**Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems San Diego, CA, USA, Oct 29 - Nov 2, 2007**

**ThB2.4**

# Exploiting Similarities for Robot Perception

Kai Welke* , Erhan Oztop†‡ , Gordon Cheng†‡ and Rüdiger Dillmann*
*University of Karlsruhe (TH), IAIM, Institute of Computer Science and Engineering (CSE)
P.O. Box 6980, 76128 Karlsruhe, Germany
†JST, ICORP, Computational Brain Project
4-1-8 Honcho, Kawaguchi, Saitama, Japan
‡ATR Computational Neuroscience Lab., Dept. of Humanoid Robotics and Computational Neuroscience
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan

*Abstract*— **A cognitive robot system has to acquire and efficiently store vast knowledge about the world it operates in. To cope with every day tasks, a robot needs to learn, classify and recognize a manifold of different objects. Our work focuses on an object representation scheme that allows storing perceived objects in a compact way. This will enable the system to store extensive information about the world and will ease complex recognition tasks. The human visual system deploys several mechanisms to reduce the amount of information. Our goal is to develop an artificial system that mimics these mechanisms to create representations that can be used in cognitive tasks. In particular, in this paper we will present an approach that exploits similarities among different views of objects. The proposed representation scheme allows for reduction of storage required for the representation of objects and preserves the information about the similarity among objects. This is achieved by selecting 'important views' of objects, depending on their stability. Furthermore, by extending the same approach to multiple objects, we are able to exploit similarities between objects to find a common representation and to further reduce the storage requirements.**

## I. INTRODUCTION

The main focus of our work is to develop an object representation and learning scheme, that is suitable for learning in humanoid robots, e.g. via action-perception coupling, much the same way as humans learn about objects in their environment. To achieve the ability of recognizing objects from all viewing directions, we introduced a learning and representation scheme that allows generalizing to specific views of objects [1]. We have shown that, given locations of important views, the objects can be represented in a depth rotational invariant manner, with a reduced amount of views stored as representations. However, in the former work the important views were selected manually.

In this paper, we introduce a solution on how to select specific important views of objects automatically. Our approach is driven by the observation that there are specific views of an object, that allow recognizing a wide range of rotational variations of the object. Such views are often referred to as *stable* views [2]. With these stable views of an object, its appearance can be described using a minimal set of views.

The robot perceives the world in different modalities, depending on the sensors and feature extraction methods used. In real world scenarios, the robot will face objects, which are similar in at least one modality and are only separable by combining different modalities. Furthermore, learning of objects from all possible viewing directions will reveal even more similarities between views of different objects. In our approach, we identify views that are shared between objects. Similar views can be subsumed and stored only once. In such cases we do not want to recognize objects in the modality, in which the similarities exist, but rather aim at a representation that preserves the information, which objects are candidates for the specific view and modality. The ability to discriminate such objects has to be achieved by combining multiple modalities. The shared view of objects in one modality can then be used to restrict the possibilities in other modalities to only a few objects.

Our approach follows a global appearance-based representation scheme of objects. In appearance-based vision systems, objects are represented with multiple retinal projections of object views. In contrast, model-based representations need more structural information, like full 3D models, which are hard to aquire during online learning [3]. Furthermore, we use global object descriptors to identify important views of the object. The majority of recent work on object recognition uses local features, which describe important locations in the object's appearance, considering measures for texturedness or cornerness. These systems perform well in real environments, are able to handle occlusion, and usually offer invariances to at least shift and rotation in the camera plane [4] [5]. Recognition of all rotations of an object with local features is possible, but impractical in terms of efficiency due to the amount of stored local feature representations.

The selection of important views of an object has strong correspondence to the notion of canonical views used in psychology [6]. In the past, different criterias that define canonical views have been introduced. Blanz et al. give a good overview of different criteria [7]. As our aim is to implement an object recognition system based on our representation and learning scheme, we are mainly interested in the *goodness for recognition* criteria. More precisely, we identify views of the objects, that are stable for at least small transformations.

While the work on canonical views copes largely with one outstanding view of the object, a representation scheme which is used for recognition has to rely on multiple important views of the object, which together should cover the

**1-4244-0912-8/07/$25.00 ©2007 IEEE.**

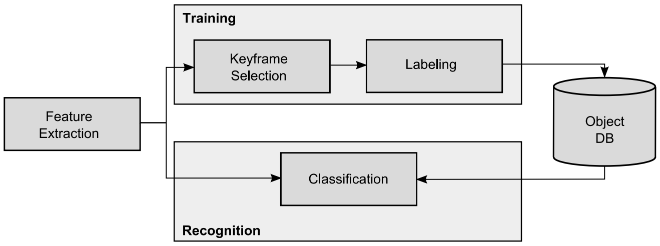


Fig. 1. System structure of the proposed learning and recognition system.

complete appearance.

Hall et al. presented an approach to extract multiple important views of an object by identifying the most unique views of the object [8]. The identification of unique views is suitable, if the objects are to be visualized or if the resulting views are used only to discriminate objects. The approach did not take into account the similarity of views. Moreover, the resulting views did not capture similarities among different objects.

Yamauchi et al. introduced an approach for the identification of important views which combines the saliency of views and the stability criterion [2]. They proposed an approach based on spherical graphs which reflect all available viewing directions of the object. Important views were identified using Zernike Moments [9] to measure the similarity between neighbored views in the spherical graph. In their work, Yamauchi et al. did not take into account similarities among different objects. Each object had its own set of keyframes regardless of the appearance of other objects. Furthermore the number of extracted views per object had to be predefined.

In the following, we present an approach that can be applied to the output of different feature extraction methods. Our approach will identify stable views for the objects considering the Euclidian distance of the output from the used extraction method. In the following we will referr to these stable views as *keyframes*. Furthermore, our method exploits similarities among different objects. The number of keyframes per object does not have to be predefined. Rather, the accuracy can be defined with an overall maximum error, thus the system generates a different number of keyframes per object, depending on the object's appearance.

## II. System description

### A. Overview

Figure 1 gives a schematic overview of the system structure used throughout this paper. The system can be divided into a training part and a recognition part. In the application on a robot system, both parts have to be executed simultanously to allow the acquisition of new objects during interaction with the environment. The following sections will primarily focus on the training part, since the recognition part needs to rely on more than one modality as explained later (subsection II-E).

As mentioned earlier, the presented approach does not depend on a certain feature extraction method. The extraction method used should fulfill the following requirements:

- The extraction method has to capture the global appearance of an object.
- The extraction method should represent each view invariant to rotations in the viewing plane.
- The extracted feature vectors should be of reasonable size to allow fast extraction of keyframes.

For the experiments in this paper, we use color cooccurrence histograms (CCHs) to extract descriptors of the global appearance of views. The extraction of CCHs will be explained in subsection II-B.

During keyframe selection, significant views of the objects are identified by clustering the feature space into classes containing similar views. Each class is identified with its centroid, which is referred to as a keyframe.

Objects are assigned to keyframes in the labeling step. Each keyframe will be associated with all objects that have views in the corresponding class. Furthermore, we define the activation of a keyframe as the number of views an object participates with in the corresponding class. The keyframes are stored in the object database, together with the labels and activations determined in the labeling phase.

During recognition, the extracted features are classified using the stored keyframes from the object database. The classification will output all objects that have views similar to the current percept and the corrsponding activations.

The following subsections will explain the different elements of the system structure in detail.

### B. Feature Extraction

Throughout this paper, we will use color cooccurrence histograms (CCHs) for the description of object appearances. CCHs were chosen because they offer some properties which allow the application in real world recognition tasks. For instance CCHs offer a description of the object, which is invariant to the rotation in the viewing plane, when the parameters are chosen accordingly. Furthermore, CCHs offer some robustness towards scaling. Finally, CCHs combine texture information (in terms of information about pairs of neighbored pixels) as well as color information.

Based on work performed by Haralick et al. [10], CCHs were first introduced by Chang et al. [11]. In their work they define an entry in the color coocurrence histogram by the cooccurring colors and their distances in an observed image:

$$CH(c_1, c_2, \Delta x, \Delta y), \tag{1}$$

where $c_1$ and $c_2$ describe two colors in RGB space and $\Delta x$ and $\Delta y$ describe their distances in terms of pixels in the observed image. To achieve a rotation invariant description, only the absolute distance of the two colors is used in their approach:

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2} \tag{2}$$

The cooccurrence histogram is derived by counting all occurrences of entries $CH$ in the observed images.

In our implementation only cooccurences with a distance $d < 1.5$ are observed. This restricts the cooccurences to

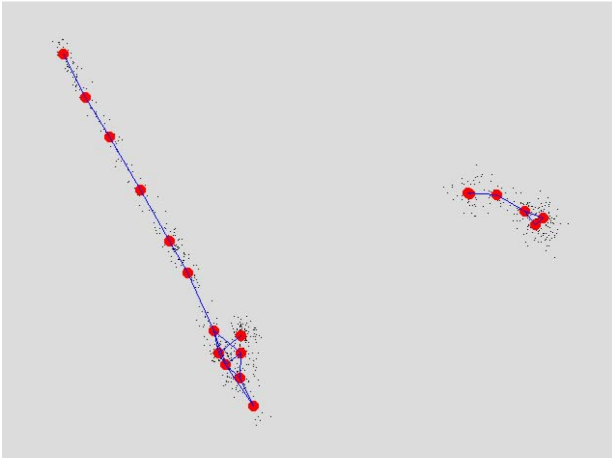


Fig. 2. Resulting growing neural gas network after 2000 iterations. 10 objects with 72 views each were used as input. The features and node positions were projected into 2D eigenspace.

neighboring pixels. Furthermore, the red and green color channels $(I_r, I_g)$ and the gradient magnitude of both color channels $(\nabla I_r, \nabla I_g)$ are used as histogram dimensions. The choice of these image descriptors is motivated by the previous works of Ekvall et al. [12]. They showed that with the combination of intensity and gradient descriptor calculated on the basis of the red and green channels good recognition results could be achieved. The cooccurrences in each channel are considered separately. Each channel is quantized to 80 clusters in a preprocessing step. This results in a feature vector of 320 dimensions, which is still of reasonable size.

### C. Keyframe Selection

The identification of similar views in the set of CCH features can be achieved by clustering the feature space into similar classes. Since the keyframe selection process will run autonomously, an unsupervised clustering method is required for our approach. Furthermore, the applied method should select the number of generated clusters dependent on the distribution of the input data rather than on a prespecified number of keyframes. One algorithm that fulfills these requirements is the Growing Neural Gas algorithm (GNG) which was first introduced by Fritzke [13]. The GNG is a self organizing map, which grows in the process of training according to the distribution of the input data. Thus the GNG algorithm creates a topological map which represents the distribution of the training data.

The GNG algorithm combines the Competitive Hebbian Learning and the Neural Gas method proposed by Martinez et al. [14] with an incremental learning approach. GNG thus overcomes the problem of prespecifying the number of nodes that is required to reach a certain goal. Heinke et al. [15] provided a comparison of different incremental neural network algorithms. Their comparison comprises Growing Cell Structures (GCS), Fuzzy Artmap (FAM), and Growing Neural Gas (GNG). As benchmark the performance of the multi-layered perceptron (MLP) was used. The GNG algorithm outperforms FAM on nearly all datasets and generates less nodes then GCS with similar performance for most datasets.

In the following, a brief introduction to the GNG algorithm is given to ease the understanding of the choice of certain parameters and the termination criterion. For a more detailed description of the algorithm the reader is referred to [13]. The network consists of the following components:

- A set of nodes $N$, each node $n \in N$ has an associated position vector $w_n$.
- A set of edges $E$, each edge $c \in E$ connects pairs of nodes and has an associated age.

The algorithm can be described with the following steps:

1) Create two nodes $n_1$ and $n_2$ with random positions $w_{n_1}$ and $w_{n_2}$.
2) Select one feature $f$ from the training set randomly.
3) Identify the nearest and second nearest nodes $n_a$ and $n_b$ to the feature $f$.
4) Increment the age of all edges starting from $n_a$.
5) Accumulate the error of node $n_a$ by the squared distance of node position $w_{n_a}$ and input signal $f$:
$$\Delta e_{n_a} = ||w_{n_a} - f||^2$$
6) Move the nearest node $n_a$ and the second nearest node $n_b$ towards the input signal $f$ using the learning rates $sp_a$ and $sp_b$:
$$\Delta w_{n_a} = sp_a(f - w_{n_a})$$
$$\Delta w_{n_b} = sp_b(f - w_{n_b})$$
7) Reset the age of the edge from the nearest to second nearest node $c_{n_a,n_b}$ to zero. If no edge exists, create a new edge.
8) Remove edges with an age larger than $a_{max}$.
9) If the accumulated error $e_{n_e}$ of one node $n_e$ exceeds the maximum error $e_{max}$ insert a new node in the following way:
   - Identify the node connected to $n_e$ with the maximum error $n_m$.
   - Insert a new node $n_c$ halfway between the two nodes $n_e$ and $n_m$:
   $$w_{n_c} = \frac{(w_{n_e} + w_{n_m})}{2}$$
   - Insert edges $c_{n_c,n_e}$ and $c_{n_c,n_m}$ and remove the edge $c_{n_e,n_m}$.
   - Set the accumulated error $e_{n_c}$ of the new node to the mean error of the nodes $n_e$ and $n_m$.
   - Decrease the accumulated error $e_{n_e}$ and $e_{n_m}$ by multiplication with a constant factor $\alpha < 1$.
10) Decrease all error variables by multiplication with a constant $\gamma < 1$:
$$e'_n = \gamma e_n$$
11) Check termination criterion. If not matched restart with step 2.

Depending on the termination criterion and the parameters used for training, the GNG algorithm will produce a topological map of the input data with respect to the distribution of the input data. Figure 2 shows an example outcome of the GNG clustering for 720 CCH features, which describe the rotations of 10 objects.

The parameters used for training the GNG were determined empirically. Aim of the parameter choice was a balance between stability of the network and fast convergence. Throughout the experiments a maximum edge age $a_{max} = 20$ was used. The learning rates were set to $sp_a = 0.16$ and $sp_b = 0.01$. The factors for the adjustment of the accumulated error in the case of a new node ($\alpha$) and for each iteration ($\gamma$) were set to $\alpha = 0.001$ and $\gamma = 0.995$.

The parameters for the maximum accumulated error per node $e_{max}$ and the termination criterion directly influence the number of nodes created for the input data. The choice of these parameters will be discussed in section III.

Each node from the network represents one cluster in the space of input features and is considered a keyframe.

### D. Labeling

The clustering results in a set of nodes $N = (n_1, \ldots, n_r)$. In order to use these nodes for object representation and recognition we have to restore the association of object views with the clusters formed by the nodes. In the following, $s$ objects $W = (F_1, \ldots, F_s)$ each described with $t$ features $F_x = (f_{x,1}, \ldots, f_{x,t})$ are considered.

To associate object labels with nodes all objects $x$ and their features $f_{x,v}$ are traversed. For each node $n_i$ the number of features of the object where the node is the nearest neighbor to the corresponding feature is determined as:

$$b_{i,x} = |\{f_{x,v} : i = argmin_{u \in \{1,\ldots,r\}} ||w_{n_u} - f_{x,v}||^2\}| \quad (3)$$

If $b_{i,x}$ is not zero, the object label $x$ is appended to the list of object labels $L_i$ for node $n_i$, if not already present:

$$L'_i = (L_i, x) \quad (4)$$

Additionally, the activation $a_{i,x}$ of the node $n_i$ for the object $x$ is calculated by the following equation:

$$a_{i,x} = \frac{b_{i,x}}{\sum_x b_{i,x}} \quad (5)$$

The activation describes how likely a feature which is associated to the node $n_i$ will belong to the object $x$. If $b_{i,x}$ is non-zero, the activation $a_{i,x}$ is appended to the list of activation $A_i$ of the node:

$$A'_i = (A_i, a_{i,x}) \quad (6)$$

It is guaranteed that for all labels of one object the sum of the corresponding activations is equal one, i.e.:

$$\sum_{x=1}^{s} a_{i,x} = 1 \quad (7)$$

This shows that if the activation for an object for the node equals 1, then the corresponding keyframe describes one object uniquely. The node will only contain one object label in this case.

In the object database, the node positions $w_{n_1}, \cdots, w_{n_r}$ are stored along with the associated labels $L_{n_1}, \cdots, L_{n_r}$ and activations $A_{n_1}, \cdots, A_{n_r}$.

### E. Classification

In the classification step, a perceived view of an object in terms of its CCH $f$ is matched with the keyframes stored in the object database. This can be accomplished by identification of the nearest neighbor $n_i$ in the set of keyframes:

$$i = argmin_{u \in \{1,\ldots,r\}} ||w_{n_u} - f||^2 \quad (8)$$

If the label list $L_i$ contains only one label, the corresponding object is found. Otherwise the classification can not be performed in a unique way. The list of labels $L_i$ contains objects that have views similar to the currently perceived view. The corresponding activations $A_i$ describe the probabilities for the individual objects.

In the case of multiple potential candidates for the current view, the feature extraction method used is not sufficient to separate between the objects in this class. In this case other modalities are required to uniquely detect the object corresponding to the perceived view. For this purpose our approach reduces the number of possibilities to similar objects in the modality observed and allows the restriction to only a few objects for the search in other modalities.

## III. Parameter Evaluation

For all experiments in this paper, object views from the Amsterdam Library of Object Images (ALOI) [16] are used. The ALOI contains images of objects on black background from 72 distinct viewing angles, which are generated by rotating the object around the vertical axis. We use 10 objects for the evaluation of our approach, which results in 720 CCHs.

As mentioned earlier, the maximum error $e_{max}$ and the termination criterion are crucial for the number of nodes that are generated by the GNG algorithm. In the following, our choice of these parameters is explained.

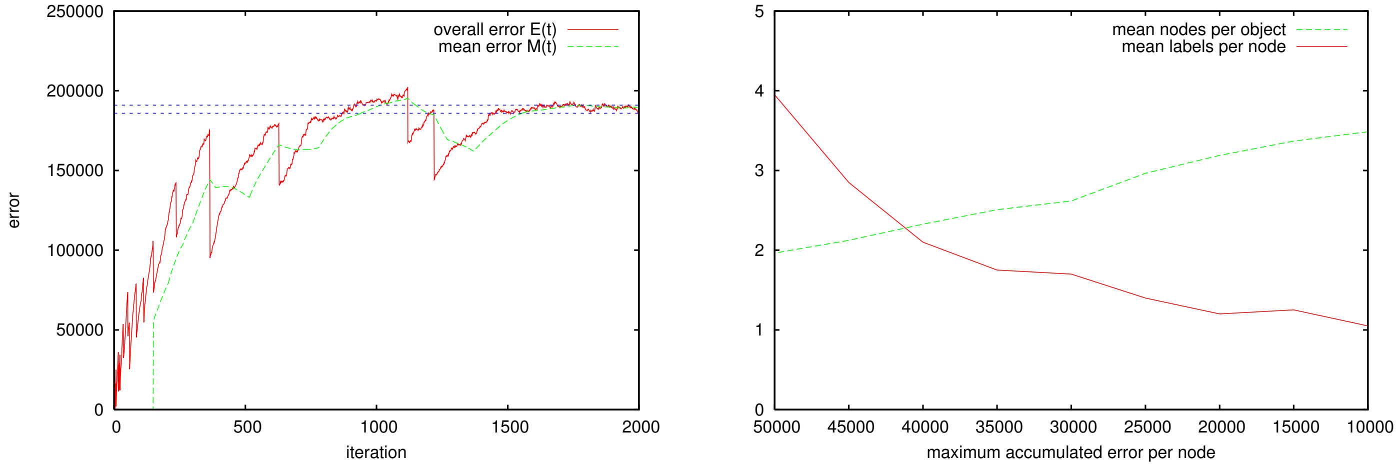
To verify if the network has converged, the overall error $E(t)$ of the network is monitored for each iteration $t$. The overall error can be determined by summing up the accumulated errors of all nodes:

$$E(t) = \sum_{x=1}^{r} e_{n_x}(t) \quad (9)$$

The overall error is smoothed by calculating the mean overall error $M(t)$ over the last 200 iterations. This helps in coping with local peaks in the course of the error over the iterations. To detect the convergence of the network, we check if $M(t)$ is in a defined range $r$ for a minimum number of iterations $\Delta t$. The termination criterion $c(t)$ is defined in the following way:

$$c(t) = \begin{cases} 0 & a \leq M(t - t_0) < b; 0 \leq t_0 < \Delta t; b - a < r \\ 1 & \text{otherwise} \end{cases}$$

We choose a range of $r = 5000$ and set the minimum time the mean overall error has to stay in this range to $\Delta t = 500$ iterations. Figure 3(a) shows the development of the overall network error $E(t)$ and the mean error $M(t)$ during one training phase. Every time the accumulated error




(a) Overall network error during one training phase. The upper and lower bounding from the termination criterion are denoted with horizontal lines.

(b) Development of the mean number of labels per node and the mean number of nodes per object dependent on the maximum accummulated error $e_{max}$.

Fig. 3. Results from the parameter evaluation

of one node exceeds $e_{max}$ the accumulated error is adjusted and a new node is inserted. This results in a diminution of the overall error. On new input data, the accumulated error of both nodes increases again. The overall error of a network containing more nodes can exceed the overall error of a network with less nodes because each single node can accumulate an error of up to $e_{max}$. The termination criterion terminates the training, if the error stays inside the range $r$ denoted by the two horizontal lines.

In order to determine the maximum node error $e_{max}$ for our experiments, two measures were observed using a range of $e_{max} \in [10000 : 50000]$. First the mean number of labels per node $\overline{L}$ was observed. Furthermore, the mean number of labels per object $\overline{I}$ was observed. In figure 3(b) both measures $\overline{L}$ and $\overline{I}$ are recorded. The graphs show that with a large $e_{max}$, the mean number of labels per node decreases fast. With decreasing $e_{max}$, the gradient of $\overline{L}$ reduces. The number of nodes per object $\overline{I}$ grows about linear with decreasing $e_{max}$. A suitable choice of $e_{max}$ should reduce the number of labels produced per node, since this decreases the uncertainty during recognition. Furthermore, not too many nodes per object should be generated, since the resulting representation has to be compact. For this reason, a maximum accumulated error of $e_{max} = 25000$ was chosen for our experiments. The choice of a maximum accumulated error less than 25000 would result in the generation of more nodes without significantly decreasing the number of labels per node.

## IV. Experimental Results

In our experiments, the GNG algorithm proved to be very stable. For the 10 objects with 720 views the number of nodes usually converged to 19. Depending on the sequence of the random selection of input data that was exposed to the network, occasionally 18 or 20 nodes were created.

Fig. 4. Important views of objects as extracted by our approach. Only views corresponding to an activation $a_{i,x}$ above 20% are shown.

### A. Keyframe Selection

The GNG network produces clusters of similar views and corresponding nodes with object labels $L_i$ and activations $A_i$. The node positions do not exactly correspond to object views. In order to visualize important views for the objects, the nearest neighbor from the set of samples for each object $x$ which is in the list of labels $L_i$ is identified. Views are reported only if the activation $a_{i,x}$ from the list of activations $A_i$ is above a given threshold. Thus, only those views are reported that are produced by clusters where the object participates with a significant amount of views.

Figure 4 shows the important views produced by our approach with a threshold of $a_{i,x} > 0.2$. The selected views depend on the used feature extraction method. Using a color descriptor like CCH results in the selection of views which are stable in terms of color.

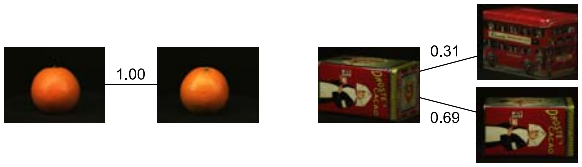



Fig. 5. During recognition, the orange on the left side can be identified uniquely. The can on the right side is associated to a keyframe with two labels. The connections are tagged with the corresponding activations $a_{i,x}$.

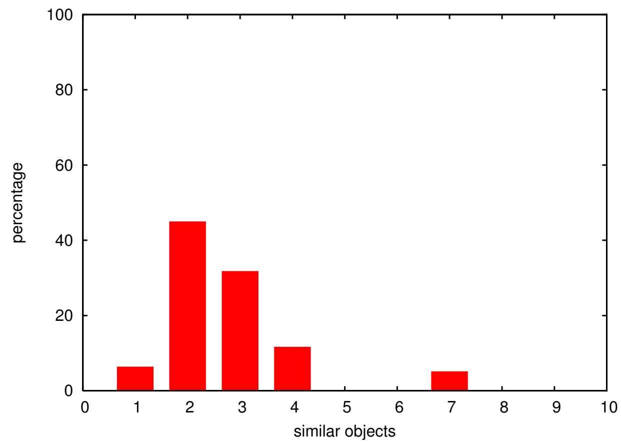



Fig. 6. Percentage of views of all 10 objects in relation to the number of similar objects associated.

### B. Recognition

In the recognition phase all object views are associated to the corresponding keyframes. Figure 5 shows two examples of associated views. In the first case, the view was associated to a keyframe which containes only one label. In the second case, the keyframe contained two labels. The keyframes are visualized with the corresponding closest views for each label contained in the label list. The connections are tagged with the activations $a_{i,x}$.

In order to provide a measure on how our approach reduces the uncertainty about the perceived object, we associate all object views to their keyframes. For each view the uncertainty can be expressed with the number of similar objects obtained from the label list. Figure 6 shows the percentage of views in relation to the number of similar objects. 6% of the object views are associated to keyframes which contain only one view and thus can be uniquely identified. 80% of the views are associated to keyframes which contain two or three labels. The remaining views are associated to keyframes with four and more views. The mean number of similar objects per view is about 2.7.

## V. Conclusion

The proposed approach allows for the extraction of keyframes on the basis of similarities among objects. For 10 objects with overall 720 views we were able to reduce the number of stored features for one modality to only 19. The experiments show, that with these 19 features, the potential candidates for a perceived object can be reduced to 2.7 on average.

An artificial perception system for a cognitive robot has to rely on more then one modality to identify and classify the manifold of different object types it encounters in real world tasks. The proposed approach will be used in conjunction with a combination of different descriptors for the object appearances. Despite the mentioned CCHs we plan to apply the same approach to other feature extraction methods eg. Zernike Moments. If chosen accordingly, the combination of different modalities will allow to identify the perceived objects uniquely.

Finally, the system will be implemented on our humanoid robot to ease the acquisition of objects during exploration of the environment.

## Acknowledgment

The work described in this paper was conducted within the EU Cognitive Systems project PACO-PLUS (FP6-2004-IST-4-027657) funded by the European Commission.

## References

[1] K. Welke, E. Oztop, A. Ude, R. Dillmann, and G. Cheng, “Learning features for an object recognition system,” in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Genova, Italy, 2006, pp. 290–295.

[2] H. Yamauchi, W. Saleem, S. Yoshizawa, Z. Karni, and H.-P. Seidel, “Towards stable and salient multi-view representation of 3d shapes,” in *Proceedings of the International Conference on Shape Modeling and Applications*, 2006.

[3] V. Lepetit, L. Vacchetti, D. Thalmann, and P. Fua, “Fully automated and stable registration for augmented reality applications,” in *Proceedings of the Second IEEE and ACM Symposium on Mixed and Augmented Reality*, 2003, pp. 93–102.

[4] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1999, pp. 1150–1157.

[5] K. Welke, P. Azad, and R. Dillmann, “Fast and robust feature-based recognition of multiple objects,” in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Genova, Italy, 2006, pp. 264–269.

[6] S. Palmer, E. Rosch, and P. Chase, “Canonical perspective and the perception of objects,” in *Attention and Performance IX*, J. Long and A. Baddeley, Eds. Hillsdale, NJ: Erlbaum, 1981.

[7] V. Blanz, M. J. Tarr, H. H. Bülthoff, and T. Vetter, “What object attributes determine canonical views?” Max Planck Institute for Biological Cybernetics, Tech. Rep. Technical Report No. 42, 1996.

[8] P. Hall and M. Owen, “Simple canonical views,” in *Proceedings of the british machine vision conference*, 2005, pp. 7–16.

[9] S.-H. Kim, I.-C. Kim, and I.-S. Kweon, “Probabilistic model-based object recognition using local zernike moments,” in *IAPR workshop on Machine Vision Applications*, Nara , Japan, Dec. 2002.

[10] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” in *IEEE Transactions on Systems, Man and Cybernetics*, 1973, pp. 610–621.

[11] P. Chang and J. Krumm, “Object recognition with color cooccurrence histogram,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1999.

[12] S. Ekvall and D. Kragic, “Receptive field cooccurrence histograms for object detection,” in *Proceedings of the IEEE/RSJ International Conference Intelligent Robots and Systems*, 2005.

[13] B. Fritzke, “A growing neural gas network learns topologies,” in *Advances in Neural Information Processing Systems*, vol. 7, 1995.

[14] T. Martinez and K. Schulten, “Topology representing networks,” in *Neural Networks*, vol. 7, 1994, pp. 507–522.

[15] D. Heinke and F. H. Hamker, “Comparing neural networks: A benchmark on growing neural gas, growing cell structures, and fuzzy artmap,” in *IEEE Transactions on Neural Networks*, vol. 9, 1998.

[16] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, “The Amsterdam library of object images,” *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005.