2015 IEEE International Conference on Systems, Man, and Cybernetics

Procedural Memory Learning from Demonstration for Task Performance

Yong-Ho Yoo
School of Electrical Engineering
KAIST
Daejeon, Republic of Korea
e-mail: yhyoo@rit.kaist.ac.kr

Jong-Hwan Kim
School of Electrical Engineering
KAIST
Daejeon, Republic of Korea
e-mail: johkim@rit.kaist.ac.kr

***Abstract*—A robot is expected to carry out a task autonomously with its own knowledge system. Using the knowledge system, the robot can recognize current situation and recall a proper sequence for performing an appropriate task in that situation. To build such knowledge system, the robot learns the knowledge from user demonstrations as if a child learns through interactions with parents and teachers. User demonstration is captured by an RGB-D camera embedded the robot. The robot needs to segment each execution from continuous RGB-D streams. In this paper, each execution is composed of an object and an action performed on the object. The sequence of executions, or the procedure, should be stored in the robot's memory for the the robot to retrieve and execute the procedure in a similar situation later. Such a procedural memory is developed based on an adaptive resonance system. Using the procedural memory learned, the robot can perform the full sequences of tasks with only partial information given on executions. The effectiveness of the proposed scheme is demonstrated for four tasks through computer simulations.**

***Index Terms*—Learning from demonstration, procedural memory, robot learning.**

I. Introduction

One of the goals of developing a robot is to enable the robot interact with humans to carry out a task. To achieve the goal, the robot should be able to understand current situation and select a proper action depending on the current situation and human's requirements [1]. The difficulty, however, lies in the fact that the robot cannot understand the situation without a knowledge system. Since humans have the knowledge systems like a long-term memory, humans can understand current situation easily so that humans take an appropriate action referring to the knowledge system. Even children can organize and accumulate their own knowledge as instructors teach their knowledge and experiences. Inspired by this fact, this paper investigates a scheme on learning from the instructors' demonstration for robots to build their own knowledge system, etc.

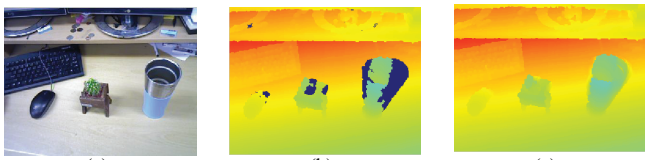
Learning from demonstration is known as robot learning or imitation learning, which makes the robot perform new tasks autonomously [2]–[6]. Manual programming all primitives required for completing a given task is possible but it needs an extended effort of an expert. Also, the engineer must provide control rules for each joint's motion of the robot in the learning phase. In contrast, learning from demonstration allows the robot to learn how to move its body simply by watching and storing the sequence of actions the instructor performed. This learning scheme is an imitation of humans using their own visual systems to watch and follow the instructors actions to perform a task [7].

There are studies for detecting a current situation using the relation between objects and actions [8]–[10]. These studies reveal that it is indeed possible to treat objects and actions as conjoint entities suggested by object-action complexes. This method categorizes the execution composed of an object and an action performed on the object that is learned from a user demonstration. For learning from demonstration, however, the robot should be able to segment each execution from continuous RGB-D streams.

It is not sufficient to detect an execution from continuous RGB-D streams for performing the sequence of executions. As a specific task is given to humans, they can recall a proper sequence for performing the task from previous experiences and follow the procedure step by step. For the robot to cooperate with humans, the robot needs a memorization ability equal to that of a human. In other words, the robot should memorize and recall a sequence of executions in the given task. There are many studies discussing what models are proper for memorization of a sequence of executions [11]–[13]. These models address how to incorporate both an episodic memory and a semantic knowledge. However, most of works have the limitation on capturing executions and relations between executions under a complex situation and on accumulating knowledge. Even some previously applied models, e.g. a neural network model can disrupt the existing knowledge system when new data is fed into this model. This problem is called the 'plasticity-stability' dilemma. The limit can be overcome by using an Adaptive Resonance Theory (ART) model [14]. Even an EM-ART model, stacked two fusion ART model, is able to memorize spatial data as an event and s sequence of events as an episode [14], [15]. The performance of the EM-ART were conducted by a first-person shooting game, as well as a word recognition benchmark test. The evaluation set, however, was generated virtually in a simulation so it is hard to apply directly in a real environment.

Inspired by the episodic memory design, this paper proposes a scheme on how to learn a procedural memory in a real environment, which is captured by the RGB-D camera. The

978-1-4799-8697-2/15 $31.00 © 2015 IEEE
DOI 10.1109/SMC.2015.426



(a) (b) (c)

Fig. 1. The filtered result using a cross-bilateral filter. (a) Input RGB. (b) Input depth. (c) In-painted depth.

procedural memory, a sequence of executions, is learned from visual demonstration based on the EM-ART. For segmenting continuous RGB-D streams into an execution, the robot should understand what objects are present and what action is executed. To boost accuracy of the object and action recognition, pixels that correspond to a plane that supports objects are removed using Random Sample Consensus (RANSAC) [17] to facilitate constructing object hypotheses. These object hypotheses can be classified using a SIFT descriptor. To recognize an action, one hand's speed estimated by two consecutive frames is concatenated by the SIFT descriptor. After segmenting the continuous RGB-D streams into each execution, this execution is an input to the EM-ART model for constructing a procedural memory. In a real environment, some executions are incorrectly encoded due to noise present. With an assumption that the incorrect recognitions are relatively rare, the voting algorithm is applied to improve the result.

The remainder of this paper is organized as follows: Section II describes a perception process. In Section III, the procedural memory learning method is proposed. Experimental results are presented in Section IV. Finally, concluding remarks follow in Section V.

II. Perception

There are four steps in our perception system. At first, a main plane information is extracted by the RANSAC algorithm using a depth image. In the second step, object hypotheses are found by removing pixels that correspond to a plane. After that, features for each object hypothesis are extracted, which are in turn classified using Support Vector Machine (SVM).

A. Segmentation

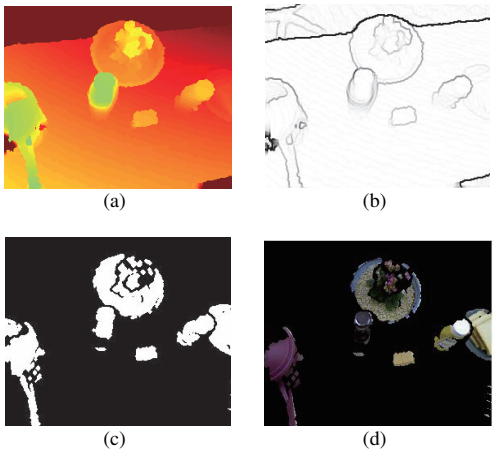
As a specific task is given to a person, s/he finds some objects that are needed to execute a given task. In an indoor environment, many objects are located on the stable plane such as table or desk. This fact can be applied to robots as well. In images given to a robot, a searching space can be limited to pixels on the plane to increase recognition speed. 3D point clouds, which can be easily derived by the RGB-D camera with a calibration, are needed to estimate a plane in a image.

Sometimes, depth information itself has noises on the edges of objects. In these regions, depth information cannot be estimated, thus leaving holes in the depth map. Before processing the segmentation, these noises must be removed. To remove these artifacts, a cross-bilateral filter is applied [18]. This filter is a nonlinear filter that smoothes a signal while preserving strong edges. The filtered result is shown in Fig. 1.

On next step, the main plane such as table, desk, and ground that supports objects of interest is removed from the image. In 3D point clouds obtained by the RGB-D camera, there are both pixels that lie within the plane and outlier pixels that lie far from the estimated plane. To estimate plane parameters, RANSAC algorithm is applied. The depth image used in this paper has 240x320 pixels so there are many 3D point clouds to estimate a plane equation. To speed up this RANSAC, 3D point clouds sampled in depth images by shifting 8 pixels are used. Whether each depth pixel belongs to object hypotheses region or not is determined by the distance between 3D point clouds and the plane estimated by RANSAC. This procedure is described in Algorithm 1.

B. Feature Extraction for Object and Action Recognition

To recognize what object is in each bounding box, feature extraction is followed by learning. In this paper, a SIFT descriptor is chosen as a good solution for objects with a large amount of details. To improve the feature extraction speed, all bounding boxes are reshaped by 32x32 size and a center point of each bounding box is fixed as a key-point. A SIFT descriptor is a 3-D spatial histogram of the image gradients. Each bounding box is divide to 4x4 bins and each pixel in each bin has an orientation quantized into 8 directions. For each pixel in each bin, orientations are quantized into 8 directions. Then, an additional Gaussian weighting function is applied to give more weights to the pixels located near the key-point. This 3-D histogram consisting of 8x4x4 bins is stacked as a single 128-dimensional feature. Object learning and recognition are based on a multi-class Support Vector Machine (SVM) for classification. To formulate object learning and recognition, denote the SIFT descriptor as $\{x_1, \ldots, x_N\}$ where N is the number of training features and the corresponding class label as $y_i \in \{1, 2, \ldots, M\}$ where M is the number of total classes.



(a) (b) (c) (d)

Fig. 2. A procedure of proposed perception system. (a) A depth image. (b) A depth gradient image. (c) A plane-filtered image. (d) Object hypotheses region.

Algorithm 1 Object region estimation using RANSAC

INPUT: 3D point clouds D
OUTPUT: Binary image P
Binary image means whether each pixel belong to object region or not.
Get 3D point clouds D using depth image
for $i = 0 \rightarrow$ max iterations **do**
  $S_i \Longleftarrow$ Random Sample(D) ▷ To estimate a plane, at least 3 points are needed.
  $M_i \Longleftarrow$ Compute Model Parameters(S_i)
  $\{e, D_{in}, D_{out}\} \Longleftarrow$ VerifyingModelParameter(D, M_i)
  if $e_i < e_{min}$ **then**
    $e_{min} \Longleftarrow e_i$
    $M \Longleftarrow$ ComputeModelParameters(D_{inlier})
    if $e_{min} < \epsilon$ **then**
      Return M_i
    end if
  end if
end for
Parameters are determined and discriminate points on the plane.
$P^* \Longleftarrow$ AbovePlane(D)
▷ Object region is restricted on the plane.

M binary SVM classifiers separate one class from all the rest. The i-th linear SVM is trained with all the training examples of the i-th class with positive labels, and all the others with negative labels. This SVM solves the following problem that yields the i-th decision function $f_i(x) = w_i^T x + b_i$.

$$\begin{array}{ll} \min\limits_{w_i,\xi_i} & \frac{1}{2}\|w_i\|^2 + C\sum_{j=1}^{N}\xi_i^j \\ \text{s.t.} & z_j(w_i^T x_j + b_i) \geq 1 - \xi_j^i, \quad \xi_j^i \geq 0 \quad \forall j \end{array} \tag{1}$$

where C is a tuning parameter to balance the margin and the training error and $z_j = 1$ if $y_j = i$ and $z_j = -1$ otherwise. For recognition, a sample feature x is classified into one label which has the largest value of the decision function:

$$i^* = arg \max_{i=1,\ldots,M} f_i(x) = arg \max_{i=1,\ldots,M} (w_i^T x + b_i). \tag{2}$$

To cooperate with humans, object information is insufficient for the robot. The robot has to understand what action is given to each object. For this action recognitions, a skin detector that processes the color of pixels is used. Let selected regions of a human skin as action hypotheses. The action recognition is done similar to object recognition. Yet, there is a difference between object and action recognition. Object recognition is possible using only a static image. But, action recognition such as 'move' needs a feature of temporal pattern. As SIFT descriptor is chosen only for a static image, additional features are needed for action recognition. To balance between recognition speed and accuracy, a speed of each action hypothesis is added to the SIFT descriptor. The speed is simply calculated through the displacement between center position of each action hypothesis in two consecutive frames.

Fig. 3. A SIFT descriptor: spatial histogram of the image gradient.

III. ART Model based Learning

When an instructor teaches some procedures for a specific task, the child memorizes a sequence of executions. After memorizing the procedure, the child can follow a memorized sequence without the aid of the instructor. Likewise, a robot can memorize a sequence of human executions in demonstration. Of course, programming all sequences by hand enables the robot to perform a given task. However, it is obviously inefficient to program all sequences for all possible tasks as there exist numerous actions humans can perform. If a robot can observe and memorize a sequence of demonstrations autonomously, the robot can prepare a proper service by observing human demonstrations.

In this paper, the structure of the adaptive resonance system is shown in Fig. 4(b). It consists of an input field (F_1), procedure field (F_2), and task field (F_3). In learning process of the EM-ART model, executions captured in continuous demonstrations are extracted. Then, the sequence of executions is encoded as a procedure and stored in the robot's memory.

A. Execution encoding using Fusion ART

1) Complement coding: Let $I^k = (I_1^k, I_2^k, \ldots, I_n^k)$ denote an input vector, where $I_i^k \in [0, 1]$ indicates the input i to channel k. An activity vector x_i^k is derived by complement

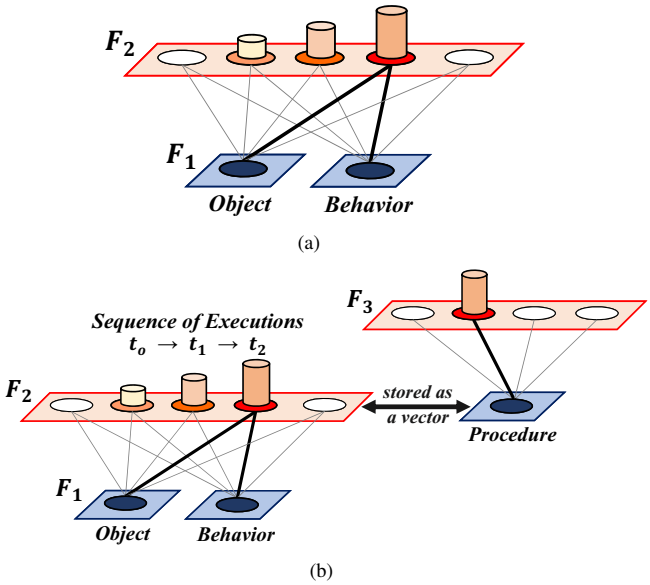


Fig. 4. A designed EM-ART model for procedural memory consists of three fields: the input field F_1, procedure field F_2, and task field F_3.

coding. The activity vector x_i^k is the input vector I_i^k augmented with its complements $\bar{I}_i^k$ such that $I_i^k = 1 - I_i^k$. The main purpose for complement coding is to achieve the pattern generalization , which is to represent a pattern with a range of value instead of an exact value. It can also prevent the weights of the codes from eroding to zeros during learning process.

2) Code activation: The activation value T_j in F_2 is calculated from

$$T_j = \sum_{k=1}^{n} \gamma^k \frac{\left|\mathbf{x}^k \wedge \mathbf{w}_j^k\right|}{\alpha + \left|\mathbf{w}_j^k\right|} \tag{3}$$

where $\mathbf{w}_j^k$ is the weight vector associated with the execution j and input channel k, α^k is a choice parameter for each channel k, the fuzzy AND operation $\wedge$ is defined as $(\mathbf{p} \wedge \mathbf{q})_i \equiv min(p_i, q_i)$, and the norm $|.|$ is defined as $|\mathbf{p}| \equiv \sum_i p_i$ for vectors $\mathbf{p}$ and $\mathbf{q}$.

3) Code competition: Code competition is to a method to select the largest activation value T_j in F_2. The winner is indexed at J as follows:

$$T_J = max\left\{T_j : \text{for all } F_2 \text{ node} j\right\}. \tag{4}$$

4) Template matching: For each channel k, a similarity between the activity vector x^k and the weight w_J^k associated with the selected execution node is calculated using the following function:

$$m_J^k = \frac{\left|\mathbf{x}^k \wedge \mathbf{w}_J^k\right|}{|\mathbf{x}^k|} \geq \rho^k \tag{5}$$

where ρ^k is the vigilance parameter that controls the network resonance. If there are no matched nodes in F_2, an uncommitted node is added to F_2 as a new category node.

5) Template learning: Once the resonance occurs, weights w_J^k are updated as

$$\mathbf{w}_J^{k(new)} = (1 - \beta^k)\mathbf{w}_J^{k(old)} + \beta^k(\mathbf{x}^k \wedge \mathbf{w}_J^{k(old)}) \tag{6}$$

where β^k is the learning rate.

B. Procedure Learning and Retrieval

EM-ART model is applied to memorization of a procedure. EM-ART consists of two stacked multi-channel self-organizing fusion ART networks [15]. Fig. 4 illustrates a designed EM-ART model for procedural memory learning from demonstration. EM-ART can encode both executions and a sequence of executions. Encoding a procedure as a sequence of executions is similar to encoding an execution with an input vector. To make a temporal sequence of executions an input vector in F_2, a time point of each execution should be saved. To retrieve a stream of executions, it is sufficient to save a relative time point among executions. When the j-th node in F_2 is activated first, set $y_j = 1$ and add this index j to activation set Y. Then, before setting newly activated node $y_j = 1$, decay previous activated nodes' values in Y according to

$$y_j^{new} = y_j^{old}(1 - \tau) \tag{7}$$

where τ is a decaying factor. After all executions performed, set zeros to non-activated nodes in Y. Then, the Y is an input vector for encoding the procedure. The Y is classified as a performed task in F_3 by (3) and (4).

In the retrieval phase, some executions may be incorrectly encoded as the recognition result sometimes includes errors. Assuming that the incorrect recognition happens relatively rare, the voting algorithm is applied to prevent inputs with noises from being encoded as an wrong execution. In the voting algorithm, a few sequence of executions are collected and it is followed by selecting the most frequently occurred execution as an input for the retrieval.

IV. Experiments

A. Experimental Setup

Our perception system was implemented on the humanoid robot, Mybot-KSR, developed in RIT Lab. at KAIST. In Mybot-KSR's robotic head, a Primesense Carmine 1.09 was mounted, which is an RGB-D camera used to capture real indoor scene and objects. Camera's depth ranges from 0.35 m to 1.4 m. By assuming that most objects are on the stable plane, such as table and desk, objects and human hand region are taken easily. To demonstrate the procedural memory learning, we defined four scenarios 1) Water the flower; 2) Pour the contents of a bottle; 3) Sort the toys; and 4) Toast a slice of bread. The object lists used in experiments are shown in Fig. 6. Behaviors used in the experiments were "Grasp", "Move", "Tilt", "Put down"and "Push down". A sequences of executions in the demonstration of each scenario were described as follows:

1) Water the flower.
   - Grasp a watering pot on the table.
   - Move the watering pot to the flowers.
   - Tilt the watering pot toward the flowers.

Fig. 5. Experimental setup: an RGB-D camera is mounted on the humanoid robot and all objects are on the tables.

Fig. 6. Objects used in procedural memory learning.

- Put down the watering pot on the table.

2) Pour the contents of a bottle.
   - Grasp a bottle on the table.
   - Move the bottle to the cup.
   - Tilt the bottle toward the cup.
   - Put down the bottle on the table.

3) Sort the toys.
   - Grasp a toy on the table.
   - Move the toy to the box.
   - Put down the toy in the box.

4) Toast a slice of bread.
   - Grasp a bread on the dish.
   - Move a bread to the knob of the toaster.
   - Put down the bread in the toaster.
   - Push down the toaster.

The demonstrator performed each scenario three times. In each demonstration, robot could actively extract executions that were captured in continuous RGB-D streams. The procedural memory was built autonomously by using these successive executions as inputs to the EM-ART for the procedure learning.

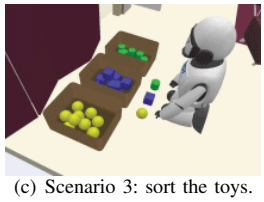
B. Experimental Results

Fig. 8 shows segmented executions in a continuous demonstration using an RGB-D camera. Using the depth information, pixels that correspond to the plane such as a table were removed. Through this, object hypotheses can be found easily. Each feature of object hypotheses was extracted by SIFT, followed by SVM for training and test.

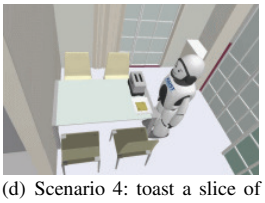
Fig. 9 shows the result that correspond to Scenario 1, water the flower. In this test, partial sequences from the beginning of first encoded executions were used as cues for recognition of the corresponding scenario. The values in Fig. 9 mean activation values in each procedure node. Longer the length of partial sequences, the activation value of the node that corresponds to the scenario was increased. As all successive executions were fed into the memory structure, procedure recognition accuracy was high. Figs. 10 to 12 show results for the rest. These graphs show tendencies similar to Fig. 9. The results confirm the procedural memory modeled by EM-ART can deal with the cues of partial sequences. Also, as more successive executions were fed into the memory structure, the accuracy of the procedure recognition could be increased.

(a) Scenario 1: water the flower.

(b) Scenario 2: pour the contents of a bottle.

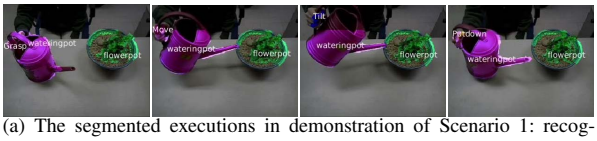


(c) Scenario 3: sort the toys.

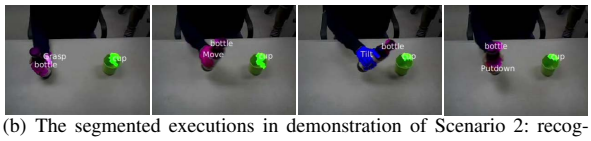


(d) Scenario 4: toast a slice of bread.

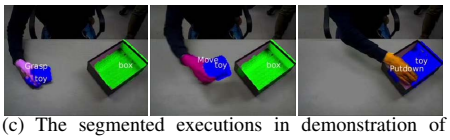
Fig. 7. Four scenarios for demonstration for procedure learning and retrieval.



(a) The segmented executions in demonstration of Scenario 1: recognized objects were a wateringpot and flowerpot.



(b) The segmented executions in demonstration of Scenario 2: recognized objects were a bottle and a cup.



(c) The segmented executions in demonstration of Scenario 3: recognized objects were a toy and a box

(d) The segmented executions in demonstration of Scenario 4: recognized objects were a dish, a bread, and a toaster.

Fig. 8. The recognized results: a continuous demonstration was segmented into each execution, which consists of object and action.

V. Discussions

This paper proposed a scheme on how a robot learn a sequence of executions for carrying out a task from visual demonstration of an instructor using an RGB-D camera. A procedure was composed of executions and each execution contains an object and an action. The EM-ART model capable of spatio-temporal data was employed for the procedural memory structure. The EM-ART model could encode the procedure learned from demonstration and also retrieve it with a partial information of executions. Four scenarios provided defined for demonstration of procedural memory learning and retrieval. By visual demonstration from the instructor in front of the RGB-D camera, each execution was segmented from the continuous RGB-D streams. The sequence of executions were encoded as a procedure into the EM-ART memory structure. With only partial cues from the beginning of executions, it could retrieve a correct procedure.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government

(MSIP) (No. NRF-2014R1A2A1A10051551) and the Technology Innovation Program, 10045252, funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

REFERENCES

[1] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends in Cognitive Sci.*, vol. 6, no. 1l, pp 481-487, 2002.

[2] B. D. Argall, *et al.,* "A survey of robot learning from demonstration," *Robotics Autonomous Syst.*, vol. 57, no. 5, pp. 469-483, 2009.

[3] S. Calinon, *et al.,* "Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression," *IEEE Robot. Automat. Mag.*, vol. 17, no. 2, pp. 44-54, Jun. 2010.

[4] B. Akgun, M. *et al.,* "Keyframe-based learning from demonstration," *Int. J. Social Robotics*, 2012.

[5] D. Grollman and O. C. Jenkins, "Incremental learning of subtasks from unsegmented demonstration," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst.*, Taipei, Taiwan, 2010.

[6] D. H. Grollman and A. Billard. "Donut as I do: Learning from Failed Demonstrations," in *Proc. IEEE Int. Conf. Robotics and Automation*, shanghai, China, 2011.

[7] A. N. Meltzoff, *et al.,* "Learning about causes from people: Observational learning in 24-month-old infants," *Developmental Psychology*, vol. 48, no. 5, pp. 1215-1228, 2012.

[8] E. E. Aksoy, *et al.,* "Categorizing object-action relations from semantic scene graphs," in *Proc. Int. Conf. Robotics and Automation*, Anchorage, Alaska, 2010.

[9] F. Worgotter, *et al.,* "Cognitive agents-a procedural perspective relying on predictability of object-action complexes (oacs)," *Robotics and Autonomous Syst.*, vol. 57, no. 4, pp. 420-432, 2009.

[10] N. Kruger, *et al.,* "Object-action complexes: Grounded abstractions of sensorimotor processes," *Robotics and Autonomous Syst.*, vol. 59, no. 10, pp. 740-757, 2009.

[11] S. T. Mueller and R. M. Shiffrin, "REM-II: A model of the developmental co-evolution of episodic memory and semantic knowledge," in *Proc. Int. Conf. Learning and Develop.*, Bloomington, U.S.A., 2006.

[12] A. Nuxoll and J. E. Laird, "Extending cognitive architecture with episodic memory," in *Proc. Nat. Conf. Artificial Intell.*, Vancouver, Canada, 2007.

[13] A. B. Samsonovich and G. A. Ascoli, "A simple neural network model of the hippocampus suggesting its pathfinding role in episodic memory retrieval," *Learning and Memory*, vol. 12, no. 2, pp. 193-208, 2005.

[14] G. Carpenter, *et al.,* "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *IEEE Trans. Neural Netw.*, vol. 4, pp. 759-771, 1991.

[15] W. Wang, *et al.,* "Neural modeling of episodic memory: Encoding, retrieval, and forgetting, " *IEEE Trans. Neural Netw. and Learning Syst.*, vol. 23, 2012.

[16] W. Wang, *et al.,* "A self-organizing approach to episodic memory modeling," in *Proc. Int. Joint Conf. Neural Netw.*, Barcelona, Spain, 2010.

[17] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381-395, Jun, 1981.

[18] S. Paris and F. Durand. "A fast approximation of the bilateral filter using a signal processing approach," in *Proc. European Conf. Comput. Vision*, Graz, Austria, 2006.

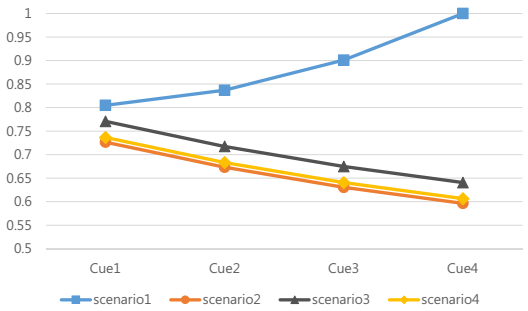


Fig. 9. Activation values in demonstration of Scenario 1: water the flower. A cue in x-axis means how many partial cues are encoded from the beginning of procedures.

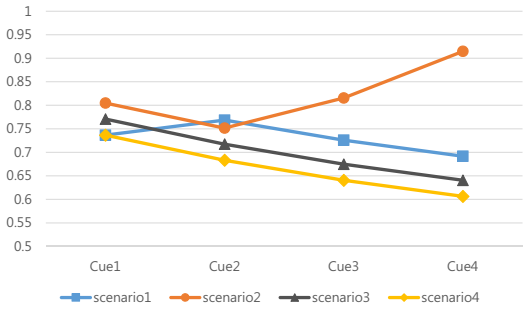


Fig. 10. Activation values in demonstration of Scenario 2: pour the contents of a bottle.

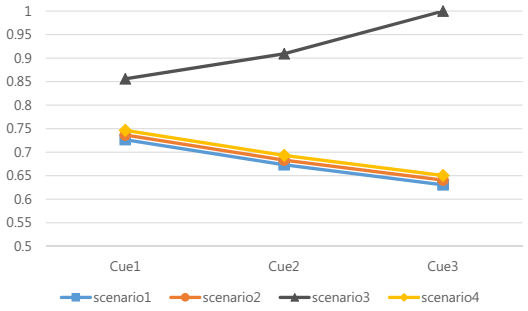


Fig. 11. Activation values in demonstration of Scenario 3: sort the toys.

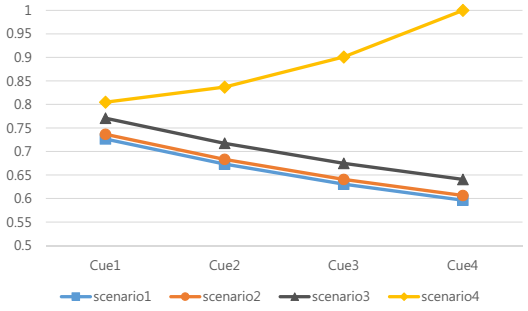


Fig. 12. Activation values in demonstration of Scenario 4: toast a slice of bread.