ART-Based Fusion of Multi-modal Information for Mobile Robots

Elmar Berghöfer, Denis Schulze, Marko Tscherepanow, Sven Wachsmuth

ART-based Fusion of Multi-Modal Information for Mobile Robots

Elmar Berghöfer[1], Denis Schulze[1,2], Marko Tscherepanow[1], and Sven Wachsmuth[1,2]

[1]Applied Informatics, Faculty of Technology
[2]CITEC, Cognitive Interaction Technology, Center of Excellence
Bielefeld University, Universitätsstraße 25, 33615 Bielefeld, Germany
{eberghoe,dschulze,marko,swachsmu}@techfak.uni-bielefeld.de

Abstract. Robots operating in complex environments shared with humans are confronted with numerous problems. One important problem is the identification of obstacles and interaction partners. In order to reach this goal, it can be beneficial to use data from multiple available sources, which need to be processed appropriately. Furthermore, such environments are not static. Therefore, the robot needs to learn novel objects. In this paper, we propose a method for learning and identifying obstacles based on multi-modal information. As this approach is based on Adaptive Resonance Theory networks, it is inherently capable of incremental online learning.

Keywords: sensor data fusion, incremental learning, Adaptive Resonance Theory

1 Introduction

Mobile robots moving side by side with humans in a common environment are confronted with different types of problems. One example of such a problem is a situation where obstacles are blocking the planed route. The decision how to handle such a situation depends on the type of the obstacle. In general such an obstacle can be any type of physical object. Some of these may be fixed, such as pillars, tables or cupboards. Others are movable such as wheelchairs. Humans can block the robots path as well, but as a special case of obstacle the robot could ask a human to move out of its way. Furthermore, identifying humans as possible interaction partners would be of interest in most human robot interaction scenarios. An example of an environment for a mobile robot could be an office building or a hospital in which the robot has to deliver different things. The robot then has the possibility to interact with, manipulate (move), or circumnavigate objects and persons. In order to solve these tasks, the robot has to identify the type of occurring obstacles.

Robot systems, which can be applied in such scenarios, usually possess multiple sensor modalities, e.g. [16]. These sensor data need to be fused to take advantage of all of them so that the robot can act appropriately. Furthermore,

we assume that the environment may change; for instance, new obstacles or persons might appear. In this case, the robot should learn these objects. Therefore, this paper focuses on methods for the incremental learning of objects based on data from different sensors.

In Section 2, we discuss related work on this topic. Afterwards, our approach is presented in Section 3. In Section 4, we evaluate this approach based on data originating from a real robotic system. Finally, we summarise our most important results in Section 5 and give an outlook on possible future work.

2 Related Work

Established information fusion architectures applied in robotics [7, 9] resort to predefined rules to find corresponding object representations in multiple sensor modalities. On the one hand, these approaches have the advantage that the results of different sensors (e.g., a laser scanner or a face detector, [7]) or different sub-architectures (e.g., speech recognition system and visual tracking system, [9]) can be used to integrate information for higher level processing systems like planners. On the other hand, it is often necessary to define new rules by hand, e.g. if a new sensor or new subsystem is integrated into the system.

In the scenario outlined in the introduction, a system is required that is able to automatically learn the correspondence between different sensor representations of a specific object. From the literature, several machine learning approaches are known that can be employed to perform this kind of sensor data fusion. For example, in [6] a time-delayed neural network (TDNN) is applied in an automatic lipreading system to fuse audio and visual data. In [11], another TDNN is applied to visual and audio data to detect when and where a person is speaking in a scene. A major drawback of these networks is the problem of catastrophic forgetting; i.e., learned associations from input data to output classes could be adversely influenced if the network trained online.

The majority of existing neural network architectures suffers from the problem that increasing stability causes a decrease of plasticity and vice versa, which is summarised in the so-called *stability-plasticity dilemma* [3]. In order to prevent these problems, neural networks based on the *Adaptive Resonance Theory* (ART) were developed. A first network realising this idea was published in 1987 [3]. It is usually referred to as ART1 and limited to unsupervised learning of binary data. Afterwards, a multitude of ART-based networks possessing different properties and application fields have been developed. In addition to unsupervised ART networks [1, 5, 12], supervised ARTMAP networks were introduced [4, 14]. ART-based approaches have already been used for information fusion; for example, a sensor fusion approach using different sensors for distance measurement on a mobile B14 robot was proposed in [10].

The approach proposed in this paper was intended to connect several advantages of the above-mentioned concepts. Therefore, we developed a new ARTMAP network optimised for classification tasks based on multi-modal information. This novel ARTMAP network can learn the dependencies between different

sensory representations of objects and retains the advantages of the already published ART approaches.

3 Our Approach

The following section deals with the theoretical background and a detailed description of our approach. Therefore, we firstly introduce some basic notations and concepts that are necessary for the understanding of the proposed simplified fusion ARTMAP (SiFuAM).

3.1 Basic Principles of ART Networks

The activation of ART networks is computed by means of the comparison of a bottom-up input vector and a top-down prototype. This prototype is represented by the weight vectors of the neurons in the output layer $F2$ of the network and for the most ART systems it can be interpreted as a region in the input space. An ART system learns a clustering of the input data. To achieve this, changes caused by new input vectors are restricted to similar prototypes and new input vectors that are too different will cause the net to create a new neuron representing a new cluster. As a result, vectors of low populated areas of the input space can be represented immediately, which makes the net flexible. While the learning rule guarantees that the regions only changes so that a data point in the input space that was covered once will not be excluded again. These properties of the ART architecture, render it capable of stable and plastic incremental learning.

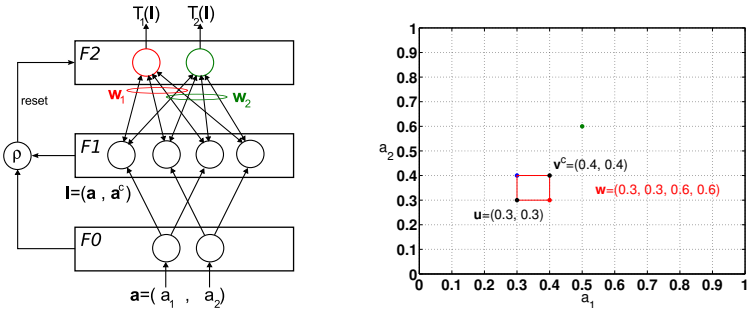


Fig. 1. Fuzzy ART architecture and functioning. The left subfigure shows the exemplary structure of a Fuzzy ART network for two-dimensional input vectors $\boldsymbol{a}$ that has learned two categories. The $F1$ layer performs the complement coding of the input vector. With complement coding, the weight vectors $\boldsymbol{w}_j$ can be interpreted as rectangular regions in the input space (right).

For a basic understanding of the later described architectures, a description of Fuzzy ART [5] is required. The Fuzzy ART architecture is visualised in Fig. 1.

The first layer of the network generates a so-called *complement coded* vector $\boldsymbol{I}$ from the input $\boldsymbol{a}$. This step is a normalisation process to prevent proliferation of generated clusters. $\boldsymbol{I}$ is defined as $\boldsymbol{I} = (\boldsymbol{a}, \boldsymbol{a}^c)$ and the elements of $\boldsymbol{a}^c$ are calculated by $a_i^c = 1 - a_i$. Each $F2$ neuron j represents one cluster and its activation $T_j(\boldsymbol{I})$ (note: j is used for indexing the output layer neurons, and J will be the index of the neuron with the highest activation) is given by:

$$T_j(\boldsymbol{I}) = \frac{|\boldsymbol{I} \wedge \boldsymbol{w}_j|}{\alpha + |\boldsymbol{w}_j|},\ 0 < \alpha \ll 1,\ j = 0 \ldots N\ . \tag{1}$$

Where N is the number of $F2$ neurons, α is used to privilege small regions and $\wedge$ denotes the fuzzy AND operator: $x \wedge y = min(x, y)$ (used element by element on a vector). The applied vector norm $|\cdot|$ is the L1 Norm. After the best matching node J has been determined, its weight vector $\boldsymbol{w}_J$ will be used to calculate a matching value (2) for the represented category. The matching value will be compared to a value p called vigilance, which is a parameter controlling the generalisation of the net. As Fuzzy ART is using the complement coding, the vigilance defines also the maximum size of the hyper-rectangular regions.

$$\underbrace{\frac{|\boldsymbol{I} \wedge \boldsymbol{w}_J|}{|\boldsymbol{I}|}}_{\text{matching value}} \geq p,\ p \in [0, 1]\ . \tag{2}$$

If the vigilance criterion (2) is not fulfilled, the winner neuron will be reset (blocked for the current input), and the next best matching neuron will be determined. Otherwise, the net reaches resonance and the index of the winner neuron can be interpreted as the label of the cluster in which the input was categorised. The net learns new input by modifying the weight vector $\boldsymbol{w}_J$ of the winner neuron to represent the new information as given by:

$$\boldsymbol{w}_J^{new} = \beta(\boldsymbol{I} \wedge \boldsymbol{w}_J^{old}) + (1 - \beta)\boldsymbol{w}_J^{old},\ \beta \in [0, 1]\ . \tag{3}$$

The parameter β defines the learning rate. The special case $\beta = 1$ is called fast-learning and causes the region to include the new point after a single learning step. For β<1 the net becomes more insensitive to noise, but requires more input.

Another possibility is that the net can not find any neuron to be in resonance with the current input $\boldsymbol{I}$. In this case an uncommitted neuron will be selected, and its weight vector is set to $\boldsymbol{I}$. Therefore, a new input which lies in a region of the input space that is not covered by - or close enough to - an existing region will generate a new category.

3.2 Simplified Fuzzy ARTMAP (SFAM)

The SFAM architecture described in [14] is an extension of the FuzzyART architecture making it usable for supervised learning. A Fuzzy ART network is used as a part of the SFAM architecture (see Fig. 2), but the neurons in the $F2$ layer are extended with an associated class label. The training is supervised by the

match-tracking algorithm. In contrast to Fuzzy ART, SFAM receives a sequence of pairs of an input vector $\boldsymbol{a}$ and an associated correct class label b. The input $\boldsymbol{a}$ is presented to the internal Fuzzy ART, then the class label of the winner neuron will be compared to the given class label b. Depending on matching or not, the vigilance of the Fuzzy ART may be raised temporarily to force the selection of another neuron. If necessary a new neuron will be committed.

A trained SFAM system can be used for class prediction of a given input $\boldsymbol{a}$. In this case, b is interpreted as the output.

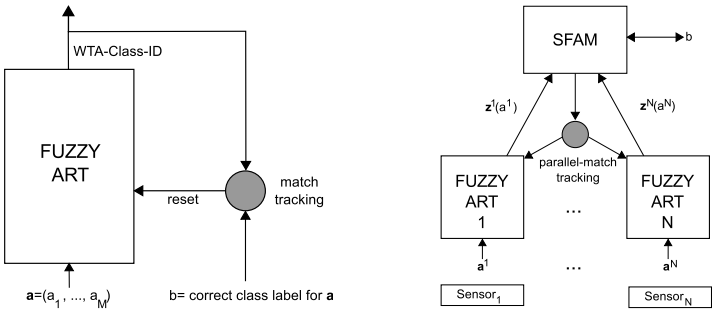


Fig. 2. SiFuAM in comparison to SFAM. An SFAM network (left) is trained with M-dimensional input vectors $\boldsymbol{a}$ and corresponding class labels b. In contrast, SiFuAM networks (right) receive N input vectors $\boldsymbol{a}^i$ from different sensor channels. These vectors are passed to individual ART modules. Then, a vector $\boldsymbol{z}$ comprising the concatenated output vectors of the ART modules is propagated to a common SFAM network which learns the corresponding class label b.

3.3 Simplified Fusion ARTMAP (SiFuAM)

Our approach is based on the Fusion ARTMAP architecture [2, 8]. Fusion ARTMAP has the benefit of reflecting the influence of single sensor channels on the classification. Because of our object identification scenario, we developed a simplified version, which we call the "Simplified Fusion ARTMAP" (SiFuAM). The design of the SiFuAM architecture is shown in Fig. 2. It consists of one Fuzzy ART module per input channel and a superior SFAM module. Due to the use of Fuzzy ART, the input vectors $\boldsymbol{a}^i$ of each channel were grouped into categories. A modified weight vector of the category will be used to create the input vector for the SFAM, while the input itself can be generated by different sensors or just different features calculated on the data from one physical sensor.

During learning, the SFAM network receives the correct class label b in addition to the actual input. A teaching input consists of $i = 1 \ldots N$ feature input vectors $\boldsymbol{a}^i$, and the target class label b. At the first step each ART^i module tries to assign its input to a known category. If that fails the ART module has to

create a new category with $\boldsymbol{a}^i$ as its initial weight vector. When all ART networks have categorised their input, a vector $\boldsymbol{z}$ will be created from the weight vectors of the winner neurons. It is important to notice that the ART modules are not allowed to do a training step with their inputs yet. The input vector for the SFAM module is given by: $\boldsymbol{z}_c = (\boldsymbol{z}^1, \dots \boldsymbol{z}^N)$, where the $\boldsymbol{z}^i$ are generated by:

$$\boldsymbol{z}^i = \beta(\boldsymbol{I}^i \wedge \boldsymbol{w}_J^i) + (1-\beta)\boldsymbol{w}_J^i,\ \beta \in [0,1]\ . \tag{4}$$

Here, $\boldsymbol{w}_J^i$ represents the weight vector of the winner neuron of the i'th Fuzzy ART module. The vector $\boldsymbol{z}^i$ of a Fuzzy ART represents $\boldsymbol{w}_j^i$ as if it was already trained with $\boldsymbol{I}^i$. If the SFAM categorises the concatenated vector $\boldsymbol{z}_c$ into a category whose class label matches b then all Fuzzy ART modules and the SFAM are doing a training step. If not, the so-called *parallel match-tracking* algorithm is activated, which searches for the least confidential Fuzzy ART module (ART_{lc}), i.e., the one with the lowest matching value (2). Then the vigilance of all Fuzzy ART modules and the SFAM will be raised just enough so that the ART_{lc} resets the winner neuron. In doing so the least confidential channel will be blamed for the misclassification. Hence, not the whole network has to change but only the part which is most likely the reason for the mistake. The ART_{lc} will choose another category and, therefore, another weight vector which leads also to a changed vector $\boldsymbol{z}_c$. This will be repeated until the SFAM classifies the input correctly. If all Fuzzy ART modules need to create a new category the SFAM has to create a new category as well which is labelled with b.

If the trained SiFuAM network is used for class prediction, b is the output.

4 Evaluation

Our approach was evaluated based on data originating from the Bielefeld Robot Companion (BIRON) [16]. For testing the learning system, a dataset from the data streams of two sensors was recorded: a colour camera ($1,600\times1,200$ pixels, approx. 120cm above the floor) and a laser range finder (LRF) (approx. 20cm above the floor) providing a 180° laser scan with 360 data points. The system should learn to identify and to distinguish persons from immobile, non-interactive obstacles, in particular pillars.

4.1 Data Collection

In order to render the output of both sensors compatible, only LRF data lying in the view angle (86°) of the camera were considered. From the camera image, two independent feature vectors were generated: a "face feature vector" (FFV) and a "structure feature vector" (SFV) reflecting the occurrence of vertical objects.

The FFV describes the fraction by which each pixel column of the camera image is covered by a face. A 1-dimensional Gaussian mask with 161 elements is used to calculate 21 weighted average values to reduce the dimension. The face

hypotheses, their position, and size, were calculated by a face detector from the OpenCV[1] library based on [15].

The SFV is computed by means of a morphological opening operator with a structuring element of 1/3 of the image height and a width of one pixel. The resulting image is subtracted from the original image to remove all structures which do not have a high vertical dimension. Then, the structuring element is used in horizontal alignment for a morphological closing operation to remove all wide structures. Finally, a threshold is applied resulting in a new black and white image. This threshold is chosen such that all values smaller than 20% of the maximal value are suppressed. The final 21-dimensional SFV is calculated similar to the FFV.

For the LRF data, only an averaging is made by a discrete 1-dimensional Gaussian mask of 3 elements, so that 1 value per degree was calculated (originally an element per 0.5 degree). Hence, the LRF vector is reduced to 86 elements, each of this corresponding to one degree of the region covered by the camera (1° LRF represents 80 pixels of image width).

Following the assumption that not all of the information in the picture is required to identify an object, a sliding window is used to perform a search over a picture. Therefore, the picture is split into 17 overlapping slices of a width of 320 pixels with an offset of 80 pixels. Each slice is then represented by 5 values from the LRV and SFV as well as 18 values from the LRF vector corresponding to this image window.

The recorded dataset consist of 167 samples from 4 different persons and 200 samples from 2 different pillars. The centre of the person or the pillar was marked manually for each sample. All windows containing at least 50% of an object are labelled with the corresponding class label, 1 for humans and 2 for pillars. The slices which contain no object are marked with class label 0 for background, and are used as negative examples.

4.2 Results

The evaluation is done by a cross validation on the dataset, therefore the data elements from one person and one pillar are excluded for test and the rest were used for training set. So 8 different combinations of test and training sets were generated and the average test error was calculated. This was repeated for all combinations of the net parameters β in the range $[0.5, 1]$ and p in the range $[0, 0.99]$ ($p = 1$ results in a network just memorizing the input). Since the system is meant to be used in an online (sequential) learning scenario, each element from the training set was presented one at a time and only once.

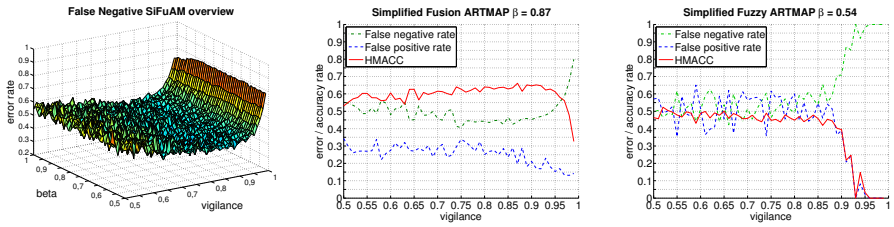
For the interpretation, the classification performance of the two networks mainly two error rates are of interest, the false negative rate (FNR) and the false positive rate (FPR). The FNR accumulates the errors, where an object (person or pillar) was not detected, while the FPR accumulates the false positive detection in the background. Our net should not learn the background class

[1] Version 2.1, http://opencv.willowgarage.com

as an object. Therefore, a rejection of background slices was not counted as an error. Minimizing the FPR could be done easily by rejecting every input, which certainly would be disadvantageous. Trying to reduce the FNR by finding objects everywhere is also an unwanted scenario. To optimize the classification result, the FNR and the FPR should be minimal at the same time. We use the harmonic mean accuracy (HMACC) because it has higher values where both errors have small values and it penalises big differences between them. Due to this the maximum of the HMACC is a good parameter choice. Its use for error analysis is also shown by Tscherepanow et al. in [13], and it is given by:

$$HMACC(p) = \frac{2}{\frac{1}{1-FNR(p)} + \frac{1}{1-FPR(p)}} \ . \tag{5}$$

As can be seen in Fig. 3(a), which shows the FNR for the SiFuAM respecting the different β and vigilance values, the change of β has a minor effect to the classification for our dataset. Therefore, the two plots on the right show the error rates only for the optimal values[2] of β of the SiFuAM and the SFAM. The third graph represents the HMACC.



(a) (b) (c)

Fig. 3. Error Plots. The plot (a) exemplary shows the false negative rate of SiFuAM networks for all values of β. The other two plots show the error rates and the accuracy for SiFuAM (b) and SFAM (c) according to their best false negative value of β.

In Fig. 3, the HMACC value is plotted for the SiFuAM and SFAM where the SFAM has its best accuracy at vigilance $p = 0.52$ reaching a value of 0.52 and the SiFuAM has the best result at $p = 0.87$ reaching accuracy of 0.66. Also the fact that the SiFuAMs FPR curve has better values at high HMACCs has to be emphasised, because due to the use of the sliding window approach with the chosen values for window width and offset results in a higher number of windows representing background. Hence the absolute amount of possibilities to do a false positive prediction are twice as hight as those of performing a false negative. To illustrate which errors occur and in which quantity the following Table 1 shows the confusion matrices for the SiFuAM and SFAM with the best parameter values of each net.

[2] averaged over all values of p

Table 1. Confusion Matrices of SiFuAM ($\beta = 0.6,\ p = 0.87$) and SFAM ($\beta = 0.54,\ p = 0.52$). The percentage values are rounded. All rejections are counted as background predictions (background was not an object type to be learned).

correct \ prediction		**background**	**person**	**pillar**
SiFuAM	background	**11618 (79%)**	778 (5%)	2386 (16%)
	person	309 (23%)	**967 (73%)**	44 (3%)
	pillar	1443 (45%)	148 (5%)	**1585 (50%)**
SFAM	background	**7584 (51%)**	834 (6%)	6364 (43%)
	person	556 (42%)	**652 (49%)**	112 (8%)
	pillar	1628 (51%)	174 (5%)	**1374 (43%)**

The results show that the pillar class is more difficult to separate from the background than the person class. In general, SiFuAM predicts the correct class labels more often than SFAM. In particular, SiFuAM predicts less pillars and persons in background areas (false positive). Nevertheless, pillars are frequently considered as background by both types of networks. A reason for this can be that several sample images contain a bright background light caused by a large window close to the considered pillars, which compromised the SFV. In contrast, the pillar and person classes are better separated and only rarely mixed up.

5 Conclusion and Future Work

As shown in the previous section, the SiFuAM is able to learn a classification on a small set of data coming from real sensors of a mobile robot, even if they are partly very noisy. The SFAM which just uses a concatenated vector of all sensor data was outperformed by the SiFuAM especially in the FPR, which makes it more likely that considering effects of single sensor channels for learning is a good idea. Also analysing the weights after several training steps will give information of useless data channels that may be removed.

The classification was only done on sensor data snapshots. A future goal is, to use the SiFuAM directly embedded on a mobile robot system, where the sensor data are read out continuously. The use of time sequential information can increase the overall classification rate dramatically. For example a pillar is detected correctly in nearly every second image and has a low rate of false positives, hence analysing a short sequence of data can be used to generate hypothesis for pillar objects with a high reliability. Also better feature values for the LRF which are independent of the absolute distance of an object will be an advantage. Furthermore the ability of the net to incrementally learn online, can be used to learn new examples at any time labelled by a human tutor via human robot interaction.

Acknowledgements. This work was partially funded by the German Research Foundation (DFG) and Excellence Cluster 277 “Cognitive Interaction Technology”.

References

1. Anagnostopoulos, G.C., Georgiopoulos, M.: Hypersphere ART and ARTMAP for unsupervised and supervised incremental learning. In: Proceedings of the International Joint Conference on Neural Networks. vol. 6, pp. 59–64 (2000)
2. Asfour, Y., Carpenter, G., Grossberg, S., Lesher, G.: Fusion ARTMAP: An adaptive fuzzy network for multi-channel classification. In: Proceedings of the International Conference on Industrial Fuzzy Control and Intelligent Systems. pp. 155–160 (1993)
3. Carpenter, G.A., Grossberg, S.: A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision, Graphics, and Image Processing 37(1), 54–115 (1987)
4. Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., Rosen, D.B.: Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Transactions on Neural Networks 3(5), 698–713 (1992)
5. Carpenter, G.A., Grossberg, S., Rosen, D.B.: Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks 4, 759–771 (1991)
6. Cutler, R., Davis, L.: Look who's talking: speaker detection using video and audio correlation. In: Proceedings of the International Conference on Multimedia and Expo. pp. 1589–1592. IEEE (2000)
7. Fritsch, J., Kleinehagenbrock, M., Lang, S., Plötz, T., Fink, G.A., Sagerer, G.: Multi-modal anchoring for human-robot interaction. Robotics and Autonomous Systems 43(2–3), 133–147 (2003)
8. Harrison, R., Borges, J.: Fusion ARTMAP: Clarification, implementation and developments. Tech. Rep. 589, The University of Sheffield, Department of Automatic Control and Systems Engineering, Mappin Street, Sheffield S1 3JD (1995)
9. Jacobsson, H., Hawes, N., Kruijff, G.J., Wyatt, J.: Crossmodal content binding in information-processing architectures. In: Proceedings of the International Conference on Human Robot Interaction. pp. 81–88. ACM (2008)
10. Martens, S., Gaudiano, P., Carpenter, G.: Mobile robot sensor integration with fuzzy ARTMAP. In: Proceedings of the International Symposium on Intelligent Control. pp. 307–312 (1998)
11. Stork, D., Wolff, G., Levine, E.: Neural network lipreading system for improved speech recognition. In: Proceedings of the International Joint Conference on Neural Networks. pp. 289–295. IEEE (1992)
12. Tscherepanow, M.: TopoART: A topology learning hierarchical ART network. In: Proceedings of the International Conference on Artificial Neural Networks. LNCS, vol. 6354, pp. 157–167. Springer (2010)
13. Tscherepanow, M., Jensen, N., Kummert, F.: An incremental approach to automated protein localisation. BMC Bioinformatics 9(445) (2008)
14. Vakil-Baghmisheh, M.T., Pavešić, N.: A fast simplified fuzzy ARTMAP network. Neural Processing Letters 17, 273–316 (2003)
15. Viola, P., Jones, M.: Robust real-time object detection. In: Second International Workshop on Statistical an Computational Theories of Vision (2001)
16. Wachsmuth, S., Siepmann, F., Schulze, D., Swadzba, A.: ToBI – Team of Bielefeld: The human-robot interaction system for RoboCup@Home 2010. Tech. rep., Bielefeld University, Applied Informatics (2010), `http://aiweb.techfak.uni-bielefeld.de/files/Team_ToBI_TDP_2010_0.pdf`