A biologically motivated visual memory architecture for online learning of objects

Stephan Kirstein, Heiko Wersing, Edgar Körner

2008

Preprint:

This is an accepted article published in Neural Networks. The final authenticated version is available online at: https://doi.org/[DOI not available]

Available online at www.sciencedirect.com
ScienceDirect
Neural Networks 21 (2008) 65–77
Neural Networks
www.elsevier.com/locate/neunet

A biologically motivated visual memory architecture for online learning of objects

Stephan Kirstein*, Heiko Wersing, Edgar Körner

Honda Research Institute Europe GmbH, Carl-Legien-Str. 30, 63073 Offenbach am Main, Germany

Received 29 June 2006; accepted 9 October 2007

Abstract

We present a biologically motivated architecture for object recognition that is based on a hierarchical feature-detection model in combination with a memory architecture that implements short-term and long-term memory for objects. A particular focus is the functional realization of online and incremental learning for the task of appearance-based object recognition of many complex-shaped objects. We propose some modifications of learning vector quantization algorithms that are especially adapted to the task of incremental learning and capable of dealing with the stability-plasticity dilemma of such learning algorithms. Our technical implementation of the neural architecture is capable of online learning of 50 objects within less than three hours.

Keywords: Learning vector quantization; Incremental and life-long learning; Stability-plasticity dilemma; Hierarchical feature extraction

1. Introduction

Learning and recognition of visual objects is a task so easy for humans, that we rarely notice its importance in carrying out everyday exercises. Especially the robustness and the capacity for learning countless objects during the entire life makes the human visual system superior to all currently existing technical object recognition approaches. Another aspect of human vision is the capability of quickly analyzing and remembering completely unknown objects. This online learning ability is also relevant for many cognitive robotics and computer vision tasks, e.g. for incrementally increasing the knowledge of an assistive robot (Steil & Wersing, 2006).

In this paper we propose a biologically motivated recognition architecture that combines a hierarchical model of the ventral pathway of the human visual system (Wersing & Körner, 2003) with a memory model implementing interacting short-term and long-term memory. The target of the model is to obtain a flexible object representation that is capable of high-performance appearance-based object recognition of complex objects together with a particularly rapid online learning scheme.

Several recent studies (Palmeri & Gauthier, 2004; Tarr & Bülthoff, 1998) presented strong biological evidence in favor of view dependent and appearance-based representations in the human brain, as opposed to a strongly structuralist representation using three-dimensional primitives (Biederman, 1987). One of the main modeling approaches to explain the robustness and invariance of appearance-based recognition are hierarchical feature-detection models of the ventral visual pathway. Fukushima (1980) introduced this type of model with the Neocognitron, based on sequential stages of local template matching and spatial pooling. The Neocognitron provided the starting point for the development of several recent hierarchical feature-extraction models (Körner, Gewaltig, Körner, Richter, & Rodemann, 1999; Riesenhuber & Poggio, 1999; Rolls & Milward, 2000; Wersing & Körner, 2003). Based on earlier work, we use in this contribution the model of Wersing and Körner (2003) to obtain a high-dimensional topographical feature-map representation of the visual input that already offers some invariance with regard to translation, rotation, and scaling. On top of this feature representation we develop a biologically motivated flexible object memory model that consists of a rapidly learning short-term and a slower long-term

* Corresponding author. Tel.: +49 69/89011 750; fax: +49 69/89011 759.
E-mail addresses: stephan.kirstein@honda-ri.de (S. Kirstein), heiko.wersing@honda-ri.de (H. Wersing), edgar.koerner@honda-ri.de (E. Körner).

0893-6080/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2007.10.005

memory system. The combination of both memory systems is able to cover the strong appearance variation that is generated from three-dimensional rotation of objects.

In psychology and neuroscience, the separation of memory into the two systems of short-term (STM) and long-term memory (LTM) is an established concept (Izquierdo, Medina, Vianna, Izquierdo, & Barros, 1999). These two systems, being optimized for different tasks, can be distinguished with regard to the level of detail, the number of items that can be stored and the time span the information can be memorized. The defining property of STM is the ability to learn fast and immediately recognize a once presented object stimulus, even if the object was completely unknown before (O'Reilly & Norman, 2002). In technical applications this capability is often called "one-shot learning". The storage capacity of the STM is limited to a few objects and the information can be memorized only for a relatively short period, in comparison to the LTM which can represent many objects for long periods.

A large body of experimental evidence beginning from classical work of Scoville and Milner (1957) shows, that the medial temporal lobe is involved in the transfer of information from STM to LTM, with first changes due to learning occurring in the hippocampus (Wirth et al., 2003). However, the role of the temporal lobe memory system is only temporary. After successful storing of contents in the neocortex, the LTM becomes gradually independent of the medial temporal lobe structures (Squire & Zola-Morgan, 1991). Recently it has been affirmed that the medial temporal lobe is important for both spatial and recognition memory (Broadbent, Squire, & Clark, 2004).

The transfer from STM to LTM results in a reduction of the representational effort and should be able to extract a more generalized structure of the presented objects (O'Reilly & Norman, 2002). The LTM is mainly located in the neocortex (Miyashita & Hayashi, 2000) and has a much larger storage capacity and storage duration compared to the STM. This information transfer itself is not well understood until now, but it is assumed that a consolidation process is used for this transfer, which also continues during sleep (Maquet, 2001; Buzsáki, 1996).

Our object memory model is motivated by the functional differentiation in the two STM and LTM systems of human brains. Our target is to perform supervised and online learning of object views using the STM, which has the ability to incrementally build up an object representation without destroying already learned knowledge. This STM provides fast learning, but also has a limited capacity. For the buildup of the LTM we propose an incremental learning vector quantization (iLVQ) method that realizes the transfer from the fast learning STM into the slower learning LTM, which results in a more integrated and condensed object representation. Furthermore we define several extensions to learning vector quantization (LVQ) networks (Kohonen, 1989) used for our LTM model that are necessary for our target of an incrementally and life-long learning system. We demonstrate the technical realization of the proposed approach in an interactively trainable online learning system that can robustly recognize several objects.

In the following Section 2 we discuss related work, including several online learning and life-long learning approaches. After a short introduction to the hierarchical feature processing, we define our short-term and refined long-term memory model, based on an incremental learning vector quantization approach in Section 3. We demonstrate its effectiveness for an implementation of real-time online object learning of 50 objects in Section 4, and finish with a discussion in Section 5 and final conclusions in Section 6.

2. Related work

We first define some common terms for our review of related work. The term *online learning* is used in this paper for the ability of fast learning and immediately recognizing trained stimuli, which mainly applies to "one-shot learning" methods like the adaptive resonance theory (ART) (Carpenter, Grossberg, & Rosen, 1991). A special property of online learning is the possibility of active learning with an interactive correction of errors during the training process.

We define *incremental learning* as the ability of a network architecture to allocate increasing numbers of neurons, dependent on the complexity of the current task. Such network architectures are normally initialized with a minimal number of neurons and are able to add resources based on some node insertion criteria using the training error.

An extension to incremental learning architectures are neuronal networks approaching the *life-long learning problem* and the so-called "stability-plasticity dilemma". Dealing with plasticity means that a learning method must always be able to represent newly occurring data, which normally is solved with some incremental learning architecture. On the other hand it should prevent the destruction or forgetting of already acquired knowledge and should maintain the stability of the representation. The life-long learning problem is often encountered in the form of a changing and non-stationary training set, where only a portion of data is visible to the learning method. In our setting of an object recognition task, this can mean that only the most recent objects are visible to the learning method, while the older ones that have disappeared from the training set should still be conserved.

2.1. Online learning and man–machine interaction

Most research on trainable and model-free object recognition algorithms has so far focused on learning based on large data sets of images recorded beforehand and then performing offline training of the corresponding classifiers. Since in these approaches learning speed is not a primary optimization goal, typical offline training times last many hours. This is usually caused by the natural high dimensionality of visual sensorial input, which poses a challenge to most current learning methods. Another problem is that most powerful classifier architectures such as multi-layer perceptrons or support vector machines do not allow online training with the same performance as for offline batch training.

To reduce the dimensionality of the problem, the complexity of the sensorial input has been reduced to simple blob-like

stimuli (Jebara & Pentland, 1999), for which only positions are tracked. Based on the positions, interactive and online learning of behavior patterns in response to these blob stimuli can be performed. A slightly more complex representation was used by Garcia, Oliveira, Grupen, Wheeler, and Fagg (2000), who have applied the coupling of an attention system using features like color, motion, and disparity with a fast learning of visual structure for simple colored geometrical shapes like balls, pyramids, and cubes. They represent shape as low-resolution feature maps computed based on convolutions with Gaussian partial derivatives. Using shape and color map representations the system can learn to direct attention to particular objects.

Histogram-based methods are another common approach to tackle the problem of high dimensionality of visual object representations. Steels and Kaplan (2001) have studied the dynamics of learning shared object concepts based on color histograms in an interaction scenario with a dog robot. The object representation allows online learning using the limited computational resources of the pet robot, but lacks a stronger concept of shape discrimination. Another model of word acquisition, that is based on multi-dimensional receptive-field histograms (Schiele & Crowley, 2000) for shape and color representation was proposed by Roy and Pentland (2002). The learning proceeds online by using a short-term memory for identifying the reoccurring pairs of acoustic and visual sensory data, that are then passed to a long-term representation of extracted audiovisual objects.

Arsenio (2004) has investigated a developmental learning approach for humanoid robots based on an interactive object segmentation model that can use both external movements of objects by a human and internally generated movements of objects by a robot manipulator. Using a combination of tracking and segmentation algorithms the system is capable of online learning of objects by storing them using a geometric hashing (Rigoutsos & Wolfson, 1997) representation. Based on a similarity threshold, objects are separated into different classes using color and pairwise edge histograms. The discriminatory power, however, seems to be limited to a small number of objects and still strongly depends on color. What is more important is the integration of the online object learning into a model for tracking objects and learning task sequences and to recognize objects employed on such tasks from human–robot interaction.

An interesting approach to supervised online learning for object recognition was proposed by Bekel, Bax, Heidemann, and Ritter (2004). Their classification architecture consists of three major stages. The two feature-extraction stages are based on vector quantization and a local principal component analysis (PCA) measurement. The final stage is a supervised classifier using a local linear map architecture. The image acquisition of new object views is triggered by pointing gestures on a table, and is followed by a short training phase, which takes some minutes. The main drawback is the lack of an incremental learning mechanism to avoid the complete retraining of the architecture.

Online learning has also been investigated for robotics in domains of behavior and movement control. In this field the dimensionality of the representation space can be still quite large for robotic systems with many degrees of freedom although it does not reach the full complexity of visual input. For a full review, which is beyond the scope of this manuscript, see (Steil et al., 2006). As an important example that particularly focuses on incremental online learning we would like to mention Vijayakumar, D'Souza, and Schaal (2005), who propose a locally weighted projection regression (LWPR) algorithm, which is especially used for learning robot movements. The advantage of this method is the possibility to train complex robot movements online with only a few trials. The basic idea of the LWPR algorithm is to reduce the high number of possible input dimensions (up to 90 joints) to the essential ones necessary for the particular movement. The proposed method works well, if such a low-dimensional distribution in the input space exists.

2.2. Network architectures for incremental and life-long learning

One established neuronal network architecture that is able to learn online with the same performance as for offline training is the adaptive resonance theory (ART) and especially Fuzzy ARTMAP (Carpenter, Grossberg, Markuzon, Reynolds, & Rosen, 1992). The relation of this network architecture to our short-term memory model will be discussed later (see Section 3.2) in more detail. In recent years the ART network family was applied to several problems including recognition of handwritten digits (Carpenter, Grossberg, & Iizuka, 1992) and a sensorimotor anticipation architecture for robot navigation (Heinze, Gross, & Surmeli, 2001). An overview of several other ART-based applications can be found in (Carpenter & Grossberg, 1998).

Incremental radial basis function (RBF) networks (Fritzke, 1994a) and the growing neuronal gas (GNG) model (Fritzke, 1995) were suggested with a focus on incremental learning. Although it is possible to train these networks with a slowly changing training set, these architectures are mainly designed for offline training. Typically these networks cannot be trained on a limited training set without significantly losing generalization performance, because of a permanent increase in the number of neurons and the drift of nodes to capture the current training data (Hamker, 2001).

Furao and Hasegawa (2006) propose several improvements to the unsupervised version of the GNG and especially target the life-long learning of non-stationary data for problems like the clustering of faces or topology learning of images. They use a two-layered network, where the first layer is used to generate a topology structure of the input data and the second layer is used to determine the number of clusters. Furthermore they propose several utility estimation measurements for evaluating the insertion of nodes or to decide which nodes can be removed. Additionally they use an individual learning rate for each node, which improves the life-long learning capability strongly. A related approach was proposed by Hamker (2001), who introduced a neuronal network architecture for supervised learning, called life-long

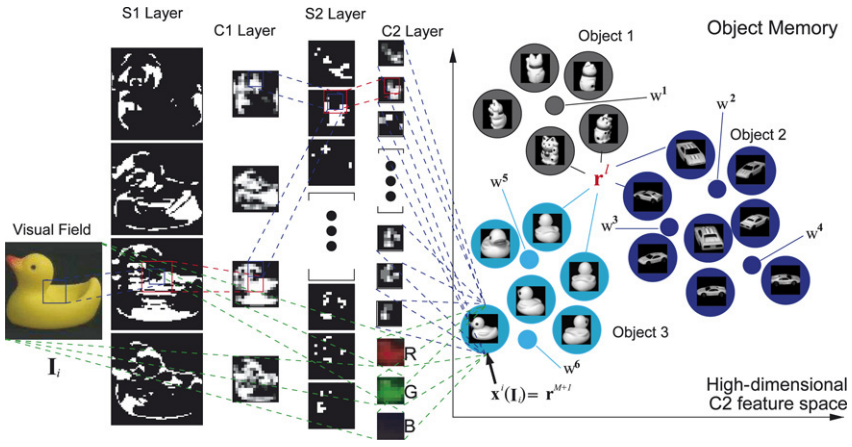

Fig. 1. The visual hierarchical network structure with feature processing, STM and LTM. Based on a color image input $\mathbf{I}_i$, shape and color processing is separated in the feature hierarchy and fused in the view-based object representation. The S1 layer performs a coarse local orientation estimation using Gabor filters, a winner-take-most mechanism and a final threshold function. The S1 features are pooled down to a quarter in each direction in layer C1. Neurons in the S2 layer are sensitive to local combinations of the features of the C1 layer. The C2 layer again reduces the resolution by a half in each direction. When the color pathway is used, three down-sampled maps of the individual RGB channels are added to the C2 feature maps, with the same resolution as one shape feature map. The short-term memory consists of template vectors $\mathbf{r}^l$ that are computed as the output $\mathbf{x}^l(\mathbf{I}_i)$ of the hierarchy and added based on sufficient Euclidean distance in the C2 feature space to previously stored representatives of the same object. The refined long-term memory representatives $\mathbf{w}^k$ are learned from the labeled short-term memory nodes $\mathbf{r}^l$ using an incremental vector quantization approach.

learning cell structures (LLCS). The LLCS networks are based on the growing cell structures (Fritzke, 1994b) and provide several extensions, like the calculation of an individual node learning rate, the definition of an insertion rule and the use of several measurements to detect useless nodes. The LLCS networks are also able to detect regions in low-dimensional data where points of different classes overlap. This avoids an unlimited insertion of neurons in those areas.

3. Hierarchical online learning model

Our incremental learning model consists of three major processing stages: First the input image is processed using a hierarchical and topographically ordered model of the ventral visual pathway for spatial feature extraction. The extracted feature maps of object views are then stored in a template-based short-term memory that allows online learning and immediate recognition. Finally the short-term memory representatives are accumulated into a condensed long-term memory using an incremental LVQ method. In the following section we describe these three processing stages in more detail.

3.1. Feed-forward feature-extracting hierarchy

Our hierarchical architecture (see Wersing and Körner (2003) for details) is based on a feed-forward architecture with weight-sharing and a succession of feature sensitive and pooling stages which is related to the Neocognitron proposed by Fukushima (1980). Fig. 1 shows an overview of this feature-extracting architecture and the object memory concepts composed of a STM and LTM model.

Starting point for the feature-extracting process are RGB color images $\mathbf{I}_i = (\mathbf{I}_i^R, \mathbf{I}_i^G, \mathbf{I}_i^B)$. The shape pathway is based on a gray-value intensity image $\mathbf{I}_i'$, obtained from the color image by weighted addition of the RGB channels:

$$\mathbf{I}_i' = \frac{1}{3}\mathbf{I}_i^R + \frac{1}{3}\mathbf{I}_i^G + \frac{1}{3}\mathbf{I}_i^B. \tag{1}$$

The first feature-matching layer S1 is composed of four orientation sensitive Gabor filters $\mathbf{w}_1^l(x, y)$ which perform a local orientation estimation. To compute the response $g_1^l(x, y)$ of a simple cell of this layer, responsive to feature type l at position (x, y) first the image vector $\mathbf{I}_i'$ is convolved with a Gabor filter $\mathbf{w}_1^l(x, y)$:

$$g_1^l(x, y) = |\mathbf{w}_1^l(x, y) * \mathbf{I}_i'|. \tag{2}$$

Additionally a winners-take-most (WTM) mechanism between features at the same position is performed:

$$h_1^l(x, y) = \begin{cases} 0 & \text{if } \dfrac{g_1^l(x, y)}{M} < \gamma_1 \text{ or } M = 0, \\ \dfrac{g_1^l(x, y) - M\gamma_1}{1 - \gamma_1} & \text{else,} \end{cases} \tag{3}$$

where $M = \max_k g_1^k(x, y)$ and $h_1^l(x, y)$ is the response after the WTM mechanism which suppresses submaximal responses. The parameter $0 < \gamma_1 < 1$ controls the strength of the competition. The activity is then passed through a simple threshold function with a common threshold θ_1 for all cells in layer S1:

$$s_1^l(x, y) = \mathrm{H}\left(h_1^l(x, y) - \theta_1\right), \tag{4}$$

where $\mathrm{H}(x) = 1$ if $x \geq 0$ and $\mathrm{H}(x) = 0$ else and $s_1^l(x, y)$ is the final activity of the neuron sensitive to feature l at position (x, y) in the S1 layer.

The C1 layer subsamples the S1 features by pooling down to a quarter in each direction (e.g. 64×64 S1 features are pooled

down to 16 × 16 C1 features):

$$c_1^l(x, y) = \tanh\left(\mathbf{p}_1(x, y) * \mathbf{s}_1^l\right), \tag{5}$$

where $\mathbf{p}_1(x, y)$ is a normalized Gaussian pooling kernel with width σ_1, identical for all features l, and tanh is the hyperbolic tangent function.

The S2 layer is sensitive to local combinations of the orientation estimation features extracted from layer C1. The so-called combination features of this S2 layer (for this experiments 50 different shape features l are used) are trained with sparse coding (see Wersing and Körner (2003) for details). The response $g_2^l(x, y)$ of one S2 cell is calculated in the following way:

$$g_2^l(x, y) = \sum_k \mathbf{w}_2^{lk}(x, y) * c_1^k, \tag{6}$$

where $\mathbf{w}_2^{lk}(x, y)$ is the receptive-field vector of the S2 cell of feature l at position (x, y), describing connections to the plane k of the previous C1 cells. Similar to the S1 layer a WTM mechanism (see Eq. (3)) and a final threshold function (see Eq. (4)) are performed in this S2 layer.

The following C2 layer again performs a spatial integration and reduces the resolution by half in each direction (i.e. 16×16 S2 features are down-sampled to 8×8 C2 features). The pooling is done with the same mechanism as in layer C1 (see Eq. (5)). When the color pathway is used, three additional down-sampled RGB maps are added to the shape feature channels. The 8×8 pixel resolution of each color channel is identical to one of the 50 shape features. Although an antagonistic (red–green, blue–yellow) color space is more biologically plausible, we assume that the performance difference to the RGB color space is only minor. This is due to the fact that color is only coarsely represented in our model and that the transformation between both color spaces is only linear, which should have only little influence on the Euclidean distance calculation which provides the basis for our STM and LTM model. Finally we denote the C2 output of the hierarchy for a given input image $\mathbf{I}_i$ as $\mathbf{x}^i(\mathbf{I}_i)$.

The output of the feature representation of the complex feature layer (C2) can be used for robust object recognition that is competitive with other state-of-the-art models (Wersing & Körner, 2003). The main property of C2 activations is their very high dimensionality combined with sparsity. Normally only about one-third of all C2 features are active for a given input stimulus, but this sparsity allows an efficient handling of the high-dimensional vectors (the actual dimensionality is $8 \times 8 \times (50 + 3) = 3392$). Another property of the data used in our scenario is related to the rotation of objects around three axes freely by hand. This free rotation of objects causes strong appearance variation of a single object class and also causes strong fluctuations in the extracted C2 feature vectors. In contrast to these strong intra-class variations of objects, feature vectors of different objects can be located relatively close together in this high-dimensional space, because similar object poses of related objects (e.g. different cups) can result in distinguishable but quite similar feature vectors.

In the following C2 activation maps are used to build up a template-based short-term memory which selects relevant representatives $\mathbf{r}^l$ and therefore reduces the training time of the long-term memory which is afterwards trained on the selected representatives.

3.2. Online vector quantization as short-term memory

The online vector quantization model provides fast appearance-based learning of three-dimensional objects, which can immediately be recognized. The proposed model stores template-based representatives $\mathbf{r}^l$ in a so-called short-term memory. The number of representatives $\mathbf{r}^l$ for a specific object is related to the complexity of the object and not specified beforehand. The learning process is based on the similarity to already stored representatives $\mathbf{r}^l$ of the same object. Therefore this online vector quantization model reduces the number of representatives $\mathbf{r}^l$ in contrast to a naive approach where every view $\mathbf{x}^i(\mathbf{I}_i)$ is stored in memory. Especially already seen views or very similar views are not collected into the short-term memory.

The labeled object views are stored in a set of M representatives $\mathbf{r}^l$, $l = 1, \ldots, M$, that are incrementally collected, and labeled with class Q^l. We define R_q as the set of representatives $\mathbf{r}^l$ that belong to object q. The acquisition of templates is based on a similarity threshold S_T. New views of an object are only collected into the short-term memory (STM) representation if their similarity to the previously stored views is less than S_T. The parameter S_T is critical, characterizing the compromise between representation resolution and computation time needed for one training or validation step. We denote the similarity of view $\mathbf{x}^i$ and representative $\mathbf{r}^l$ by A_{il} and compute it based on the previous calculated C2 feature map in the following way:

$$A_{il} = \exp\left(-\frac{\|\mathbf{x}^i - \mathbf{r}^l\|^2}{\sigma}\right). \tag{7}$$

Here, σ is chosen for convenience such that the average similarity in a generic recognition setup is approximately equal to 0.5. We use the exponential function just to obtain an intuitive notion of similarity, any other monotonous transformation of the Euclidean distance would also be possible.

For one learning step the similarity A_{il} between the current training vector $\mathbf{x}^i$, labeled as object q and all representatives $\mathbf{r}^l \in R_q$ of the same object q is calculated and the maximum value is computed as:

$$A_i^{\max} = \max_{l \in R_q} A_{il}. \tag{8}$$

The training vector $\mathbf{x}^i$ with its class label is added to the object representation, if $A_i^{\max} < S_T$. If M representatives were presented before, then choose $\mathbf{r}^{M+1} = \mathbf{x}^i$ and $Q^{M+1} = q$. Otherwise we assume that the vector $\mathbf{x}^i$ is already sufficiently well represented by one $\mathbf{r}^l$, and do not add it to the representation. We call this basic template-based representation online vector quantization (oVQ). Due to the non-destructive incremental learning process, online learning and recognition

can be done at the same time, without a separation into training and testing phases. To model a limited STM capacity, in the simulations an upper limit can be set on the number of objects that can be represented. This means that, when too many objects are presented, representatives belonging to the oldest learned object are removed from the STM.

For the online recognition of a new and unclassified test view $\mathbf{I}_j$ we first process the object view through the feature-extracting hierarchy. The C2 output of this hierarchy $\mathbf{x}^j(\mathbf{I}_j)$ is then used for a nearest neighbor search to the set of all representatives stored in the short-term memory. The nearest neighbor search selects the best matching node $\mathbf{r}^{l_{\max}}$, where $l_{\max}$ satisfies:

$$l_{\max} = \arg\max_l (A_{jl}). \tag{9}$$

The class label $Q^{l_{\max}}$ of the winning representative $\mathbf{r}^{l_{\max}}$ is then assigned to the current unclassified test view $\mathbf{x}^j$.

The oVQ algorithm can handle the used high-dimensional C2 data (see Section 3.1 for details) in an efficient way. It is especially suited for the sparse C2 feature vectors, which allows us to store ten thousands of representatives, while keeping the ability to train and validate new occurring feature vectors online. The similarity threshold S_T, the only critical parameter in our STM model, controls the tradeoff between a more detailed and exhaustive object view sampling and the amount of representatives in the STM.

Based on the description of our oVQ algorithm the relation to Fuzzy ARTMAP (Carpenter et al., 1992) and Fuzzy ART (Carpenter et al., 1991) will be discussed in the following. Both architectures have the common feature that they can immediately recognize a specific object view after a single occurrence ("one-shot learning"), which makes them suitable for online learning. It is also possible to incrementally add new objects without destroying already learned capabilities and the learning process in both algorithms is based on a similarity condition called vigilance ρ for ART networks.

Due to the special properties of the C2 feature vectors in our object recognition scenario the more complex Fuzzy ART network family is not suitable for storing a large amount of freely rotated objects. One major drawback of Fuzzy ART is related to the sparsity of feature vectors, which essentially requires complement coding to avoid that too many adaptive weights become zero. A large amount of zero weights is an unattractive condition for Fuzzy ART networks that should be prevented (Carpenter et al., 1992), because in such a case the "choice function" used for calculating the winner node always results in nearly perfect matches, which results in choosing a winner node independent of the input. Additionally the already very high-dimensional feature vectors are doubled in size by this coding schema. Based on the complement coding and the vigilance parameter ρ, input vectors are assigned to hypercubes around the representative vectors with the size inversely proportional to ρ. This vigilance parameter ρ is, similar to the S_T in our model, a critical parameter and ρ should also be chosen as small as possible, to avoid the allocation of an enormous amount of resources. On the contrary, small vigilance parameters ($\rho < 0.9$) cause other problems, because it allows the creation of large hypercubes during the learning process. This leads to the undesired convergence of many adaptive weights to zero as the consequence of strong intra-class variations of the sparse feature vectors described in Section 3.1. These strong intra-class variations together with relatively closely located vectors of related objects in similar poses will most probably result in many partially overlapping hypercubes. If such hypercubes are belonging to different classes and validation vectors are located in these areas, then the generalization ability of the network will be reduced, because the "choice function" results in the same optimal value for all nodes involved in this overlapping and the selection of the winner is dependent on the search order.

3.3. Incremental LVQ as long-term memory

Our STM model provides fast learning and achieves good recognition performance, as we will demonstrate in the results section. Nevertheless the large amount of memory for storing the high-dimensional C2 feature vectors of all objects is the main disadvantage and is also biologically not plausible. Therefore we propose a transfer from the STM into the LTM, inspired by the transfer from medial temporal lobe into the neocortex in biological vision. To build up such a LTM model we use an incremental LVQ algorithm (iLVQ). This network architecture described in the following section should strongly reduce the representational effort of objects without reducing the generalization performance of the recognition system. Additionally the LTM model is approaching the life-long learning problem, which allows learning of objects during the complete history of the iLVQ network.

The labeled STM representatives $\mathbf{r}^l$ in the C2 feature space provide the input ensemble for our proposed long-term memory (LTM) representation, which is optimized and built up incrementally. The main reason for training the long-term memory based on the collected STM representatives $\mathbf{r}^l$ is that the STM already rejects very similar object views and reduces the number of training views for the long-term memory. This reduction causes an advanced training time in contrast to the case where every input view is used. Additionally we assume a limited STM capacity with only the most recently shown objects being represented. Therefore an algorithm is needed that is able to incrementally add new objects or even refine object representations without destroying already learned object representations, thereby taking into account the stability-plasticity dilemma.

The learning vector quantization (LVQ) networks proposed by Kohonen (1989) are a well-known neuronal network architecture for supervised learning. The single-layered LVQ networks are typically trained with a fixed number of nodes; therefore the number of nodes for each class must be selected before the training phase starts. It is quite difficult to accurately determine the necessary number of nodes for a particular class. If the number of nodes is too large convergence is slow, whereas a too low number only provides a poor generalization performance of the network. Additionally the number of

necessary nodes is also related to the complexity of a particular class itself. To take care of this fact a lot of a priori knowledge must be available to select an appropriate number of LVQ nodes. To avoid this problem we use an incremental approach for the LTM model, which is able to automatically determine the necessary number of nodes, based on the complexity of the object and the difficulty of the learning task. We also extend the basic LVQ networks with respect to the stability-plasticity dilemma of life-long learning. All extensions of the basic LVQ network architecture will be described in the following.

For the training of our incremental LVQ (iLVQ) network, a stream of randomly selected input STM training vectors $\mathbf{r}^l$ is presented and classified using the labeled iLVQ representatives in a Euclidean metric. The training classification errors are collected, and each time a given sufficient number of classification errors has occurred, a set of new iLVQ nodes will be inserted. The addition rule is designed to promote insertion of nodes at the class boundaries. During training, iLVQ nodes are adapted with standard LVQ weight learning that move nodes into the direction of the correct class and away from wrong classes. An important change to the standard LVQ method is an adaptive modification of the individual node learning rates to deal with the stability-plasticity dilemma of incremental learning. The learning rate of winning nodes is more and more reduced to avoid too strong interference of newly learned representatives $\mathbf{r}^l$ with older parts of the object long-term memory.

We denote the set of iLVQ representative vectors at time step t by $\mathbf{w}^k(t)$, $k = 1, \ldots, K$, where K is the current number of nodes. C^k denotes the corresponding class label of the iLVQ center $\mathbf{w}^k$. The training of the iLVQ nodes is based on the current set of STM nodes $\mathbf{r}^l$ with class Q^l that serve as input vectors for the LTM. Each iLVQ node $\mathbf{w}^k$ obtains an individual learning rate:

$$\Theta_k(t) = \Theta(0) \exp\left(-\frac{a_k(t)}{d}\right) \tag{10}$$

at step t, where $\Theta(0)$ is an initial value, d is a fixed scaling factor, and $a_k(t)$ is an iteration-dependent age factor. The age factor a_k is incremented every time when the corresponding $\mathbf{w}^k$ becomes the winning node.

New iLVQ nodes are inserted, if a given number $G_{\max}$ of training vectors are misclassified during iterative presentation of the $\mathbf{r}^l$. We choose a value of $G_{\max} = 30$, since a high $G_{\max}$ value guarantees an optimal representation of objects with a minimal number of LVQ nodes, but also slows down the convergence speed of this learning algorithm. Within this error history, misclassifications are memorized with input $\mathbf{r}^l$ and the corresponding winning iLVQ node $\mathbf{w}^{k_{\max}}(\mathbf{r}^l)$. We denote S_p as the set of previously misclassified $\mathbf{r}^l$ within this error history that were of original class $p = Q^l$. For each non-empty S_p a new node $\mathbf{w}^m$ is added to the representation, independent of the number of entries in S_p. This insertion technique limits the insertion of nodes, if many views of a particular class are wrongly classified. The iLVQ insertion rule is illustrated in Fig. 2. New neurons are initialized to the element of $\mathbf{r}^l \in S_p$ that has the minimal distance to its corresponding but wrong

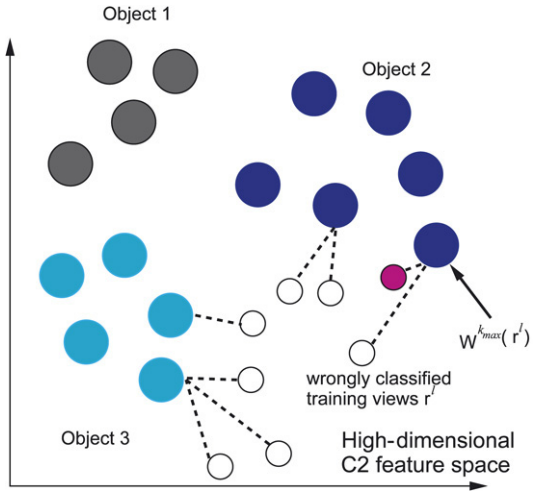

Fig. 2. Illustration of iLVQ node insertion rule. Wrongly classified training views $\mathbf{r}^l$ of class p are collected into S_p, which contains all wrongly classified views of the given class p. These views $\mathbf{r}^l \in S_p$ are shown with small circles, whereas the iLVQ nodes are shown as large filled circles. Additionally the distance of the $\mathbf{r}^l$ to their corresponding but wrong winning iLVQ node is shown (dashed lines). The insertion rule determines the wrongly classified $\mathbf{r}^l$ with minimal distance to the iLVQ node $\mathbf{w}^{k_{\max}}(\mathbf{r}^l)$. This training view (the small filled circle) is then used for initializing a new iLVQ node with class label $C^m = p = Q^l$ of the training view.

winning iLVQ node $\mathbf{w}^{k_{\max}}(\mathbf{r}^l)$ and the class of the iLVQ node is given as $C^m = Q^l$. This rule adds new nodes primarily near to class borders, where typically the most classification errors occur. This node insertion rule can be related to boundary classifiers like support vector machines (see Burges (1998) for an introduction to SVM), where so-called support vectors at the classification border are selected to form the decision boundary. In contrast to this the iLVQ algorithm forms Voronoi clusters, where the cluster centers can be quite far apart from the classification border.

A test view $\mathbf{x}^j(\mathbf{I}_j)$ is classified by determining the winning iLVQ node $\mathbf{w}^{k_{\max}}$ with smallest distance to the current C2 feature vector $\mathbf{x}^j$ and assigning the corresponding label $C^{k_{\max}}$ as the output class.

The formal definition of the iLVQ learning algorithm will be described in the following:

(1) Choose randomly a representative $\mathbf{r}^l$ from the set of current STM nodes. Calculate the Euclidean distance between the $\mathbf{r}^l$ and all iLVQ nodes $\mathbf{w}^k$ and select the winning node with minimal distance to the $\mathbf{r}^l$:

$$k_{\max} = \arg\max_k(-\|\mathbf{r}^l - \mathbf{w}^k\|). \tag{11}$$

After this selection process the winning node $\mathbf{w}^{k_{\max}}$ is adapted using the common LVQ learning rule:

$$\mathbf{w}^{k_{\max}}(t+1) = \mathbf{w}^{k_{\max}}(t) + \kappa\,\Theta_{k_{\max}}(t)(\mathbf{r}^l - \mathbf{w}^{k_{\max}}(t)), \tag{12}$$

where $\kappa = 1$ if the class label Q^l of the representative $\mathbf{r}^l$ and the class label $C^{k_{\max}}$ of the winning node $\mathbf{w}^{k_{\max}}$ are identical, otherwise $\kappa = -1$ and the winning node will be shifted into the opposite direction as the input

representative $\mathbf{r}^l$. The learning rate for the winning node $\mathbf{w}^{k_{\max}}$ at time step t is calculated according to Eq. (10).

(2) After the adaptation of the winning node $\mathbf{w}^{k_{\max}}$ the age factor $a_{k_{\max}}$ of this node will be incremented:

$$a_{k_{\max}}(t+1) = a_{k_{\max}}(t) + 1. \tag{13}$$

This increment of $a_{k_{\max}}$ results in a slightly smaller learning rate if the $\mathbf{w}^{k_{\max}}$ iLVQ node becomes in a further training step again the winning node.

(3) If the current representative $\mathbf{r}^l$ was misclassified ($C^{k_{\max}} \neq Q^l$), then G will be increased ($G(t+1) = G(t)+1$) and $\mathbf{r}^l$ will be added to the current set of misclassified views S_{Q^l} of object Q^l.

(4) Every training step it will be checked if $G = G_{\max}$, if so we insert for each $S_p \neq \emptyset$ a new iLVQ node. If more than one representative $\mathbf{r}^l$ of class $p = Q^l$ was wrongly classified, it must be decided which $\mathbf{r}^l$ is used to initialize the new iLVQ node. For the initialization of the new iLVQ node of class $C^m = p$ we determine the index of the iLVQ representative $l_{\min}$ with minimal distance to the wrongly classified elements in S_p according to:

$$l_{\min} = \arg\min_{l|\mathbf{r}^l \in S_p} \|\mathbf{r}^l - \mathbf{w}^{k_{\max}}(\mathbf{r}^l)\|, \tag{14}$$

where $\mathbf{w}^{k_{\max}}(\mathbf{r}^l)$ is the winning iLVQ node for view $\mathbf{r}^l$. Insert a new iLVQ node with $\mathbf{w}^{K+1} = \mathbf{r}^{l_{\min}}$. Reset $G = 0$ and $S_p = \emptyset$ for all p.

(5) Start a new training step (goto step 1) until sufficient convergence is reached.

Our proposed LTM model defines several extensions to the LVQ network architecture, which are necessary to fulfill the given incremental and life-long learning object recognition task. Especially the definition of an individual node learning rate or the definition of a node insertion rule are methods also used by Hamker (2001) and Furao and Hasegawa (2006). They propose node insertion based on accumulated errors of each individual node, whereas we only observe the wrong classification itself. If some classification errors occur, nodes are inserted for every wrongly classified object class. Also the initialization of the new nodes differs, we add nodes near class borders but based on a wrongly classified training vector, whereas Hamker and Furao & Hasegawa insert a new node in the neighborhood of an already existing node, for which activation does not occur necessarily. On the contrary, this slows down the learning algorithm, because such a node may not contribute to the representation. Based on the proposed node deletion criteria of both authors the detection of such useless nodes requires several training steps.

4. Experimental results

In the following we describe experiments on using the coupled STM and LTM architecture in a recognition scenario for freely rotated objects. We describe the resulting image ensemble which is shown in Fig. 4 and specify how we do the preprocessing for segmenting the objects.

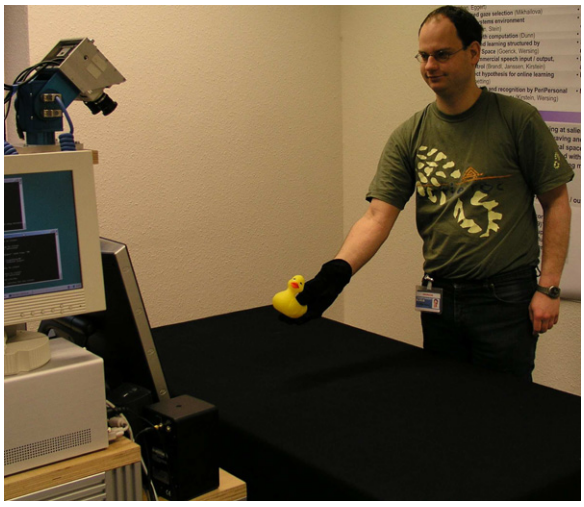

Fig. 3. Experimental setup. Objects are rotated freely by hand in front of a camera. Additionally we use a black glove and show the objects in front of a black table to simplify the foreground–background separation, which is not the focus of this contribution. Using our short-term memory model the recognition system can be trained online to recognize 50 different objects.

4.1. Experimental setup

For our experiments we use a setup, where we show objects, held in hand and freely rotate them around three axes (see Fig. 3). To ease figure-ground segmentation we use a black glove and rotate the objects in front of a black background. The color images are taken with a camera and are segmented with a simple local entropy-thresholding (Kalinke & von Seelen, 1996) method. Recently, a larger integrated system was developed that could relax the strong constraints on the background using more advanced segmentation methods (Wersing et al., 2006).

After the segmentation of the object view we normalize it in size (64 × 64 pixels). For collecting the database we rotated every object freely by hand for some minutes, such that 750 input images $\mathbf{I}_i$ for each object are collected. Another independently taken set of 750 images for each of the objects is recorded as validation database. Fig. 4(a) shows all 50 different objects of our HRI50 database. The difficulty of this database results from the high-appearance variation of objects during rotation around three axes. The database also contains a lot of objects which are similar in shape or color, e.g. the different cups, boxes or cans. Some rotation examples for different objects are shown in Fig. 4(b). Additionally some segmentation errors and minor occlusion effects are shown in Fig. 4(c) and Fig. 4(d).

4.2. Online vector quantization as short-term memory

In the first experiment we investigate the time necessary for training the template-based oVQ short-term memory with up to 50 objects, and evaluate the recognition performance. The training speed is limited by the frame rate of the used camera (12.5 Hz) and the computation time needed for the entropy segmentation, the extraction of the corresponding sparse C2 feature vector $\mathbf{x}^i$ with 3200 shape dimensions and

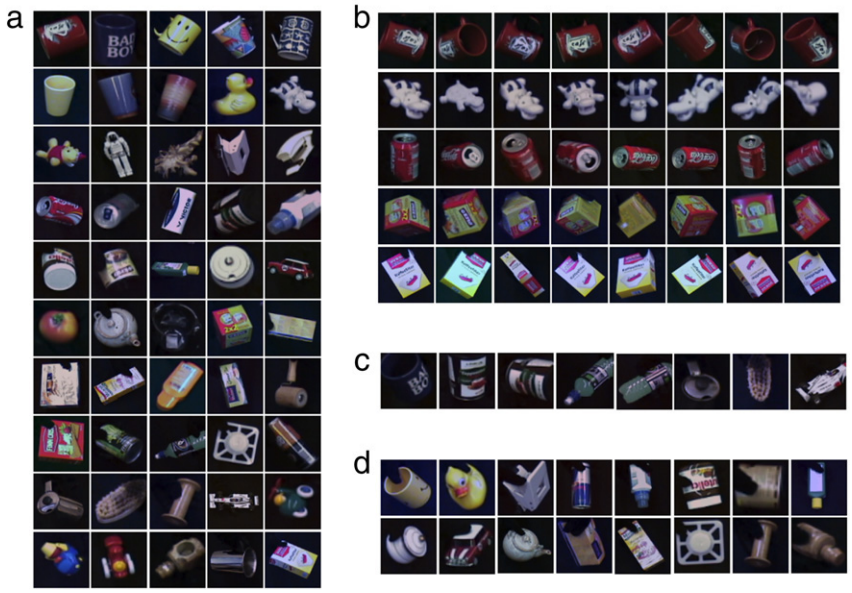

Fig. 4. Example object images of the HRI50 database. (a) 50 freely rotated objects, taken in front of a dark background and using a black glove for holding. (b) Some rotation examples. (c) A few examples for incomplete segmentation. (d) Examples for minor occlusion effects. The main difficulties of this training ensemble are the high-appearance variation of objects during rotation around three axes, and shape similarity among cans, cups and boxes, combined with segmentation errors (c), and slight occlusions (d).

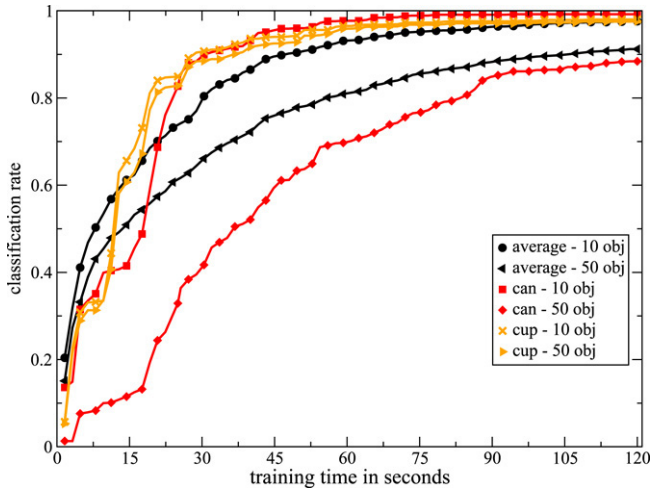

Fig. 5. Classification rate of two selected objects dependent on the training time for learning the 10th and 50th object, and same learning curves averaged over 20 object selections. While training proceeds, at each point the classification rate is measured on all 750 available test views of the current object. Good recognition performance can be achieved within two minutes, also for the 50th object.

192 color dimensions, and the calculation of similarities A_{il} (see Section 3). The similarity threshold was set to $S_T = 0.85$ for this experiment, and there was no limit imposed on the number of STM representatives. Altogether we achieve an average frame rate of 7 Hz on a 3 GHz Xeon processor. Fig. 5 shows how long it takes until a newly added object can be robustly separated from all other objects. For the shown curves of a cup and a can from our database we trained 9 or 49 objects and incrementally added the cup or can as an additional object. At the given points the correct classification rate of the current object is computed using the 750 views from the disjoint test ensemble. Additionally we show the learning curves, averaged over 20 randomly chosen object selections. On an average, training of one object can be done in less than 2 min, with rapid convergence.

To evaluate the quality of the feature representation obtained from the visual hierarchy, we performed a systematic comparison of the use of three different types of feature vectors. The first kind of feature vectors contains only shape information of the objects and has $8\times8\times50$ dimensions (8×8 activations for each of the 50 extracted shape feature maps). The second type of vectors with a dimension of $8 \times 8 \times (50 + 3)$ C2 features contains shape and additional coarse color information. Finally we used plain $64 \times 64 \times 3$ pixel RGB images as input vectors $\mathbf{x}_i$ for the oVQ model. Due to the high dimensionality and lack of sparsity we can only represent up to 17.000 representatives in this case. This plain image setting also captures the baseline similarity of this ensemble, and can serve as a reference point, since there are currently no other established standard methods for online learning available. Additionally we varied the similarity threshold S_T to investigate the tradeoff between representation accuracy and classification errors. The results are shown in Fig. 6. Each symbol of the STM graphs in Fig. 6 corresponds to a particular threshold S_T. For a given S_T we let our short-term memory model decide, which training vectors are necessary and calculate the classification rate based on the selected representatives. For a fair comparison, error rates for roughly equal numbers of chosen representatives should be compared. Using the hierarchical shape features reduces the error rates considerably, compared to the plain color images, especially for small number of representatives. The addition of the three coarse RGB feature maps additionally reduces error rates by about one-third. For a complete training of all 50 objects with a real camera, accomplished within about three

Table 1
Comparison of results achieved with the iLVQ, SLP, and SNOW approach using the COIL-100 database

	Shape			Shape and coarse color		
	All	STM	lim. STM	All	STM	lim. STM
iLVQ	98.6%	97.9%	96.0%	99.5%	99.3%	98.4%
SLP	99.9%	99.5%	28.0%	99.9%	99.8%	27.6%
SNOW	96.5%	94.2%	59.2%	97.6%	96.7%	50.0%

Classification rates of all three approaches are shown based on C2 shape features and the combination of shape and coarse color features. Additionally we compare the results using all available training data, the use of the proposed with STM $S_T = 0.9$ and a limited STM.

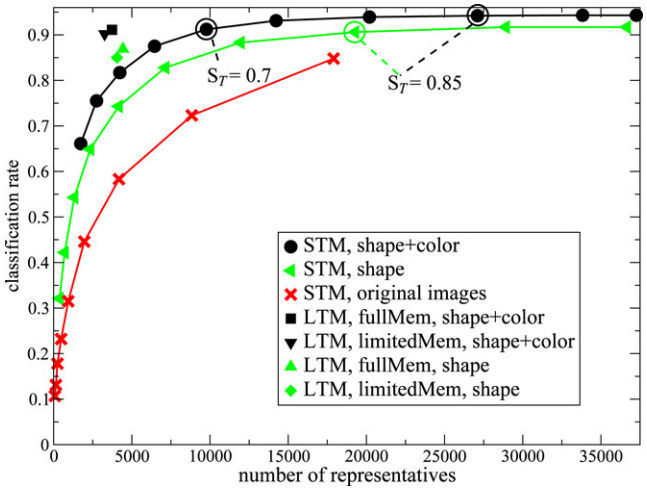

Fig. 6. Comparison of classification rates of the STM and LTM model for the HRI50 database. Classification rates of the STM model are calculated for different similarity thresholds S_T and different types of C2 feature maps, whereas the LTM model was trained with limited or unlimited STM using shape and coarse color information. It can be seen that the use of the visual hierarchy shape features reduces the error rate, compared to the plain color images. The additional use of coarse color features again reduces the error rates of the STM model considerably. For the LTM model tests a similarity threshold of $S_T = 0.85$ was used for training the STM model, where its representatives $\mathbf{r}^j$ serve as input for the LTM. It can be seen that the LTM model reduces the required resources from about 27 000 STM representatives to less than 3800, with a slightly reduced classification performance. Further it should be mentioned that the iLVQ reaches nearly the same classification performance for the limited STM compared to the unlimited case.

hours, the remaining classification error is about 6% using color and shape features and 8% using only shape information.

4.3. Incremental LVQ as long-term memory

The performance of the proposed iLVQ long-term memory model is shown in Fig. 6 in relation to the results obtained from the STM model. We compare the effect of using only a limited STM memory history for the transfer into the LTM representation, compared to the usage of unlimited STM. For the experiments with the iLVQ networks we used a similarity threshold $S_T = 0.85$ for the STM model and applied this threshold to the STM training with shape features and also combined shape and coarse color features. This threshold was chosen as a compromise between the resulting generalization performance for both feature representations and the number of selected STM representatives.

With our LTM model we are able to strongly reduce the necessary number of representatives from about 27 000 STM representatives to less than 3800 LTM iLVQ nodes using shape and color features. However this is achieved at the price of a slightly reduced performance of 91.1% correct classification, compared to the performances of the STM representatives which reaches a classification performance of 94.2% at the given value of S_T. If we compare the STM setting, where the classification rate matches approximately 91%, which corresponds to a lower similarity threshold of $S_T = 0.7$, the number of representatives is still three times larger than for the LTM, as can be seen from Fig. 6.

For a better comparison of our LTM model to other state-of-the-art approaches, experiments with the well-known COIL-100 database (Nayar, Nene, & Murase, 1996) are performed. This database consists of 100 different objects rotated around one axis, where the 72 different views for each object are taken at pose intervals of 5°. For our experiments we resized the original images to 64 × 64 pixels to allow a better comparison to our own HRI50 database. For all experiments with the COIL-100 database, 36 object views (10° apart) are used for training and the remaining views for testing.

Additionally we compared our architecture to a one-layered sigmoidal network and the SNOW (Roth, Yang, & Ahuja, 2002) approach. The sigmoidal network consists of an input and output layer, without hidden layers. For every object we used one output node, whereas each node has a linear scalar product activation and a sigmoidal transfer function. The SNOW approach is especially designed for a sparse feature representation as used in our experiments. It is also better suited for incremental and life-long learning due to its conservative learning schema. The SNOW model is based on a multiplicative Winnow update rule (Littlestone, 1988), which is applied to wrongly classified training vectors only. Furthermore exclusively the weights of currently activated input dimensions are modified at a training step which theoretically provides more life-long learning stability than sigmoidal networks where typically all weights are updated at each learning step. For SNOW we used the same network size as for the sigmoidal networks, i.e. one output node for each object.

For the comparison of the iLVQ, SLP and SNOW approach we performed a systematic analysis using all available training data of the used image ensemble, compared to the use of the proposed STM model and a limited STM, where only the recent 10 objects are available for training. Furthermore we compare the results achieved with two different feature ensembles based

Table 2
Comparison of results achieved with the iLVQ, SLP, and SNOW approach using the HRI50 database

	Shape			Shape and coarse color		
	All	STM	lim. STM	All	STM	lim. STM
iLVQ	88.5%	86.9%	85.8%	91.6%	91.1%	90.2%
SLP	84.1%	80.7%	21.9%	91.2%	91.1%	21.7%
SNOW	52.8%	51.9%	20.3%	55.6%	54.2%	20.7%

Classification rates of all three approaches are shown based on C2 shape features and the combination of shape and coarse color features. Additionally we compare the results using all available training data, the use of the proposed STM with $S_T = 0.85$ and a limited STM.

on the C2 shape features and the use of additional coarse color features. The results of this comparison are shown in Table 1 for the COIL-100 database and Table 2 for the HRI50 database.

For the COIL-100 database (see Table 1) it can be seen that the single layer perceptron achieves better classification results as our proposed iLVQ method for the cases where no limit on the training data was imposed. The SNOW network is slightly worse than iLVQ and SLP, but the classification rate is still comparable to other state-of-the-art approaches applied to this database. It should be noted that the performance we achieved with our C2 shape features representation is superior to the results published by Roth et al. (2002) (one-against-all: 90.52%), which highlights the quality of the hierarchical feature representation. For all three models, the introduction of the STM model with approximately 30% reduction of training data causes only minor increase in errors. For the experiments using only a limited STM of 10 objects, it can be seen that only the iLVQ method can handle this with almost no performance loss. Although the performance decrease of the SNOW approach is distinctly less than for SLP, both methods quickly fail to distinguish objects from earlier training phases, resulting in low-recognition rates. This is the well-known catastrophic forgetting effect (Hamker, 2001).

The results obtained with the HRI50 database are shown in Table 2. In comparison to the COIL-100 results the iLVQ method achieves better results on this more difficult database than the SLP approach, which is most distinct for the use of shape features only. This better performance is mainly caused by the incremental learning of the iLVQ approach allowing an adaptation to the difficulty of the classification task, while the SLP approach does not allow incremental learning. It can also be seen that the SNOW approach cannot capture the higher-appearance variation of the HRI50 database, which results in poor classification performance. For the training with the limited STM the iLVQ also achieves good results on the HRI50 database. In contrast to the COIL-100 database the SNOW approach is also worse than SLP for the limited STM experiments, which is mainly due to the overall poor performance of SNOW on the HRI50 database.

5. Discussion

We have proposed a biologically motivated approach for the learning of visual object representations. It is based on a hierarchical feature-extraction model serving as the input for a coupled short-term and long-term memory. Our main focus was to demonstrate the capability of online learning of many complex-shaped objects in combination with a model for a consolidation of fast but limited short-term memory into a condensed long-term memory representation. In the following we discuss the components of our model with reference to related work.

Our feature-detection approach is different from most of the related work on online learning for object recognition (Garcia et al., 2000; Steels & Kaplan, 2001; Roy & Pentland, 2002; Arsenio, 2004; Bekel et al., 2004), because the representation is not based on a dimension reduction of the high-dimensional visual input. Due to the receptive-field-based topographical representation, we obtain multiple shape feature-map representations with a resulting dimensionality that is of the same order as the visual input. Within the maps, however, only sparse activation is present, which is caused by the coding strategy in the hierarchical network.

The short-term memory model is defined as a template-based representation that adds new object representatives using a Euclidean metrics within the high-dimensional space of shape and color feature-map responses. Due to the purely incremental nature of this learning method we can perform online learning of objects by capturing sufficient appearance variation of the object under investigation. Adaptive resonance (ART) networks are another common approach to perform one-shot and online learning. Many applications of ART and its relative Fuzzy ARTMAP have so far concentrated on representation spaces with much lower dimensionality (Carpenter et al., 1992). The necessity of complement coding (see discussion in Section 3.2), doubling the input space dimensionality, and problems with sparse vectors make ART networks not very suitable for representing the feature activations of the visual hierarchy we use here.

For the application to online learning, using only the STM model achieved good generalization in combination with a large storage capacity of 50 objects, compared to other work on online learning of objects which usually did not consider more than 10–12 objects (Bekel et al., 2004; Arsenio, 2004). This capacity is a direct consequence of the high-dimensional representation space, and is also achieved if only shape representations are used. The STM model enables learning in direct interaction with a human teacher, whereas the long training time of most current recognition architectures does not allow this user interaction. However, the representational effort of storing a large number of high-dimensional feature maps can be large. To overcome this limitation we introduced a long-term memory model.

Our long-term memory model has to satisfy the two main requirements: It has to incrementally add and consolidate representational resources dependent on the complexity of the objects to be learned and care for the stability-plasticity dilemma caused by using only a limited STM memory of the previous object presentations. Due to the problems of standard architectures like MLPs, which suffer from catastrophic forgetting in such a scenario, most previous work on online object learning does not consider incremental learning, but rather collects the training data and then performs a standard batch learning procedure (Bekel et al., 2004).

As a demonstration of the catastrophic forgetting effect we performed experiments with the SLP and SNOW approach and could show a strong degradation of classification performance for our desired interactive and life-long learning task. Additionally we performed experiments with the COIL-100 database for a better evaluation of our HRI50 image ensemble. We could show that the LTM model can reach state-of-the-art recognition performance for the COIL-100 database. In direct comparison the HRI50 image ensemble is more challenging due to distinctly less classification rates. The difficulty of HRI50 database is caused by object rotation around three axes, whereas the COIL-100 objects are only rotated around one axis. This results in much higher-appearance variations which pose problems for the SNOW approach, while the iLVQ approach automatically scales to the difficulty of the recognition tasks resulting in good recognition rates for more challenging databases.

We have based our LTM architecture on a learning vector quantization (LVQ) model, which we have extended by methods of incremental node insertion, and flexible adaptation of the local node learning rates. Our approach can be compared to recent work on life-long learning for incremental neural architectures (Hamker, 2001; Furao & Hasegawa, 2006), targeting learning for non-stationary distributions without destruction of previously learned representations (see Section 3.3). Our iLVQ algorithm differs from the work of Hamker and Furao & Hasegawa mainly in the used node insertion rule. We insert neurons only if classification errors during the training phase occur and do not utilize the accumulated error of the nodes themselves. We assume that this leads to a smaller number of allocated resources compared to the distance-based insertion mechanism, especially in high-dimensional spaces. Hamker has demonstrated the efficiency of his proposed LLCS networks based on several low-dimensional non-stationary benchmark datasets. How this network architecture performs on more realistic problems with high-dimensional input spaces can, however, only be speculated until now. Furao and Hasegawa (2006) applied the proposed method to a setting of face clustering, but it seems to be that the unsupervised learning method is not efficient in high-dimensional input spaces with strong variation, which may be the reason for the use of smoothed input images in their experiments.

Hamker and Furao & Hasegawa propose utility measurements to detect rarely activated nodes or to decide if the insertion of a node was ineffective and does not cause a decreasing error rate. The drawback of the proposed methods is that they tend to delete nodes representing rarely occurring data with only very few feature vectors, which are typically quite important in our scenario where the objects are rotated freely by hand. Especially the LLCS (Hamker, 2001) utility measurements delete nodes which are not supported by other nodes in their direct neighborhood. The deletion of such nodes slows down the learning process and can also destroy parts of the representation which infrequently occur again. Although we did not care for an explicit node deletion procedure in our iLVQ model, we think that similar mechanisms of utility measurements could be advantageous for reducing the representational effort in the LTM model.

6. Conclusion

By using principles of hierarchical visual processing in the ventral visual pathway and a functional separation of short-term and long-term memory we have developed a model for visual online learning of several complex-shaped objects. To the best of our knowledge this neural model is the first approach capable of real incremental online learning and recognition of large numbers of complex-shaped objects, carried out in real-time with human interaction. Such a capability is highly relevant in the contexts of man–machine interaction and humanoid robotics, introducing many new possibilities for interaction and learning scenarios for incrementally increasing the visual knowledge of an assistive robot.

Acknowledgments

We thank C. Goerick, M. Dunn, J. Eggert and A. Ceravola for providing the image acquisition and processing system infrastructure.

References

Arsenio, A. M. (2004). Developmental learning on a humanoid robot. In: *Proc. international joint conference on neuronal networks* (pp. 3167–3172).

Bekel, H., Bax, I., Heidemann, G., & Ritter, H. (2004). Adaptive computer vision: Online learning for object recognition. In: *Proc. DAGM-symposium* (pp. 447–453).

Biederman, I. (1987). Recognition by components — a theory of human image understanding. *Psychological Review*, *94*, 115–147.

Broadbent, N. J., Squire, L. R., & Clark, R. E. (2004). Spatial memory, recognition memory, and the hippocampus. In: *Proceedings of the National Academy of Sciences, USA, 101* (pp. 14515–14520).

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*(2), 121–167.

Buzsáki, G. (1996). The hippocampo-neocortical dialogue. *Cerebral Cortex*, *6*(2), 81–92.

Carpenter, G. A., & Grossberg, S. (1998). Adaptive resonance theory (ART), In: *The handbook of brain theory and neural networks* (pp. 79–82).

Carpenter, G. A., Grossberg, S., & Iizuka, K. (1992). Comparative performance measures of Fuzzy ARTMAP, learned vector quantisation, and back propagation for handwritten character recognition. In: *Proc. IJCNN* (pp. 794–799).

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: An adaptive resonance architecture for incremental learning of analog maps. In: *Proc. IJCNN* (pp. 309–314).

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, *4*(6), 759–771.

Fritzke, B. (1994a). Fast learning with incremental radial basis function networks. *Neural Processing Letters*, *1*(1), 2–5.

Fritzke, B. (1994b). Growing cell structures — a self-organizing network for unsupervised and supervised learning. *Neural Networks*, *7*(9), 1441–1460.

Fritzke, B. (1995). A growing neural gas network learns topologies. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Advances in neural information processing systems*: *7* (pp. 625–632). Cambridge MA: MIT Press.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*(4), 193–202.

Furao, S., & Hasegawa, O. (2006). An incremental network for on-line unsupervised classification and topology learning. *Neural Networks*, *1*(19), 90–106.

Garcia, L.-M., Oliveira, A. A. F., Grupen, R. A., Wheeler, D. S., & Fagg, A. H. (2000). Tracing patterns and attention: Humanoid robot cognition. *IEEE Intelligent Systems*, *15*(4), 70–77.

Hamker, F. H. (2001). Life-long learning cell structures — continuously learning without catastrophic interference. *Neural Networks*, *14*, 551–573.

Heinze, A., Gross, H.-M., & Surmeli, D. (2001). Integration of a Fuzzy ART approach in a biologically inspired sensorymotor architecture. In: *Joint conference on neural networks* (pp. 1261–1266).

Izquierdo, I., Medina, J. H., Vianna, M. R., Izquierdo, L. A., & Barros, D. M. (1999). Separate mechanisms for short- and long-term memory. *Behavioral Brain Research*, *103*(1), 1–11.

Jebara, T., & Pentland, A. (1999). Action reaction learning: Automatic visual analysis and synthesis of interactive behaviour. In: *Int. conf. computer vision systems*.

Kalinke, T., & von Seelen, W. (1996). Entropie als Mass des lokalen Informationsgehalts in Bildern zur Realisierung einer Aufmerksamkeits-steuerung. In: *Proc. DAGM-symposium* (pp. 627–634).

Kohonen, T. (1989). *Springer series in information sciences. Self-organization and associative memory* (third ed.). Springer-Verlag.

Körner, E., Gewaltig, M.-O., Körner, U., Richter, A., & Rodemann, T. (1999). A model of computation in neocortical architecture. *Neural Networks*, *12*(7–8), 989–1005.

Littlestone, N. (1988). Learning when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, *2*, 285–318.

Maquet, P. (2001). The role of sleep in learning and memory. *Science*, *294*, 1048–1052.

Miyashita, Y., & Hayashi, T. (2000). Neural representation of visual objects: Encoding and top-down activation. *Current Opinion in Neurobiology*, *10*(2), 187–194.

Nayar, S. K., Nene, S. A., & Murase, H. (1996). Real-time 100 object recognition system. In: *Proc. of ARPA image understanding workshop*.

O'Reilly, R. C., & Norman, K. A. (2002). Hippocampal and neocortical contributions to memory: advances in the complementary learning systems framework. *Trends in Cognitive Sciences*, *6*(12), 505–510.

Palmeri, T. J., & Gauthier, I. (2004). Visual object understanding. *Nature Reviews Neuroscience*, *5*, 291–303.

Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, *2*(11), 1019–1025.

Rigoutsos, I., & Wolfson, H. (1997). Geometric hashing: An overview. *CSAE: Computational Science & Engineering, IEEE Computer Society*, *4*, 10–21.

Rolls, E. T., & Milward, T. (2000). A model of invariant object recognition in the visual system: Learning rules, activation functions, lateral inhibition and information-based performance measures. *Neural Computation*, *12*(11), 2547–2572.

Roth, D., Yang, M.-H., & Ahuja, N. (2002). Learning to recognize 3d objects. *Neural Computation*, *14*(5), 1071–1104.

Roy, D., & Pentland, A. (2002). Learning words from sights and sounds: a computational model. *Cognitive Science*, *26*(1), 113–146.

Schiele, B., & Crowley, J. L. (2000). Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, *36*(1), 31–50.

Scoville, W. B., & Milner, B. (1957). Loss of recent memory after bilateral hippocampal lesions. *The Journal of Neuropsychiatry and Clinical Neurosciences*, *20*, 11–21.

Squire, L. R., & Zola-Morgan, S. (1991). The medial temporal lobe memory system. *Science*, *253*, 1380–1386.

Steels, L., & Kaplan, F. (2001). AIBO's first words: The social learning of language and meaning. *Evolution of Communication*, *4*(1), 3–32.

Steil, J. J., & Wersing, H. (2006). Recent trends in online learning for cognitive robotics. In: *Proc. Eur. symp. artif. neur. netw.* (pp. 77–87).

Tarr, M. J., & Bülthoff, H. H. (1998). Image-based object recognition in man, monkey, and machine. In *Images-based Recognition in Man, Monkey, and Machine [Special issue] Cognition*, *67*, 1–20.

Vijayakumar, S., D'Souza, A., & Schaal, S. (2005). Incremental online learning in high dimensions. *Neural Computation*, *17*(12), 2602–2634.

Wersing, H., Kirstein, S., Götting, M., Brandl, H., Dunn, M., & Mikhailova, I. (2006). A biologically motivated system for unconstrained online learning of visual objects. In: *Proc. international conference on artificial neural networks*, Vol. 2 (pp. 508–517).

Wersing, H., & Körner, E. (2003). Learning optimized features for hierarchical models of invariant object recognition. *Neural Computation*, *15*(7), 1559–1588.

Wirth, S., Yanike, M., Frank, L. M., Smith, A. C., Brown, E. N., & Suzuki, W. A. (2003). Single neurons in the monkey hippocampus and learning of new associations. *Science*, *300*, 1578–1581.