1.5\) and the maximal scale to \(\alpha_{cap} = 2.1\). For size and aspect ratio we chose \(\alpha_{base,f} = 1.27\) and \(\alpha_{cap} = 1.5\). The timescale of the time-varying parameter \(\lambda_{event,f}(t)\) was chosen as \(\tau_{event} = 8\).

References


