福井大学リポジトリ
UNIVERSITY OF FUKUI
REPOSITORY

Semantic Spatiotemporal Memory Toward 3D Robotic Vision

メタデータ	言語: eng 出版者: 公開日: 2014-07-30 キーワード (Ja): キーワード (En): 作成者: Murase, Kazuyuki, Hafiz, Abdul Rahman メールアドレス: 所属:
URL	http://hdl.handle.net/10098/8433

(c) 2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

Semantic Spatiotemporal Memory Toward 3D Robotic Vision

Abdul Rahman Hafiz
Department of Human and Artificial Intelligence System, Graduate School of Engineering, University of Fukui, JAPAN
Email: abdu@u-fukui.ac.jp

Kazuyuki Murase
Department of Human and Artificial Intelligence System, Graduate School of Engineering, University of Fukui, JAPAN
Email: murase@u-fukui.ac.jp

***Abstract*—3D robotic vision is proposed using a neural network model that forms sparse distributed memory traces of spatiotemporal episodes of an object. These episodes are generated by the robot interaction with the environment or by robot's movement around 3D object and its perspective to the objects. The traces are distributed in each cell and synapse that participates in many traces. This sharing of representational substrate enables the model for similarity based generalization and thus semantic memory. The results are provided showing that spatiotemporal patterns map to similar traces, as a first step for robot 3D vision system. The model achieves this property by measuring the degree of similarity between the current input pattern on each frame and the expected input given the preceding frame and then adding an amount of noise, inversely proportional to the degree of similarity, to the process of choosing the internal representation for the current frame and the predictable input given the preceding frame.**

***Keywords*-3D robotic vision; Semantic Spatiotemporal Memory; Embodiment;**

I. Introduction

We used a sparse distributed neural network model, TESMECOR [1], [2] (Temporal Episodic and Semantic Memory using Combinatorial Representations), that can learn full episodes from a single trial, and we showed the advantage of using this model in the 3D robot vision system. The model predicts its episode, on each frame, and computes the similarity between the predicted and real input patterns and then adding an amount of noise inversely proportional to the similarity into the process of choosing an internal representation (IR) for that frame. When expected and actual inputs match entirely, no noise is added, allowing those IR cells having maximal input via previously modified weights to be reactivated for fully deterministic recall. When they entirely mismatch, enough noise is added to over write the previous learned weights, resulting in activation of an IR having little overlap with preexisting traces.

The contradicting purposes of episodic memory and pattern recognition, has led other researchers to propose that the brain uses two complementary systems. [3], [4], [5] propose that the function of the hippocampus is to learn new individual information, whereas the purpose of neocortex is to integrate information across individual instances.

We show that TESMECOR performs better when getting feedback from the robotic system, by slowing down or speeding up according to robot movement speed around the object, more than that, by providing the direction of robot-movement to the model. The model can learn various episodes for a single object regardless of the robot movement.

II. Robot & Episodic Spatiotemporal Memory

In the design of our robot we took into consideration the robot's ability to change its perspective to the object and be able to move around it. To achieve that, we designed our robot with two parts the head, and the body Fig.1

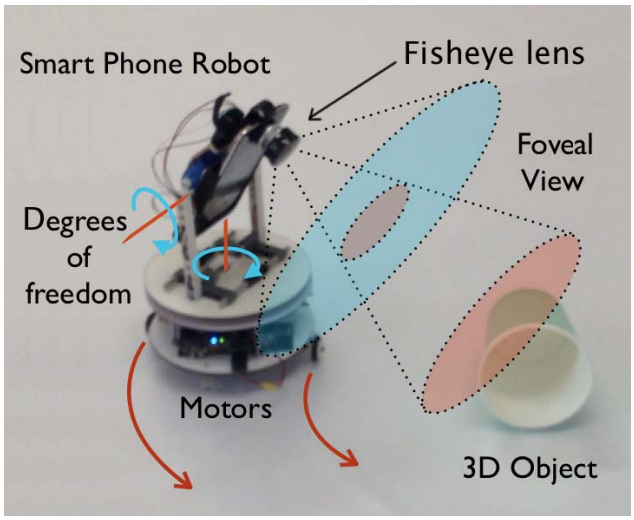
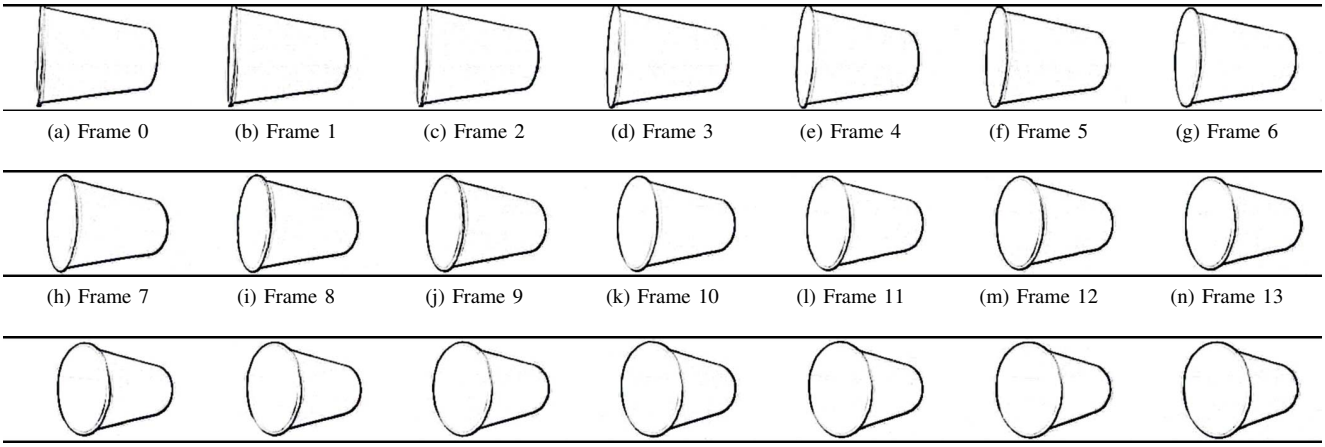


Figure 1: The robot we used in our experiment, using smartphones technology and our previously proposed Foveal Vision System [6]

By dividing the robot vision to foveal and periphery vision, we could implement an attention system similar to our previous work [7]. The head can rotate vertically and horizontally to locate the object in the center of its vision (fovea), while the body helps the robot to move to change the robot position relative to the object.

978-1-4799-3184-2/13 $31.00 © 2013 IEEE
DOI 10.1109/RVSP.2013.61



(a) Frame 0 (b) Frame 1 (c) Frame 2 (d) Frame 3 (e) Frame 4 (f) Frame 5 (g) Frame 6

(h) Frame 7 (i) Frame 8 (j) Frame 9 (k) Frame 10 (l) Frame 11 (m) Frame 12 (n) Frame 13

(o) Frame 14 (p) Frame 15 (q) Frame 16 (r) Frame 17 (s) Frame 18 (t) Frame 19 (u) Frame 20

Figure 2: Different time slices (frames) of an episode for a 3D object (cup) while the robot rotating around it (clockwise), after applying Gaussian Smoothing and Edge Detection

We used smart-phones platform as a robot brain, and used its camera as eye for our robot, and by utilizing the capability of parallel processing of the platform we could achieve real-time image filters onboard. The processed information then send wirelessly to a server for training and recalling stages.

Using this design the robot can rotate around the object and generate episodes of an 3D object as shown in Fig.2.

As shown in Fig.3, TESMECOR model consists of two layers. Layer 1 (L1) consists of binary feature detectors and its layer 2 (L2) consists of competitive modules (CMs). L2 cell has horizontal connections to all other L2 cells via a horizontal matrix (H-matrix) of binary weights, except those in its own CM.

The model operates in the following way. On each frame, a pattern is presented to L1. On that same frame, one L2 cell is chosen at random to become active in each CM corresponding to an active L1 cell. In addition, the horizontal weights from the L2 cells active on the prior frame to those that become active on the current time are increased to their maximal value of one. In this way, spatiotemporal memory traces are embedded in the H-matrix.

On each Frame, the global degree of match between the actual current input and the predicted input, given the spatiotemporal context of the current input, modulates the amount of noise injected into the process of selecting which L2 cells will become active. The smaller the match, the more noise is added and the greater the difference between the internal representation (IR) that would have become active purely on the basis of the deterministic inputs reflecting prior learning and the IR that actually does become active. The greater the match, the less is noise added and the smaller the difference between the most highly implicated IR (on the basis of prior learning) and the actually chosen IR.

The following is the TESMECOR's processing algorithm,

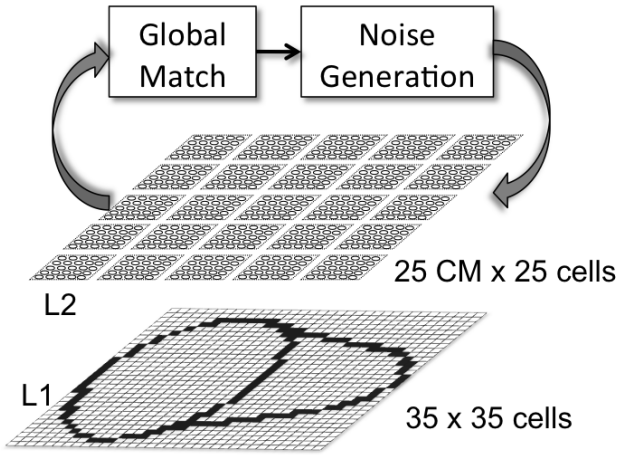


Figure 3: TESMECOR architecture. Each L1 cell has a connection with each cell in L2. Each L2 cell has horizontal connections to all other L2 cells except those in its own CM

which is computed on each time slice for each L2 cell.

In eq.1, each L2 cell, i, computes its total weighted input, $\psi_{i,t}$, from the set, Γ_t, of currently active L1 cells.

$$\psi_{i,t} = \sum_{j \in \Gamma_t} W_{ji} \tag{1}$$

In eq.2, the ψ values are normalized within each CM. That is, we find the maximum ψ value, in each CM and divide all the individual values by the greater of that value and F-matrix threshold, ${}^{F}\Theta_t$. ${}^{F}\Theta_t$ is needed to ensure that small feedforward signals are not amplified in subsequent normalization steps.

$$\Psi_{i,t} = \frac{\psi_{i,t}}{\max(\max_{j \in CM}(\psi_{i,t}), {}^{F}\Theta_t)} \tag{2}$$

In eq.3, each L2 cell, *i*, computes its total weighted input, $\phi_{i,t}$, from the set, Δ_{t-1}, of L2 cells active on the prior time slice.

$$\phi_{i,t} = \sum_{j \in \Delta_{t-1}} W_{ji} \quad , t > 0 \tag{3}$$

In eq.4, the ϕ values are normalized within each CM. That is, we find the maximum ϕ value, in each CM and divide all the individual values by the greater of that value and an H-matrix threshold,${}^{H}\Theta_t$.${}^{H}\Theta_t$ is needed to ensure that small H values are not amplified in subsequent normalization steps.${}^{H}\Theta_t$ also varies from one time slice to the next.

$$\Phi_{i,t} = \frac{\phi_{i,t}}{\max(\max_{j \in CM}(\phi_{i,t}), {}^{H}\Theta_t)} \quad , t > 0 \tag{4}$$

In eq.5 works differently on the first time slices of episodes than on the rest. When t >0, we multiply the two pieces of evidence, $\Psi_{i,t}$ and $\Phi_{i,t}$, that cell *i* should become active but we do this after passing them through separate exponential filters. Since $\Psi_{i,t}$ and $\Phi_{i,t}$, are both between 0 and 1, the final $\chi_{i,t}$ values output from this step are also between 0 and 1. The exponential filters effect a generalization gradient: the higher the exponents, *u*,*w*, and *v*, the sharper the gradient and the more sensitive the model is to differences between inputs (i.e., the finer the spatiotemporal categories it would form) and the less overlap between the internal representations chosen by the model.

$$\chi_{i,t} = \begin{cases} \Psi_{i,t}^{u} \Phi_{i,t}^{v} & , t > 0 \\ \Psi_{i,t}^{w} & , t = 0 \end{cases} \tag{5}$$

In eq.6, we normalize the combined evidence vector, again subject to a threshold parameter, ${}^{\chi}\Theta_t$, that prevents small values from erroneously being amplified.

$$X_{i,t} = \frac{\chi_{i,t}}{\max(\max_{j \in CM}(\chi_{i,t}), {}^{\chi}\Theta)} \quad , t > 0 \tag{6}$$

In eq.7, we determine the maximum value, $\pi_{k,t}$, of the $X_{i,t}$ values in each CM. These π values constitute local, i.e., within each CM, comparisons between the model's expected and actual inputs.

$$\pi_{k,t} = \max_{j \in CM_k} X_{i,t} \quad , 1 \leq k \leq Q \tag{7}$$

In eq.8, we compute the average of these local comparison results across the *Q* CMs of L2, resulting in the model's global comparison, G_t, of its expected and actual inputs.

$$G_t = \sum_{k=1}^{Q} \pi_{k,t} / Q \tag{8}$$

In eq.9, we convert the $X_{i,t}$ values back into a probability distribution whose shape depends on G_t. We want to achieve the following: if G_t is 1.0, indicating that the actual input has perfectly matched the model's expected input, then, in each CM, we want to choose, with probability 1.0, the cell belonging to the IR representing that expected input. On the other hand, if G_t = 0, then we want to make all the cells, in any given CM, be equally likely to be chosen winner. the function, *f*, is a sigmoid that meets the above goals.

$$P_{i,t} = \frac{f(X_{i,t}, G_t)}{\sum_{j \in CM} f(X_{i,t}, G_t)} \tag{9}$$

To summarize, on each frame, every L2 cell compares two evidence vectors, the H-vector, reflecting the sequence of patterns leading up to the present frame (temporal context), and the F-vector, reflecting the current spatial pattern (spatial context). These vectors are separately nonlinearly filtered and then multiplicatively combined. The combined evidence vector is then renormalized and nonlinearly filtered before being turned into a probability distribution that governs the final selection of L2 cells to become active.

III. Results

In this section, we provide the results of preliminary investigations of the model demonstrating that it performs similarity-based generalization and categorization in the spatiotemporal pattern domain.

The three cases as shown in Fig.4 are performed, each with two different speed, fast and slow, generating 6 episodes of 20 frames. The model was then tested by presenting sequence of 5 frames of the perturbed episodes as prompt. Following the prompt frames, the model entered a free-running mode (i.e. cutting off any further input) and processing continued from that point merely on the basis of signals propagating in the H-projection.

In Table 1, R^1, R^2 is the recall accuracy with frame 0 to 5 and frame 10 to 15 given as prompt for the model respectively.

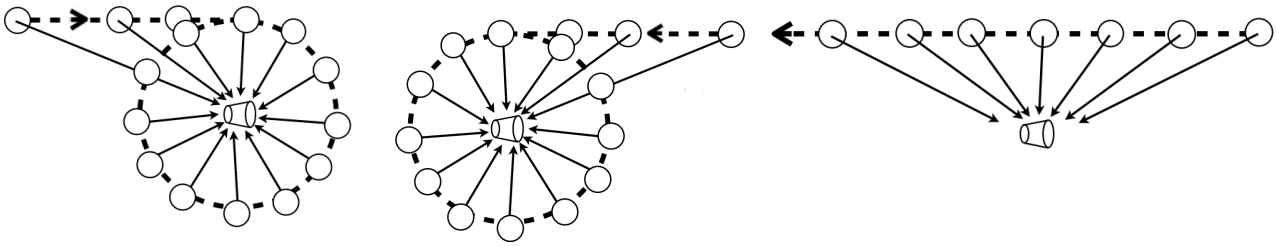
Table I: Categorization Results

Cases	R^1_{Case}	R^2_{Case}
Case1 fast	90.1	89.0%
Case1 slow	92.3	93.7%
Case2 fast	93.3	92.9%
Case2 slow	89.7	91.4%
Case3 fast	82.3	87.3%
Case3 slow	85.7	88.2%

To calculate the recall accuracyR_e, for a given episode *e*, we used eq.10.

$$R_e = (C_e - D_e)/(C_e + I_e) \tag{10}$$

where C_e is the number of L2 cells correctly active during recall of eth episode, D_e is the number of deleted L2 cells, and I_e is the number of intruding L2 cells.



(a) Case 1 (b) Case 2 (c) Case 3

Figure 4: Robot movement around the object

IV. Discussion

These results indicate that the model was extremely good at locking into the trace corresponding to the most-closely-matching original episode. The accuracy measure, R_{set} (eq.10) measures how close the recall L2 trace is to the L2 trace of the most-closely-matching original episode. The view taken here in is that given that the pattern to be recalled are spatiotemporal, the most relevant measure of performance is the measure of accuracy on the last frame of the test episode. If the model can "lock into" the correct memory trace by the end of the recalled trace, then that should be sufficient evidence that model has recognized the input as an instance of a familiar episode.

More than that the result R^2 from the table show that the model can correctly locking into the episode starting from any time frame, this indicate that the robot can recognize its position relative to the 3D objects.

We believe that this approach is a first step to study how the brain can think and dream in a 3D world, by using this approach our robot could arguably rotate 3D objects in its mind.

V. Conclusion

These results provide preliminary evidence that using TESMECOR as a robot vision system allows the system to exhibits generalization, and categorization, in the spatiotemporal domain, in addition to that, it allows the robot to recognize its location and movement around 3D objects. In the future we want to farther investigate the hierarchical Spatiotemporal Memory model, which will allow our robot to integrate its various sensory input to the same neural network model.

Acknowledgment

This study was supported by grants to K.M. from Japanese Society for promotion of Sciences and Technology, and the University of Fukui.

References

[1] G. J. Rinkus, *A Combinatorial Neural Network Exhibiting both Episodic Memory and Generalization for Spatio-Temporal Patterns.* Ph.D. Thesis, Graduate School of Arts and Sciences, Boston University. 1996.

[2] G. J. Rinkus, *A cortical sparse distributed coding model linking mini- and macrocolumn-scale functionality.* Frontiers in Neuroanatomy, 4(17). 2010.

[3] G. Carpenter, and S. Grossberg, *Massively parallel architectures for a self-organizing neural pattern recognition machine.* Computer Vision, Graphics and Image Processing. 37, 54-115. 1987.

[4] R.L. Coultrip, and R.H. Granger, *Sparse random networks with LTP learning rules approximate Bayes classifiers via Parzen's method.* Neural Networks, 7(3), 463-476. 1994.

[5] W. Duch, R.J. Oentaryo, and M. Pasquier, *Cognitive architectures: where do we go from here?* Frontiers in Artificial Intelligence and Applications, Vol. 171 (Ed. by Pei Wang, Ben Goertzel, and Stan Franklin), IOS Press, pp. 122-136. 2008.

[6] A.R. Hafiz, F. Alnajjar, K. Murase *A Novel Dynamic Edge Detection Inspired from Mammalian Retina toward Better Robot Vision.* 12th International Symposium on Robotics and Applications (ISORA2010), World Automation Congress (WAC2010), Kobe, Japan, Sept 19-23, 2010, ISOLA 301. 2010.

[7] A.R. Hafiz, K. Murase *iRov: A Robot Platform for Active Vision Research and as Education Tool.* 6th International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2011), Bielefeld, Germany, May 2325, 2011, Advances in Autonomous Mini Robots (Ulrich Ruckert, Joaquin Sitte, Felix Werner, Eds), Springer, Heidelberg, 20012, pp.173-182. 2011.