Adaptive Scene-Dependent Filters for Segmentation and Online Learning of Visual Objects

Jochen Steil, Michael Götting, Heiko Wersing, Edgar Körner, Helge Ritter

2007

Preprint:

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

Neurocomputing 70, pp 1235-1246, 2007.

Adaptive scene dependent filters for segmentation and online learning of visual objects

J. J. Steil[1,2], M. Götting[1], H. Wersing[2], E. Körner[2] and H. Ritter[1]

1- Bielefeld University - Neuroinformatics Group, Faculty of Technology
P.O.-Box 10 01 31, D-33501 Bielefeld - Germany

2- Honda Research Institute GmbH
Carl-Legien-Str. 30, 63073 Offenbach - Germany

Abstract

We propose the Adaptive Scene Dependent Filter (ASDF) hierarchy for unsupervised learning of image segmentation, which integrates several processing pathways into a flexible, highly dynamic, and real-time capable vision architecture. It is based on forming a combined feature space from basic feature maps like, color, disparity, and pixel position. To guarantee real-time performance, we apply an enhanced vector quantization method to partition this feature space. The learned codebook defines corresponding best-match segments for each prototype and yields an oversegmentation of the object and the surround. The segments are recombined into a final object segmentation mask based on a relevance map, which encodes a coarse bottom-up hypothesis where the object is located in the image. We apply the ASDF hierarchy for preprocessing input images in a feature-based biologically motivated object recognition learning architecture. and show experiments with this real-time vision system running at 6 Hz including the online learning of the segmentation. Because interaction with user is not perfect, the real world system acquires useful views effectively only at about 1.5 Hz, but we show that for training a new object one hundred views taking only one minute of interaction time is sufficient.

1 Introduction

As robotic systems become increasingly human centered, real-time capabilities for processing and behavior as well as online learning gain importance to enable flexible man-machine interaction. In this setting, cognitive vision systems today have to meet the computational challenge to process visual input with a sufficiently high frame-rate, to control their attention, and to enable

visual learning and recognition of objects while interacting with the user. To facilitate visual processing and to reduce search spaces, many approaches use attention-based vision control to generate fixations. On the lower level, these are mostly based on topographically ordered maps to focus the system resources to certain points of interest (1; 2; 3; 4). Such maps use rather simple stimuli like color, oriented edges, or intensity, although mechanisms to integrate higher level information have also been proposed (4).

It remains difficult, however, to bridge the gap between the low level perceptual cues and the symbolic levels of object representations and up to date only few vision systems meet the computational challenges to enable real-time online learning. Therefore, and due to the difficulty of scene segmentation and object recognition in real-world scenes, most work in this area has been concentrated on explicitly or implicitly constrained scenarios. In most systems recognition performance thus depends crucially on assuming uncluttered background, homogeneous coloring of foreground objects, or predefined object classes to facilitate segmentation of objects against their surrounding. One such approach to reach the semantic level based on an attention system is to search for known objects at the current fixation point with a holistic object classification system (5) and to store objects recognized in a symbolic memory (6; 7). But the system needs a large amount of training images from different views, and the object classification itself has to be trained offline and beforehand. Other powerful approaches for object learning were developed using probabilistic and Bayesian methods (8; 9; 10), but these methods are computationally very demanding as well and consequently not suitable for online and interactive learning.

An interesting approach to online learning is presented in (11) and enhances a view-based holistic object recognition system proposed in (6). It uses a classification architecture which consists of feature extraction based on vector quantization and principal component analysis and a supervised classification scheme using a local linear map (12) architecture. Image acquisition is triggered by pointing gestures indicating objects on a table and is followed by a training phase taking some minutes. A different and more biologically inspired neural approach is suggested in (13; 14), which shows online learning capabilities by using precomputed and hierarchically generated robust sparse feature sets. Here object-specific learning is directed to the highest levels of a visual hierarchy. This recognition architecture is a part of a stereo vision framework for visual object recognition, which has been described in (15). It has recently been extended to use a combination of short and long-term memory systems and temporal integration to facilitate online learning driven by user feedback in an incremental manner (16). In the experimental part of the paper, we apply the ASDF segmentation approach in this framework and show that recognition performance and incremental interactive learning are improved.

It is generally believed that segmentation and recognition are closely connected and some authors try to solve both approaches concurrently (17), which results in rather complex architectures without online capabilities. In more classical approaches, segmentation is treated as an independent preprocessing step towards recognition. For the architecture proposed in (13; 14) it has explicitely been shown that good segmentation of objects in cluttered background supports the performance for object learning as well as for recognition (14). In such learning contexts it is crucial to use unsupervised segmentation, because a priori knowledge about the object to segment is not available.

To enable unsupervised segmentation, several cluster-based segmentation approaches (18; 19) use different color spaces and sometimes the pixel coordinates as feature space. They apply a vector quantization method like k-means or self organizing maps (SOM) to partition this space and segment the image with respect to the codebook vectors. Similarly, some approaches index the colors, quantize this index space, and back-project this quantization to segments (21; 22). Though such quantization methods can potentially be fast, they assume that objects have to be homogeneously colored and can be covered by one segment. Another approach for unsupervised segementation based on combinations of image features is normalized cuts (20). However, the respective computational burden prevents this otherwise very successful algorithm froom application in our system.

If stereo images are available, disparity information can be used as a segmentation cue (23) and some approaches try to support unreliable disparity information by additional color segmentation (24). In these schemes, color segmentation is not learned and uses strong underlying homogeneity assumptions. Implicitly it is also assumed that the objects to be segmented are isolated from each other. In real scenarios this typically is not the case, in particular not if humans manipulate and present objects to the vision system like we do in the experiments presented in this paper.

Some approaches have been made to combine unsupervised color clustering methods with top-down information about the object derived from other sources (25; 26). Such methods have the advantage that in the unsupervised step smaller segments can be generated which may over-segment the objects. Therefore homogeneity assumptions can be relaxed, however, the top-down information must be sufficient to resolve the resulting ambiguities. In (27; 28; 25), the unsupervised step consequently consists of generating a hierarchy of segments ordered in a tree. A successive optimization procedure to label the segments as belonging to the object with respect to a cost function is based on the top-down information. The complexity of this method is linear in the number of pixels, but still not sufficiently fast to allow real-time performance processing with several frames per second.

Fig. 1. A typical input image (left), the relevance hypothesis mask for the object (center), and the final ASDF object segmentation mask (right).

In view of these difficulties, we present the Adaptive Scene Dependent Filter (ASDF) hierarchy and its application in the online learning environment (14) featuring two kinds of learning: adaptive filters in the processing and their contribution in an overall biologically motivated cognitive vision learning architecture. The ASDF's serve as a preprocessing step to segment objects in front of an unconstrained background, such that the segmented images facilitate processing in the higher level object recognition stages. The ASDF hierarchy enhances simpler unsupervised image segmentation methods by combining several processing pathways into a flexible, highly dynamic, online and robust image segmentation architecture. It is based on input feature maps typically available from an underlying attentive system. Contrary to pure color segmentation schemes, we allow for combinations of all kinds of topographic feature maps like edge maps, intensity, difference images, velocity fields, disparity, image position, or different color spaces for forming a combined feature space. In this paper, for the sake of computational efficiency we apply a straightforward vector quantization method to partition this feature space, though more advanced schemes like growing networks as the growing neural gas (29) or the instantaneous topological map (30) can also be used. The vector quantization generates a codebook and respective best-match segments for each codebook vector, which define an over-segmentation of the object and the surround. The segments are recombined in a later processing step such that we do not have to rely on a homogeneity assumption on the object, see e.g. the multicolored bottle in Fig. 1. On the other hand we enforce a bound on the number of segments to prevent strong over-segmentation in small parts and to preserve computation time.

Our recombination step uses information from a relevance map to determine which segments belong to the object (see Fig. 1). As opposed to the approach in (25), this recombination step does not use an explicit hypothesis about the object to be learned and relies exclusively on bottom-up information from the attentional system. It defines a region of interest at the current fixation point, which can be refined by disparity information or other cues, if those are available. Additionally to the ASDF's and the relevance map, we can also exclude specific segments based on information from a third processing path.

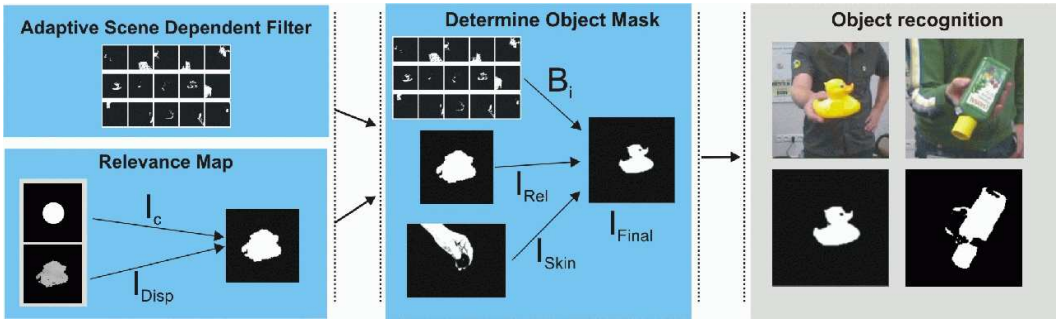

Fig. 2. The multi-path ASDF hierarchy for image segmentation and object recognition using Adaptive Scene Dependent Filters, a relevance map, skin color detection, the determination of the object segmentation mask and the object recognition module.

We use this to subtract regions representing skin and hand color that are detected using a separate specialized skin color detection. Note that if in this way complete segments or connected components of segments are accepted, then also pixels which fall outside the initial interest region can be included in the final mask. Objects present in the input image but outside the interest region are not segmented, which saves computation time. The architecture can be applied to all kinds of images in order to segment objects in the focus of attention defined by the relevance map. In particular we use it in the context of online learning of "objects-in-hand" presented by a human partner in front of an unconstrained background.

2 The Adaptive Scene Dependent Filter architecture

The Adaptive Scene Dependent Filters (ASDF's) are embedded in an ASDF hierarchy, which is a multi-path segmentation architecture as shown in the overview in Fig. 2. In the framework of the online learning system (16) for complex shaped objects, it is applied as preprocessing step for the feature-based object recognition architecture described in (13), which is the rightmost block in Fig. 2. There are three independent computation paths: the Adaptive Scene Dependent Filter learning system, the relevance mask, and the skin color filtering (top to bottom in the middle block in Fig. 2). These three paths can be scheduled in parallel for real-time performance. They project to an integration block computing the final object segmentation mask. These core components are described in more detail below.

We assume that in earlier stages of the vision architecture low level filter operations on an input image are provided which usually are combined by an additive weighting in a fixation-guiding saliency map. The input to the ASDF computation therefore are M basic filter maps F_i with feature responses $m^i_{(x,y)}, i = 1..M$ at pixel positions (x, y). We further assume that from the

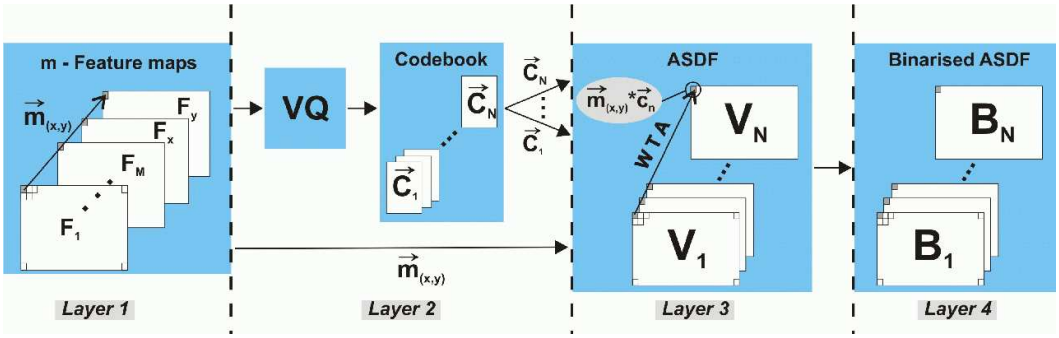

Fig. 3. The adaptive filter computation. In layer 1, M feature maps are computed from the input image and combined with two feature maps carrying the local x and y coordinate respectively. In the second layer, a VQ network is used to generate a codebook of prototypical feature combinations. A WTA algorithm is used in the third layer to compute the N scene dependent filters. In the last layer, a binarization of these filters is performed.

image acquisition and the underlying vision framework some information is available which can be used to generate an interest region that we call relevance map.

2.1 Adaptive Scene Dependent Filters

The ASDF path is organized in four layers as shown in Fig. 3. In the first layer, at each pixel position a combined feature vector consisting of weighted answers of the underlying attentional feature maps is formed. In the second layer, these feature vectors are input for a vector quantization (VQ) network to obtain N prototypic codebook vectors $\vec{c}^{\,j} \in C$. The codebook C therefore represents the distribution of feature combinations by compressing it to prototypes, which represent clusters of feature combinations occurring in the current image. The codebook vectors are used to generate new adaptive topographic activation maps in layer 3, which are binarized by a winner-take-all mechanism in layer 4 accordingly.

In layer 1, weighted feature vectors $\vec{m}_{(x,y)}$ are formed according to

$$\vec{m}_{(x,y)} = \left(\xi^1 \frac{m^1_{(x,y)}}{\sigma(m^1)^2}, ..., \xi^n \frac{m^M_{(x,y)}}{\sigma(m^n)^2}, \xi^x m^x_{(x,y)}, \xi^y m^y_{(x,y)} \right)^T,$$

where (x, y) is the respective pixel index and $m^x(x, y) = x, m^y(x, y) = y$ include the pixel position as feature. Each component is normalized by its variance $\sigma(m_i)^2$. ξ^i is an heuristically determined factor weighting the relative importance of different filter maps (see Appendix for parameter settings).

The layer 2 employs a customized VQ network with a fixed number of train-

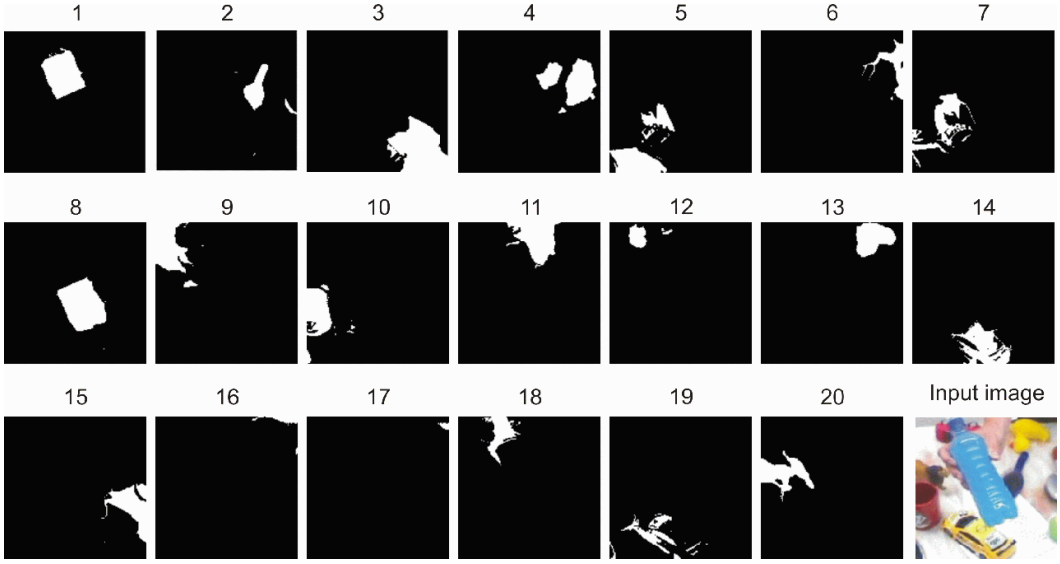

Fig. 4. Binarized ASDF segments B_i. Note the over-segmentation of the homogenous bottle region in segments 1 and 8, which together with segment 18 constitute the final object mask for the input image.

ing steps (to speed up and control the complexity of computation), a fixed number of prototypes, a special scheme for initialization them, and an activity equilibration to regularize performance as suggested in (31). Details of the algorithm and parameters are given in the Appendix. Between subsequent input frames we do not reinitialize the codebook vectors. This implements a form of continuity between frames that is highly beneficial due to the inherent constancy of the presented scene. Consequently, the VQ centers can track such segments of the scene, which are only slowly moving or changing, while the activity equilibration can quickly dedicate prototypes to newly appearing feature combinations.

The input for layer 3 consists of the adaptive codebook C and the basic filter maps F_i. It computes N scene dependent activation maps V^j as $V^j_{(x,y)} = \parallel \vec{m}_{(x,y)} - \vec{c}^{\,j} \parallel^2$, which are binarized in layer 4:

$$B^j_{(x,y)} = \begin{cases} 1 & \text{if } \parallel \vec{m}_{(x,y)} - \vec{c}^{\,k} \parallel^2 < \parallel \vec{m}_{(x,y)} - \vec{c}^{\,j} \parallel^2 \,, \forall k \neq j \\ 0 & \text{else} \end{cases} .$$

A typical result including an example of over-segmentation of a homogenous object region is shown in Fig. 4.

2.2 Relevance Map and Skin Color Mask

In the second path, a relevance map is computed which serves as a hypothesis mask to predict a region around the focused object. This mask is used as a

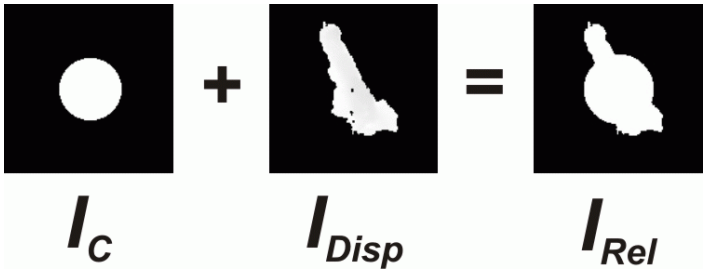

Fig. 5. The relevance map I_{Rel} as superposition of a center map and a disparity map. The center map I_c represents the a priori assumption that objects are centered which is a result of the attention system selecting the object position.

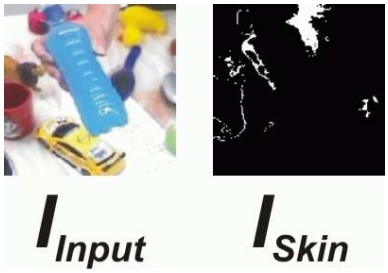

Fig. 6. A result of the skin color segmentation algorithm, which allows to remove the hand from the final object mask.

cue to select appropriate binary segments from layer 4 of the ASDF, which are recombined for the final segmentation mask. For the experiments given in the current paper, we compute the relevance map I_{Rel} as superposition of the center map I_C and a disparity map obtained from the stereo vision framework I_{Disp} around the focused object (see Fig. 5). The center map I_c encodes the a priori assumption that objects are approximately centered. This is reasonable because the current fixation is based on an attentive system driving the gaze position to salient objects (see also (15)). In principle, every other (visual) cue indicating a region where the object is expected to be located could be used here.

In the third path, we use a skin color segmentation algorithm to subtract skin color regions from the final object mask, see the example in Fig. 6. Though there are more sophisticated adaptive algorithms available if illumination changes slowly and the skin color histrogram can be predicted (32), we stick to a static version of the segmentation developed in (33), because skin color changes between consecutive frames can be very large due to reflection from the objects, the large posture variance, and occlusions. Thus a ground truth for adaptation is difficult to obtain and a conservative static estimation proves to work sufficiently well in our setting.

Fig. 7. The online learning scenario for recording training and testing images. Objects are shown freely by hand and a gaze control system using disparity focuses on the objects that are in the peripersonal space between user and the camera.

2.3 Determine Object Segmentation Mask

In the integration step, the challenge for the *Determine Object Mask*-block is to select a subset of the ASDF binary segments, which are recombined to form the final segmentation mask for the focused object.

As described before, the conceptually intuitive and straightforward way to form the final filter mask is to first compute the ASDF filters and their binarized segments and then to select suitable segments among these. One natural way of selection is to compute the overlap of the ASDF segments with the relevance mask. Thus the number of pixels $inPix$ of the intersection I_{Rel} and B_i ($inPix = \#(B_i \cap I_{Rel})$) and the number of pixels $outPix$, B_i without I_{Rel} ($outPix = \#(B_i \setminus I_{Rel})$) are computed. Then the probability of mask B_i to belong to the object is estimated by the relative frequency $outPix/inPix$. The mask is included in the final segment mask I_{final} if $outPix/inPix < 0.2$. The final mask I_{Final} is then computed as the additive superposition of the selected B_i and the skin color pixels are removed from this mask ($I_{Final} = \sum_i B_i - I_{Skin}$).

Though results obtained with a first implementation of this scheme were comparable to those given below, the computational effort was too high for full real-time performance, mainly because at first for each of the codebook vectors the ASDF filter is evaluated on the whole image. This step turns out to be the main computational burden of the whole processing hierarchy which determines the overall frame rate (of the ASDF hierarchy). Thus, to ensure reasonable frame rates for online real-time learning, in the experiments given below a computationally much more efficient shortcut is used.

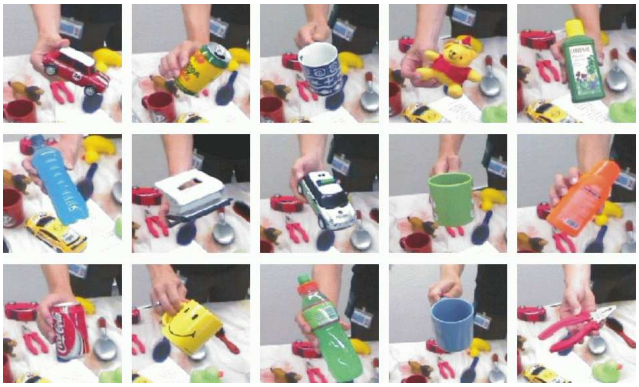

Fig. 8. The set of 15 objects used for training.

2.4 Codebook related relevance detection

The computational effort can be significantly reduced, if the actual computation of the ASDF filter outputs and the WTA is delayed and moved to the determine object module. This conceptually dispenses with the organization in clear distinctive layers, but allows to compute ASDF filter answers only on those pixels which belong to the final segmentation mask. Therefore currently the following scheme is implemented, which relies on the fact that the prototype vectors $\vec{c}_j$ carry pixel position information (c_x^j, c_y^j): For detection whether the pixel p at position (x, y) has to be included in the final segmentation mask, the reference position (c_x^k, c_y^k) of the winning codebook vector $\vec{c}_k$ is computed (i.e. $\vec{c}_k$ is the codebook vector with $k = \text{argmin}_j \parallel \vec{c}^j - \vec{m}_{(x,y)} \parallel^2, \vec{c}^j \in C$). The pixel $p_{(x,y)}$ is selected, if the position of the winning codebook vector is inside the relevance map region. For a further speed up only those pixels $p_{(x,y)}$ are considered for evaluation, which are located inside the relevance map.

3 Experimental results

3.1 Experimental Setup

To evaluate the performance of the ASDF in combination with the recognition architecture, an image database of 15 every-day objects of differing colors and shapes in front of a cluttered background has been recorded with 100 training images from a first training person and 100 test images for each object from a different person (see Fig. 7 and Fig. 8). The overall vision system runs currently at 6 Hz including image recording, control of stereo vision, ASDF, and object recognition. If the interaction in the online system is sucessful and the object is in the focus of the system, then object images for training or recognition are acquired with this rate. However, in a real scenario the human user presents

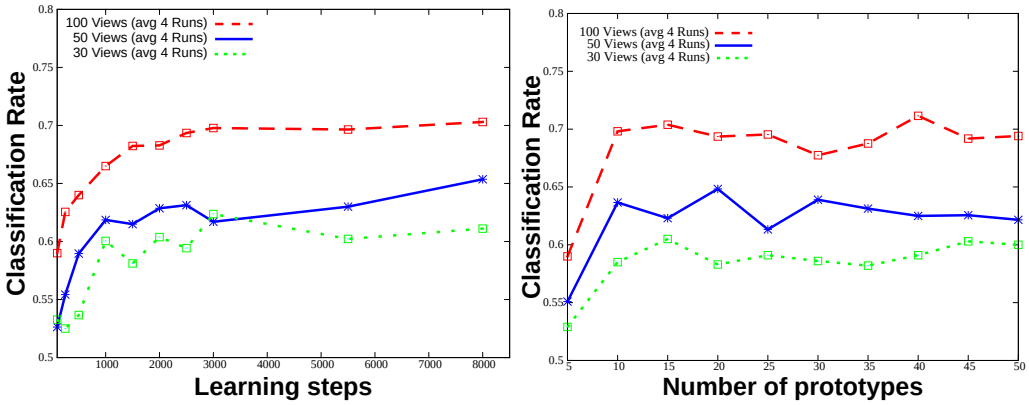

Fig. 9. Evaluation of the performace with respect to the most important training paramters, for learning steps (left) and number of prototypes (right). All plots display avarages over the 15 objects and 4 complete training sessions.

the objects not perfectly and sometimes the objects are held out of the focus of the gaze control system or laid back on the table. Therefore not every frame will give a region of interest with an object within the central peri-personal space (see (15) for details) and consequently the effective average acquisition frame rate for object views to be classified in the running hardware system is in our experiments only about 1.5 Hz (for more details on the interaction see (16)). We therefore have to distinguish the real-time system performance of 6 Hz considering only processing in the vision system and controlling the camera head from the effective training time used in the real interactive system including the human presenter in the loop. To acquire 100 views the online system, we need about one minute effective training time still resulting in a fairly convenient and fast online training.

As input for the ASDF, we use the following feature maps: red, green, blue, disparity, and the pixel positions. Fig. 9 shows the dependency of the classification performace averaged over all 15 objects and four complete training runs evaluated on the must crucial learning paramters: the number of prototypes for the activity equilibrated vector quantization and the number of training steps. Note that those two paramters also mainly determine the computational cost of the segementation process. We find that a minimum number of approximately 2500 training steps and 15 prototypes is required for the given training scenario, while supplying larger numbers in either cases does not gain much performance. To be robust against changing the visual input, we settle for a standard setting (Fig. 3) of 20 prototypes and 3000 training steps per image with a constant learning rate to ensure controlled online performance. Note that the VQ codebook is not reinitialized over images frames, i.e. the VQ learning ensures a certain time integration of the segments. This regularizes and stabilizes performance and allows to keep the number of learning steps per image at such a rather small number. Fig. 9 also shows that qualitatively the curves have a similar behavior for different numbers of training views,

where we subsample in regular intervals from the training set if less than 100 views are considered. Clearly 30 views is not enough for stable performance, 50 views, i.e. every second view of the trainig set, perform better but not yet sufficient, but 100 views, which correspond to about one minute of effective training time as dicussed above, give already a very good and smooth performance.

3.2 ASDF Mask evaluation

It is difficult to evaluate the segmentation quality of the ASDF mask in the current online learning setting, because to get a ground truth mask the images would have to be segmented by hand. This obviously is not suitable for the large number of views we acquire even in a single training session. Therefore Fig. 10 gives a visual impression of the segmentation for two complex objects in comparison to a mask given by disparity information only. This comparison is reasonable, because in the given setting the user presents the object with the hand in front of his body, such that in many cases a good disparity is available and the system performs quite well even when using only this information. Fig. 7 shows that nevertheless the ASDF masks fits the object contour much better in many cases. Besides visual inspection or hand segmentation for getting ground truth, the only evaluation of the segmentation quality is in connection with a recognition architecture and its classification performance as we do in the next sections. Note that all masks are rotation normalized along their principal axis and centered on their centroid, which facilitates recognition considerably according to the results given below.

3.3 Overall classification

We first investigate the overall performance gain that can be obtained for object recognition using the ASDF masks. We base our classification scenario on a simple nearest-neighbor-based classifier, which was shown to perform quite well in related scenarios (13). To separate the effect of using a more sophisticated hierarchical feature representation we use as input vectors either the plain (masked or unmasked) 3x144x144 pixel RGB images or the output of the visual feature hierarchy (13) with 53x8x8 dimensions. We vary the number of training vectors between 10 and 100, where we sample in regular intervals from the 100 training views that are available for each of the 15 objects. Additionally, tests with the following configuration sets for the object recognition architecture are performed:

- No mask information is used, and the image is taken unchanged.

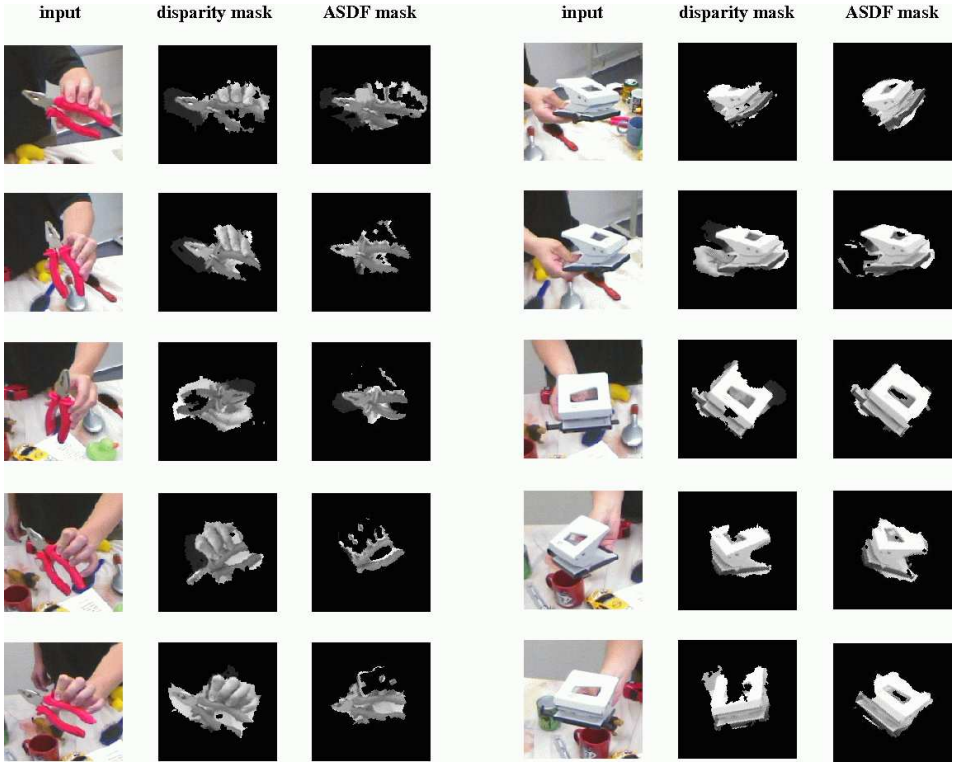

Fig. 10. Original input images together with the disparity mask and the ASDF segmentation masks, both rotation normalized.

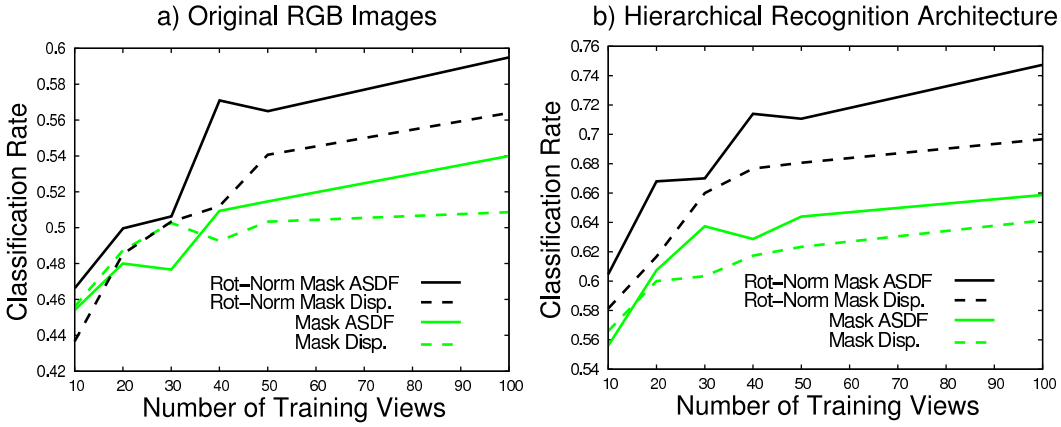

Fig. 11. Comparison of classification rates over number of training vectors using disparity and ASDF masks for direct RGB input images a) and using the recognition hierarchy in b). For the plain images in a) the ASDF masks give an improvement for sufficiently many training views, both with and without rotation normalization. Using the hierarchy for recognition in b), the performance gain is even stronger for using ASDF masks. Note that the overall classification rate is considerably larger in b), compared to a).

- A mask is generated either using ASDF, or taking the disparity relevance map and the image is masked using this mask.
- A mask is generated for ASDF and disparity, the image is then masked and

also rotation-normalized along the principal axis of the mask.

The rotation normalization is strongly dependent on a proper segmentation of the object. We also shift the mask according to its centroid within the input frame.

The results of this experiment are shown in Fig. 11. In Fig. 11 a) we show the classification rate over different numbers of training views when the original RGB images are used for classification. For smaller number of views a stronger fluctuation can be observed due to the gap-like sampling of the training history. For larger numbers of training views, the ASDF method achieves a clear performance gain over using only the disparity. The rotation normalization gives a further improvement of the classification rate. Using the hierarchy considerably improves classification performance in Fig. 11 b) (about 15%, which corresponds to a 40% reduction of the error rate), as can be seen from the labeling of the axis for the classification rate.

Again the ASDF masks improve classification performance, especially for the case of rotation-normalized masks. From inspection of the masks and the experience with the experiments we believe that the main reason for improvement is the removal of spurious surrounding of the object, which allows to better center and normalize the mask. As the recognition architecture is form and color sensitive, it can profit from this normalization and regularization. This holds in particular for complex shaped objects like those shown in Fig. 10, where we observe the largest performance increase.

If we do not use the mask information and use the full 144x144 pixel RGB image, then the classification rate on the original RGB images lies only between 15% (10 views) and 20% (100 views). If we use the feature hierarchy for recognition this is only slightly enhanced to 23-26%. This highlights the importance of obtaining a good segmentation of the object within the initial region of interest.

3.4 Incremental classification test

To evaluate the incremental learning and online performance of the overall architecture, we train first 14 objects from the training database with their complete 100 training images. Then the 15th object is trained in steps of 10 images (7 sec in Fig. 12) and a validation step is performed. Test performance is measured over all 100 test images of the currently trained object giving the classification rate as percentage of correctly recognized test views of this object at this point of online learning. Then training proceeds until all 100 training images are used. The plot shows the average learning curve for adding each of the 15 objects as the final object to 14 previously learned objects.

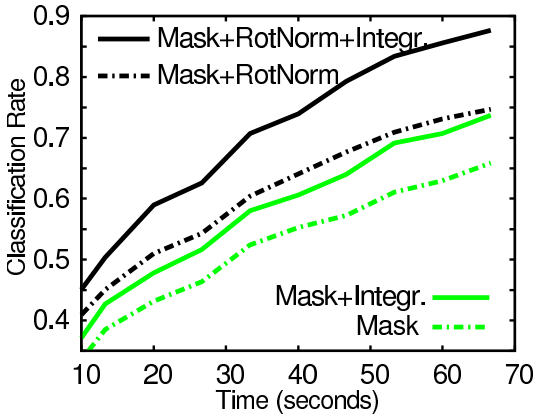

Fig. 12. Learning curve for online learning of an object. The plot shows the average classification rate for training the 15th object, after all other 14 objects have already been trained. We compare the performance when ASDF masks are used, with or without rotation normalization and temporal integration of classification results.

We compare in Fig. 12 the conditions of either using the rotation normalization with the ASDF masks or using the mask without rotation. Additionally we show the performance gain that can be obtained by a simple temporal integration scheme that uses a voting over the past 10 classifications. The class label is then assigned to the object, that was classified most often for the last ten times using the nearest-neighbor classifier. With this approach, almost 90% correct classification can be achieved, even after training each object only about 1 minute. The corresponding results for using the disparity mask or no mask at all follow the same pattern as can be observed in Fig. 11. The final performance after training all 100 views corresponds to the value for 100 training views in Fig. 11.

4 Conclusion

In this paper, we present the ASDF hierarchy for unsupervised image segmentation and give results from integrating this scheme into an interactive online learning system. It enhances the performance of visual online object recognition in cluttered environments under "object-in-hand" presentation. The main advantages of the approach are its real-time capability, which is crucial for online learning and the flexible generation and usage of a relevance map to group adaptively generated segments to object specific masks for figure ground separation. We have shown that the architecture is reliable and fast enough to work online in real-time in a realistic human-machine interaction loop, where the effective learning time is determined by the speed of the learning vision system, here 6 Hz, as well as the smooth cooperation with the user, which in the presented examples lets the effective view acquisition rate drop to 1.5 Hz. Still this is enough to interactively and robustly train a new object from

scratch within one minute under fairly unconstraint general conditions.

The ability of changing the input stimuli for the relevance map as well as for the ASDF input makes it easy to tailor the architecture for a wide range of applications. The current setting mainly relies on the combination of disparity, robust skin color detection, and color features and has proven to be robust to different clothing of the presenting persons. Other applications like e.g. scene interpretation may also include texture filters, motion maps, or other specifically useful topographic information while keeping the overall processing structure. However, in such cases the overall frame-rate will drop relative to the computational effort to compute these features. This problem is the main reason for choosing relatively simple feature maps and to strictly limit the learning time in our architecture. We believe that on single images other not real-time capable methods like for instance normalized cuts might provide even better segmentation and enhance overall performance. The strength of our systems, however, lies in the architecture which combines many very simpler but faster processing elements to gain an excellent overall performance. We regard this an important step towards more flexible vision architectures to perform fast object recognition in more unconstraint environments.

Acknowledgment

We thank S. Kirstein, J. Eggert, A. Ceravola, and M. Dunn for providing the processing system infrastructure.

Appendix: Activity Equalization for vector quantization (AEV)

The activity equalization algorithm enhances standard vector quantization to better distribute the prototypes (codebook vectors) in the input space. A vector quantizer maps a set of data vectors $M = \{\vec{m}_i\}, \vec{m}_i \in \mathbb{R}$ into a finite set of vectors $C = \{\vec{c}_i\}$ with $|C| < |M|$. C is called the codebook, with codebook vectors $\vec{c}_i$. A data vector $\vec{m}_i$ is approximated by the best matching reference vector $\vec{c}_{k(\vec{m}_i)}$, where $k = \text{argmin}_j \parallel \vec{m}_i - \vec{c}_j \parallel$ and $\parallel \cdot \parallel$ is the Euclidean distance. The winning reference vector is then adapted by $c_k(t+1) = \vec{c}_k(t) + \epsilon(\vec{m}_i - \vec{c}_k(t))$, where $\vec{m}_i$ is the selected data vector and ϵ is the learning rate. An extension to the standard VQ is the Activity Equalization for Vector quantization algorithm (AEV) (31). It aims at repositioning unused prototypes in regions of low data density to regions with higher density. Additionally we perform an incremental buildup of the set of codebook vectors. The complete enhanced VQ learning algorithm is implemented as follows:

For each training iteration step perform steps 1 to 7:

(1) Choose a data vector $\vec{m}_{(x,y)}$ randomly
(2) Compute $d_{min} = min_j||\vec{m}_{(x,y)} - \vec{c}_j||$ of $\vec{m}_{(x,y)}$ to all other $\vec{c}_j$ in the current codebook
(3) Increase the codebook step-by-step: If the current number of codebook vectors is smaller than the maximal number N_C and if $d_{min} > \bar{d}$ assign a new codebook vector $\vec{c}^{j'} = \vec{m}_{(x,y)}$
(4) For all $\vec{c}_j \in C$ compute the winning reference vector with respect to $\vec{m}_{(x,y)}$
(5) Perform an adaption step for the winning reference vector $\vec{c}_k$ with $k = \text{argmin}_j ||\vec{m}_{(x,y)} - \vec{c}_j||$.
(6) Calculate the node activity $A(j)$ for each node j
$A(j) = \sum_{\vec{x}_i \in C} Winner(\vec{m}_{(x,y)}, j)$ with
$$Winner(\vec{m}_{(x,y)}, j) = \begin{cases} 1 \text{ if node } j \text{ is best match of } \vec{m}_{(x,y)} \\ 0 \text{ else} \end{cases}$$
(7) Perform a re-positioning of idle nodes each Q training steps if $A(j) < \gamma$ where γ is a lower bound for the activity count. For each j with $A(j) < \gamma$ do: i) Determine the most active node k with $A(k) > A(j)$ for all j, ii) Place the idle node j in the vicinity of the maximally activated node k with $\vec{c}_j = \vec{c}_k + \vec{x}$, where $\vec{x}$ is a uniformly distributed random vector with components $x_i \in [-\eta, \eta]$.

To obtain the vector quantization in the ASDF module we perform n iterations of steps 1-7.

The parameters for the vector quantization are set to the following values: The learning rate is set to $\epsilon = 0.01$ and the number of training steps is set to $n = 3000$. The repositioning is done all $Q = 500$ steps with a range of $\eta = 1$ and $\gamma = 10$. The scaling values for the feature vectors $\vec{m}$ are: $\xi^{\{1..3\}} = 1000$ for the RGB maps, $\xi^4 = 5$ for the disparity map, which has values zero for background and 255 for foreground, and $\xi^{\{x,y\}} = 0.1$ for the spatial location. The maximum number of code vectors is $N_C = 20$, new codebook vectors are assigned if $d_{min} > 600$. The codebook is initialized only once at the beginning of a learning session and then kept over frames for temporal integration.

References

[1] J. A. Driscoll, R. A. P. II, K. R. Cave, A visual attention network for a humanoid robot, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-98), Victoria, B.C., 1998.
[2] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for

rapid scene analysis, IEEE Trans. Pattern Anal. Mach. Intell 20 (11) (1998) 1254–1259.

[3] C. Breazeal, B. Scassellati, A context-dependent attention system for a social robot, in: Proc. 16th Int. Joint Conf. on Artif. Intell. (IJCAI-99-Vol2), 1999, pp. 1146–1153.

[4] J. J. Steil, G. Heidemann, J. Jockusch, R. Rae, N. Jungclaus, H. Ritter, Guiding attention for grasping tasks by gestural instruction: The gravis-robot architecture, in: Proc. IROS 2001, IEEE, 2001, pp. 1570–1577.

[5] H. Ritter, J. J. Steil, C. Noelker, F. Roethling, P. McGuire, Neural architectures for robotic intelligence, Rev. Neurosci. 14 (1-2) (2003) 121–143.

[6] G. Heidemann, A multi-purpose visual classification system, in: B. Reusch (Ed.), Proc. 7th Fuzzy Days, Dortmund, 2001, Springer-Verlag, 2001, pp. 305–312.

[7] G. Heidemann, H. Ritter, Combining multiple neural nets for visual feature selection and classification, in: Proceedings of ICANN 99, 1999.

[8] K. B., C. M. Bishop, M. Szummer, Generative models and Bayesian model comparison for shape recognition, in: Proceedings Ninth International Workshop on Frontiers in Handwriting Recognition, 2004.

[9] J. Winn, N. Jojic, LOCUS: Learning object classes with unsupervised segmentation, in: Proc. ICCV05, 2005, pp. I: 756–763.

[10] C. K. I. Williams, M. K. Titsias, Greedy learning of multiple objects in images using robust statistics and factorial learning, Neural Computation 16 (5) (2004) 1039–1062.

[11] G. Heidemann, H. Bekel, I. Bax, H. Ritter, Interactive online learning, Pattern Recognition and Image Analysis 15 (1) (2005) 55–58.

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

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

[14] S. Kirstein, H. Wersing, E. Körner, Rapid online learning of objects in a biologically motivated recognition architecture, in: 27th Pattern Recognition Symposium DAGM, Springer, 2005, pp. 301–308.

[15] C. Goerick, H. Wersing, I. Mikhailova, M. Dunn, Peripersonal space and object recognition for humanoids, in: Proc. IEEE Humanoids, Japan, 2005.

[16] H. Wersing, S. Kirstein, M. Götting, H. Brandl, M. Dunn, I. Mikhailova, C. Goerick, J. Steil, H. Ritter, E. Körner, A biologically motivated system for unconstrained online learning of visual objects, in: Proc. Int. Conf. Art. Neur. Netw. ICANN, 2006.

[17] S. X. Yu, R. Gross, J. Shi, Concurrent object recognition and segmentation by graph partitioning, Online proceedings of the Neural Information Processing Systems conference.

[18] G. Dong, M. Xie, Color clustering and learning for image segmentation based on neural networks, IEEE Trans. on Neural Networks 16 (14) (2005) 925–936.

[19] Y. Jiang, Z.-H. Zhou, SOM ensemble-based image segmentation, Neural Processing Letters 20 (3) (2004) 171–178.

[20] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE PAMI 22 (8) (2000) 888-905

[21] J. K. Robert Li, Image compression using fast transformed vector quantization, in: Applied Imagery Pattern Recognition Workshop, 2000, p. 141.
[22] D. Comaniciu, R. Grisel, Image coding using transform vector quantization with training set synthesis, Signal Process. 82 (11) (2002) 1649–1663.
[23] N. H. Kim, J. S. Park, Segmentation of object regions using depth information, in: ICIP, 2004, pp. 231–234.
[24] H. Tao, H. S. Sawhney, Global matching criterion and color segmentation based stereo, in: Workshop on the Application of Computer Vision, 2000, pp. 246–253.
[25] E. Borenstein, E. Sharon, S. Ullman, Combining top-down and bottom-up segmentation, CVPRW, Washington D. C. 4 (2004) 46.
[26] M. Bravo, H. Farid, Object segmentation by top-down processes, Visual Cognition 10 (4) (2003) 471–491.
[27] R. B. E. Sharon, A. Brandt, Segmentation and boundary detection using multiscale intensity measurements, Proceedings IEEE Conference on Computer Vision and Pattern Recognition, Hawaii 1 (2001) 469–476.
[28] R. B. Meirav Galun, Eitan Sharon, A. Brandt, Texture segmentation by multiscale aggregation of filter responses and shape elements, IEEE Int. Conf. on Computer Vision 1 (2003) 716–723.
[29] B. Fritzke, A growing neural gas network learns topologies., in: Advances in Neural Information Processing System 7., MIT Press, 1995, pp. 625–632.
[30] J. Jokusch, H. Ritter, An instantaneous topological mapping model for correlated stimuli, in: Proc. Int. Joint Conf. on Neural Networks (IJCNN 1999), 1999, p. 445.
[31] G. Heidemann, H. Ritter, Efficient vector quantization using the WTA-rule with activity equalization, Neural Processing Letters 13 (1) (2001) 17–30.
[32] L. Sigal, S. Sclaroff, V. Athitsos, Skin color-based video segmentation under time-varying illumination, IEEE PAMI 26 (7) (2004) 862–877.
[33] J. Fritsch, S. Lang, M. Kleinehagenbrock, G. A. Fink, G. Sagerer, Improving adaptive skin color segmentation by incorporating results from face detection, in: Proc. IEEE ROMAN, IEEE, 2002, pp. 337–343.