A Self-Organizing Approach to Episodic Memory Modeling

Wenwen Wang, Budhitama Subagdja, Ah-Hwee Tan, *Senior Member, IEEE*
and Janusz A. Starzyk, *Senior Member, IEEE*

***Abstract*— This paper presents a neural model that learns episodic traces in response to a continual stream of sensory input and feedback received from the environment. The proposed model, based on fusion Adaptive Resonance Theory (fusion ART) network, extracts key events and encodes spatio-temporal relations between events by creating cognitive nodes dynamically. The model further incorporates a novel memory search procedure, which performs parallel search of stored episodic traces continuously. Comparing with prior systems, the proposed episodic memory model presents a robust approach to encoding key events and episodes and recalling them using partial and erroneous cues. We present experimental studies, wherein the model is used to learn episodic memory of an agent's experience in a first person game environment called Unreal Tournament. Our experimental results show that the model produces highly robust performance in encoding and recalling events and episodes even with incomplete and noisy cues.**

I. Introduction

Episodic memory is a special class of memory system that allows one to remember his/her own experiences in an explicit and conscious manner [13]. Although episodic memory is considered to be less important than semantic memory, recent research has found episodic memory to be crucial in supporting many cognitive capabilities, including concept formation, representation of events in spatio-temporal dimension and record of progress in goal processing [3].

The specific functionalities mentioned above suggest that episodic memory should not be just a storage of one's past experiences, but should support the representation of complex conceptual and spatio-temporal relations among one's experienced events and situations. While many existing episodic memory models provide the encoding of events and relations between events, most still have limitations in capturing conceptual relations between events (e.g. [6], [7]). On the other hand, those models supporting the intricate relations of concepts and events are not able to process complex sequences of events as a whole (e.g. [9], [4], [10]).

In this paper, we present our study on the representation and learning of episodic memory. Based on a generalization of fusion Adaptive Resonance Theory (ART) [12], we show how the neural network can be used in an episodic memory model, for encoding an individual's experience in the form of events as well as the spatio-temporal relations among events. The model supports complex-event storage through its multiple-channel pattern learning capability inherited from fusion ART. An additional encoding scheme is also introduced that allows complex sequences of events to be grouped and recognized. The model further incorporates a novel approximate memory search procedure, which performs parallel search of stored episodic traces continuously in response to potentially imperfect search cues.

We have conducted experimental studies, wherein the proposed model is used to learn episodic memory based on an agent's encounters in a first person shooting game called Unreal Tournament. Our experiment results show that the model is able to provide a robust level of performance in encoding and recalling events and episodes using various types of input queries involving incomplete and noisy cues.

The rest of this paper is organized as follows. Section II discusses the issues and challenges in modeling episodic memory. Section III presents the architecture of our proposed episodic memory model. Section IV and section V present the algorithms and processes for event and episode encoding and retrieval respectively. Section VI reports the experimental results based on the Unreal Tournament game. Section VII provides a brief discussion of selected work on episodic memory models. The final section concludes and highlights future work.

II. Issues and Challenges

A. Memory Formation

As discussed in Section I, two basic elements of episodic memory are events and episodes. An event can be described as a snapshot of experience. Usually, by aggregating attributes of interest, a remembered event can be used to answer critical questions about the corresponding experience, such as what, where and when. On the other hand, an episode can be considered as a temporal sequence of events that one experiences.

To enable efficient encoding of events and episodes, an episodic memory model should be able to distinguish between distinct events and episodes with a well-defined matching scheme. The basic challenge regarding building the memory storage matching scheme is: On one hand, the novelty detection should be sufficiently strict to distinguish highly similar but semantically different events (e.g. "Mary borrowed a book from Emma yesterday" is different from "Mary borrowed a book from Bob yesterday"); On the other hand, it should also be loose enough to tolerate minor differences for events within a single episode, such as slight changes within observed events and their temporal order. Hence, the critical characteristics for the matching scheme is

Wenwen Wang, Budhitama Subagdja and Ah-Hwee Tan are with the School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798; E-mail: {wa0003en, budhitama, asahtan}@ntu.edu.sg.

Janusz A. Starzyk is with the School of Electrical Engineering and Computer Science, Russ College of Engineering and Technology, Ohio University, Stocker Center r. 347, U.S.A.; E-mail: starzyk@bobcat.ent.ohiou.edu.

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

its high efficiency in determining the significant differences while tolerating all minor variances for both events and episodes encoding. Therefore, an efficient matching scheme should also lead to a parsimonious memory storage as well as faster memory operations.

B. Memory Retrieval

We identify three major tasks in episodic memory retrieval, namely event detection, episode recognition, and episode recall, described as follows.

- **Event detection** refers to the recognition of a previously learned event based on a possibly incomplete description of the current situation. The episodic memory model should be able to search for similar memorized events, which can be used to complete or refine the given description.
- **Episode recognition** refers to the identification of a stored episode in the episodic memory in response to a partial event sequence. Following the effect of episode recognition, episodic memory model may also perform event completion if the present event sequence has missing parts in the event representation. Two basic requirements of episode recognition include: (1) tolerance to incomplete cues, which only form part of the stored episodes and (2) tolerance to errors in situational information, for example, noise in event attributes and variations in the order of event sequences.
- **Episode recall** is the playback of episode(s) in response to an external cue, such as "what did I do yesterday?". When a cue is presented, episodic memory answers the cue with the most closely matched episode according to its similarity. During the episode playback, compared with the stored information, an instant cue may present minor disparities in individual event representations as well as their temporal orderings. The episodic memory model should be able to identify and tolerate this imperfection during recall.

C. Summary

Taking the above into consideration, an episodic memory model should satisfy the following basic requirements:

- Efficient event representation, which is able to describe complex situations and events
- Efficient episode representation, which explores spatio-temporal relations among events which form the episode
- Well-defined generalizations on representations, which accurately distinguishes critical and irrelevant differences among them (for both events and episodes)
- Fast memory operations, including memory encoding and retrieving
- High error tolerance to incomplete or noisy cues

III. The Proposed Model

Our proposed episodic memory (EM) model is built by hierarchically joining two multi-channel self-organizing neural networks, called fusion ART networks. Based on Adaptive Resonance Theory (ART) [2], fusion ART dynamics offers a set of universal computational processes for encoding, recognition, and reproduction of patterns.

As shown in Figure 1, the model consists of three layers of memory fields: F_1, F_2 and F_3. The F_1 layer, connected with the working memory, holds the activation values of all situational attributes. Based on the F_1 pattern of activations, a cognitive node in F_2 is selected and activated as a recognition of the event. Following that, the activation pattern of an incoming event can be learnt by adjusting the weights in the connections between F_1 and F_2.

Besides categorizing events, the F_2 layer also acts as a medium-term memory buffer for event activations. A sequence of events produces a series of activations in F_2. The activations in F_2 decay over time such that a graded pattern of activations is formed representing the order of the sequence. This activity pattern, which represents an episode, is similarly learnt as weighted connections between F_2 and the selected category in F_3.

Once an episode is recognized through a selected node in F_3 is selected, the complete episode can be reproduced by a top down activation process (readout) from F_3 to F_2. The events in the episode can also be reproduced by reading out the activations from F_2 to F_1 following the order of the sequence held in the F_2 layer.

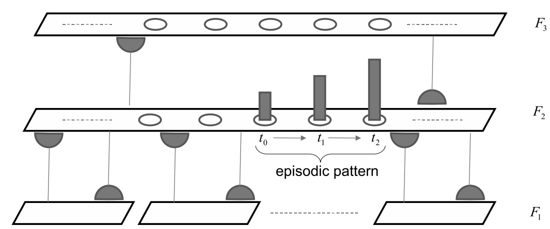

Fig. 1. Episodic model

The computational principles and algorithms used for encoding, storing and retrieving events and episodes are described in details in the following sections.

IV. Event Encoding and Retrieval

An event consists of attributes characterizing what (e.g. subject, relation, action, object), where (e.g. location, country, place), and when (e.g. date, time, day, night) an event occurs. Figure 2 shows an example of the structure of an input event based on the Unreal Tournament domain [14]. This structure is also used in the experiments for evaluating the proposed model (explained in later sections). In the structure shown, the location is expressed using a 3-dimensional cartesian coordinate system; other task and internal states include the observed distance from the enemy (another agent), the availability of collectable items, and the agent's health and ammo level.

There are four behavior choices (actions) available for the agent, including running around, collecting items, escaping from battle and engaging in fire. The consequence of a battle situation (e.g. killing and being damaged) is presented to the model as a reward value. Information about time is not included in this case, but it can be assumed that the temporal information has been represented inherently in the episode.

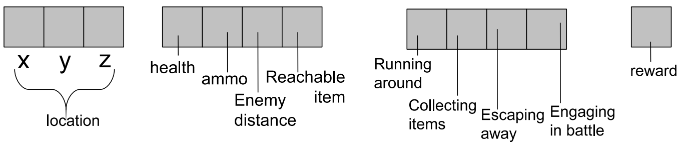

Fig. 2. Event encoding

A. Fusion ART

Fusion ART network is used to learn individual events encoded as weighted connections between the F_1 and F_2 layers. In this case, an event is represented as a multi-channel input vector. Figure 3 illustrates the fusion ART architecture, which may be viewed as an ART network with multiple input fields. Each event's attribute is represented as the activity of a node in the corresponding input field.

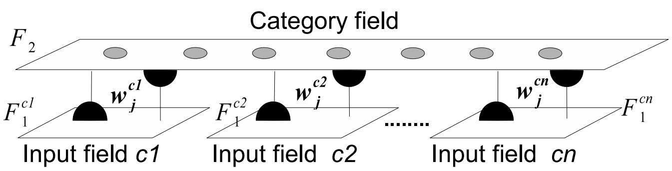

Fig. 3. Fusion ART

The detailed dynamics of a multi-channel fusion ART can be described as follows.

Input vectors: Let $\mathbf{I}^k = (I_1^k, I_2^k, \ldots, I_n^k)$ denote an input vector, where $I_i^k \in [0,1]$ indicates the input i to channel k, for $k = 1, \ldots, n$. With complement coding, the input vector $\mathbf{I}^k$ is augmented with a complement vector $\bar{\mathbf{I}}^k$ such that $\bar{I}_i^k = 1 - I_i^k$.
Input fields: Let F_1^k denote an input field that holds the input pattern for channel k. Let $\mathbf{x}^k = (x_1^k, x_2^k, \ldots, x_n^k)$ be the activity vector of F_1^k receiving the input vector $\mathbf{I}^k$ (including the complement).
Category fields: Let F_i denote a category field and $i > 1$ indicate that it is the ith field. The standard multi-channel ART has only one category field which is F_2. Let $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ be the activity vector of F_2.
Weight vectors: Let $\mathbf{w}_j^k$ denote the weight vector associated with the jth node in F_2 for learning the input pattern in F_1^k.
Parameters: Each field's dynamics is determined by choice parameters $\alpha^k \geq 0$, learning rate parameters $\beta^k \in [0,1]$, contribution parameters $\gamma^k \in [0,1]$ and vigilance parameters $\rho^k \in [0,1]$.

The dynamics of a multi-channel ART can be considered as a system of continuous resonance search processes comprising the basic operations as follows.
Code activation: A node j in F_2 is activated by the choice function

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

where the fuzzy AND operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv min(p_i, q_i)$, and the norm $|.|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$ for vectors $\mathbf{p}$ and $\mathbf{q}$.
Code competition: A code competition process follows to select a F_2 node with the highest choice function value. The winner is indexed at J where

$$T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}. \tag{2}$$

When a category choice is made at node J, $y_J = 1$; and $y_j = 0$ for all $j \neq J$ indicating a *winner-take-all* strategy.
Template matching: A template matching process checks if resonance occurs. Specifically, for each channel k, it checks the *match function* m_J^k of the chosen node J meets its vigilance criterion such that

$$m_J^k = \frac{|\mathbf{x}^k \wedge \mathbf{w}_J^k|}{|\mathbf{x}^k|} \geq \rho^k. \tag{3}$$

If any of the vigilance constraints is violated, mismatch reset occurs or T_J is set to 0 for the duration of the input presentation. Another F_2 node J is selected using choice function and code competition until a resonance is achieved. If no selected node in F_2 meets the vigilance, an uncommitted node is recruited in F_2 as a new category node selected by default.
Template learning: Once a resonance occurs, for each channel k, the weight vector $\mathbf{w}_J^k$ is modified by the following learning rule:

$$\mathbf{w}_J^{k(\text{new})} = (1 - \beta^k)\mathbf{w}_J^{k(\text{old})} + \beta^k(\mathbf{x}^k \wedge \mathbf{w}_J^{k(\text{old})}). \tag{4}$$

Activity readout: The chosen F_2 node J may perform a readout of its weight vectors to an input field F_1^k such that $\mathbf{x}^{k(\text{new})} = \mathbf{w}_J^k$.
A fusion ART network, consisting of different input (output) fields and a category field, is a flexible architecture that can be made for a wide variety of purposes. The neural network can learn and categorize inputs and can be made to map a category to some predefined fields by a readout process to produce the output. Another important feature of the fusion ART network regarding its use in episodic memory is that no separate phase of operation is necessary for conducting recognition (activation) and learning. Learning can be conducted by adjusting the weighted connections while the network search and select the best matching node. When no existing node can be matched, a new uncommitted node is allocated to represent the new pattern. Hence, the network can grow in response to novel patterns.

B. Algorithm for Event Encoding and Retrieval

Based on the above description of fusion ART, an event can be encoded as an input vector to the network as defined in Figure 2. Using the standard operations of fusion ART, the recognition task can be realized by a bottom-up activation given the input vector. On the other hand, the top-down activation (readout operation) achieves the recall task. Figure 4 illustrates the bottom up and top down operations for learning, recognition, and recalling an event.

In particular, the algorithm for learning and recognizing events can be described as follows:

Algorithm 1 (Event Encoding):

1 Given an incoming pattern of event in F_1
2 Activate and select a node (through winner-take-all) in F_2
3 WHILE the node is not in resonant condition
or the node has been selected previously
5 Do reset the current node activation
6 Do choose another node in F_2
7 IF no matching node can be found in F_2 THEN
8 Do recruit an uncommitted node in F_2
9 Do learn it as a novel event

The algorithm for event recognition and encoding above is also designed to handle complex sequences involving repetition of events. The iteration condition in line 3 Algorithm 1 ensures that the same node will not be selected if it has been selected previously as a matching category in the same episode. This leads to the duplication of an event category if the event pattern is repeated in a sequence (episode). One important parameter for event recognition and encoding is ρ^k, the vigilance parameter for each input channel k in F_1. The vigilance values are is used as thresholds for the template matching process, as described in Section IV-A. If the same vigilance value is applied to all input channels in F_1 layer, ρ^e is introduced to represent this unified vigilance value for encoding and retrieval of events.

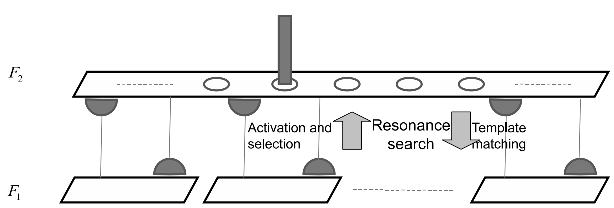

Fig. 4. Operations between F_1 and F_2

V. Episode Learning and Retrieval

A. Episode Representation and Learning Algorithm

A crucial part of EM is to retain the sequential or temporal order between events. However, in the standard model of fusion ART, this feature of sequential representation is still lacking. The EM model proposed in this paper extends the fusion ART model so that it can associate and group patterns across time. The approach of encoding temporal relation in ART-based neural network has actually been suggested in [5], [1].

The method, called *invariance principle*, suggests that activation values can be retained in a working memory (neural field) in such a way that the temporal order in which they occur are encoded by their activity pattern. To retain the temporal order, each entry of activation item multiplicatively modifies the activity of all previous items. Based on the multiplying factor, an analog pattern emerges in the neural field reflecting the order the events are presented. Thus, the temporal order of items in a sequence, encoded as relative ratios between their values, remains invariant.

The method has accurately emulated the characteristic of serial learning conforming the psychological data about human working memory [5]. The approach has also been simplified as *gradient encoding* by replacing the multiplication with the adding/subtracting operation and is successfully applied to iFALCON, a belief-desire-intention (BDI) agent architecture composed of fusion ARTs [11].

To represent a sequence in our EM model, the invariance principle is applied, so that an activation value in F_2 indicates a time point or a position in an ordered sequence. The most recently activated node in F_2 has the maximum activation of 1 while the previously selected ones are multiplied by a certain factor decaying the values over time. Suppose $t_0, t_1, t_2, ..., t_n$ denote the time points in an increasing order, and y_{t_i} is a node value of the activity vector of the category field that is activated or selected at time t_i, the activation values in F_2 form a certain pattern such that $y_{t_i} > y_{t_{i-1}} > y_{t_{i-2}} > ... > y_{t_{i-n}}$ holds where t_i is the current or the latest time point. This pattern of activation also corresponds to the so called *recency effect* in STM (Short-Term Memory) in which a later recently presented item has a higher chance to be recalled from the memory.

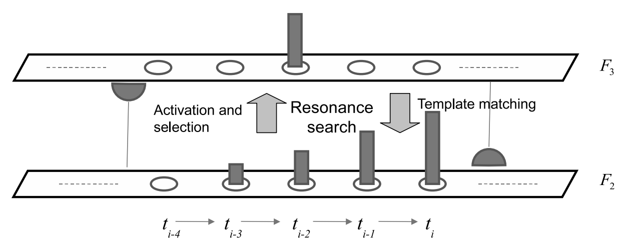

Fig. 5. Operations between F_2 and F_3

The process of episode learning in the proposed model is shown in Figure 5. While a newly activated node has an activation of 1, the activation value of any other node j in F_2 is decayed in each time step so that $y_j^{(new)} = y_j^{(old)}(1 - \tau)$, where y_j is the activation value of the jth node in F_2 and $\tau \in (0, 1)$ is the decaying factor.

Concurrently, the sequential pattern can be stored more permanently as weighted connections in the fusion ART network. As mentioned previously, F_2 and F_3 can be considered respectively as the input field and category field of another fusion ART neural network with a single input field only. Each node in F_3 represents an episode encoded as a pattern of sequential order according to the invariance principle in its weighted connections.

The overall algorithm of episode learning can be described as follows:

Algorithm 2 (Episode activation and learning):

1 FOR EACH event in an episode $\mathcal{S}$
2 Do select a node in F_2 based on the input pattern in F_1
3 Do set the activation y_j of the selected node to maximum
4 Do decay activations of all previously selected nodes i
so that $y_i^{(new)} = y_i^{(old)}(1 - \tau)$
5 At the end of $\mathcal{S}$,
Do activate, select, and learn a node in F_3
based on the pattern formed in F_2 by resonance search

One important parameter used in the episode learning algorithm is ρ^s, the vigilance parameter in the F_2 field. The vigilance parameter is used as a threshold for the template matching process as described in Section IV-A.

B. Episode Retrieval

After episodes are learnt, a particular episode can be recalled based on different types of cues. A cue for the retrieval can be a partial sequence of the episode starting from the beginning or any position in the sequence. Based on the cue, the entire episode can be reproduced through the read out operation. An important characteristic of the proposed EM model is that the retrieval can be done in a robust manner as the activation and matching processes comprise analog patterns. This feature is useful when the cue for retrieval is imperfect or noisy. The approximate retrieval is also made possible by the use of fusion ART as the basic computational principle for all parts of the EM. For example, lowering the vigilance parameter ρ^s of F_2 can make it more tolerant to noises or incomplete cues.

To retrieve an episode based on a weak cue, such as a subsequence of episode, a continuous search process is applied, in which the activity pattern of the cue is formed in F_2 while the F_3 nodes are activated and selected at the same time through the resonance search process. As long as a matching node is not found (still less than ρ^e), the next event is received activating another node in F_2 while all other nodes are decayed. For a cue as a partial episode, the missing event can mean no more new activation in F_2 while other nodes are still decayed. The algorithm for recognizing an episode based on imperfect cues can be described in Algorithm 3.

Algorithm 3 (Episode recognition):

1 FOR EACH incoming event
2 Do select a node in F_2
based on the incoming event in F_1 by resonance search
3 Do set the selected node activation y_j to maximum
4 Do decay the value of every previously selected node i
so that $y_i^{(new)} = y_i^{(old)}(1 - \tau)$
5 Do activate and select a node (winner-take-all)
in F_3 based on the current pattern formed in F_2
6 IF the selected node matches with the pattern THEN
7 Episode is recognized and the search finishes
8 Continue to the next stage of retrieval by exiting the loop

Once an episode is recognized, the complete pattern of sequence can be reproduced readily in the F_2 layer by the read out operation from the selected node in F_3 to the nodes in F_2. However, to reproduce the complete episode as a sequence of events, the corresponding values in F_1 layer must be reproduced one at a time following the sequential order of the event in the episode. The proposed EM model uses a vector complementing the values in F_2 before reading out the complete events in F_1. After the sequential pattern is readout to the field in F_2 which can be expressed as vector $\mathbf{y}$, a complementing vector $\overline{\mathbf{y}}$ can be produced so that for every element i in the vector, $\overline{y}_i = 1 - y_i$. Given the vector $\overline{\mathbf{y}}$, the node corresponding to the largest element in $\overline{\mathbf{y}}$ is selected first to be read out to the F_1 fields. Subsequently, the current selected element in the vector is suppressed by resetting it to zero, and the next largest is selected for reading out until everything is suppressed. In this way, the whole events of the retrieved episode can be reproduced in the right order.

VI. EXPERIMENTS

In this section, we demonstrate the performance of the proposed EM model in a first-person shooter game environment called Unreal Tournament (UT). During the game, a non-player character (NPC) agent receives events describing the situation it experiences. The EM model is used to learn episodic traces from those events, which are subsequently subjected to various recall tasks for performance evaluation.

A. Experiment Setup

An event in the UT can be defined as a vector, as shown in Figure 2. Each episode includes all events experienced by the agent during a battle in the game.

The data set applied in the experiments is taken from 50 battles (i.e. episodes) played by an agent. There are in total 1847 events in the data set. The number of events within an episode varies from 7 to over 250. After the EM model is built using the collected data set, tests are conducted to evaluate the accuracy of memory retrieval, subject to variations of cues, described as follows:

- The cue is a full or partial event sequence of a recorded episode starting from the beginning of the episode. The test are conducted over different cue length and vigilance parameter ρ^s.
- The cue is a full or partial event sequence of recorded episodes starting from the end of the episode. The test are conducted over different cue length and vigilance parameter ρ^s.
- The cue is a fixed-length partial event sequence of recorded episodes starting from various location of the episode. The test are conducted over different cue length and vigilance parameter ρ^s.
- The cue is a noisy or erroneous full length event sequence of recorded episodes. The test are conducted over different error rates and combinations of vigilance parameters (ρ^e and ρ^s).

For the ease of the parameter setting, all our present experiments use a stanndard vigilance value (ρ^e) throughout all the fields in the F_1 layer.

B. Retrieving from Beginning of Episodes

In this retrieval test, we extract partial sequences from the beginning of the recorded episodes as cues for retrieving the episodes. The cues are of different lengths, ranging from whole, 1/2, 1/3, 1/4, to 1/5 of the length of the episodes. Figure 6 shows the retrieval accuracy using cues of various length under different vigilance values of ρ^s.

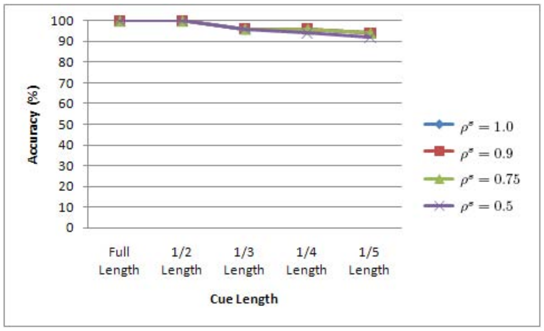

Fig. 6. Accuracies of retrieval with cues from beginning of episodes.

As shown in Figure 6, the model can accurately retrieve most stored episodes based on partial cues with different lengths over a large range of vigilance values. A longer cue typically provides a higher retrieval accuracy. Meanwhile, a lower vigilance gives more tolerance to the differences in cue, thus enhancing the accuracy.

C. Retrieving from End of Episodes

In this retrieval test, cues are extracted from the end of the recorded episodes. Similarly, cues of various length are used, ranging from whole, 1/2, 1/3, 1/4 to 1/5 of the original length of the episodes. Figure 7 shows the retrieval accuracy using cues of different length under different vigilance values of ρ^s.

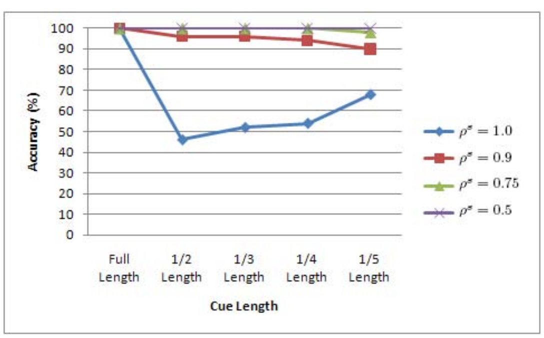

Fig. 7. Accuracies of retrieving with cues from end of episodes.

Referring to Figure 7, with a low vigilance level, the model can accurately retrieve most stored episodes based on the partial cue with different lengths. It is because a lower vigilance gives more tolerance to cue differences. Additionally, a longer cue generally provides a higher retrieval accuracy.

D. Retrieving from Arbitrary Location of Episodes

In this retrieval test, each stored episode is divided into four partial cues. Each such partial cue is forwarded to the model for episode retrieval. Figure 8 shows the retrieval accuracy under various vigilance parameter ρ^s.

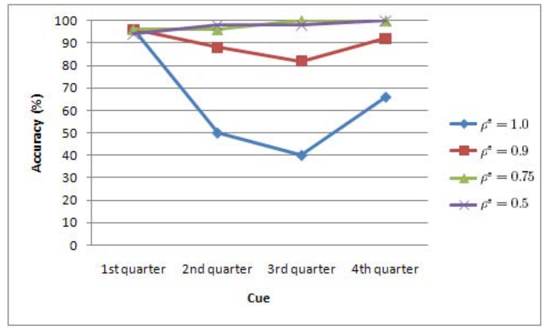

Fig. 8. Accuracies of retrieving from arbitrary location.

As shown in Figure 8, with a low vigilance level, the model can accurately retrieve most stored episodes based on partial cues extracted from various locations. However, at a high vigilance level, the retrieval accuracy drops significantly when the cue is near the middle segments within the episodes.

E. Retrieving with Noisy Events

To test the robustness of the model, we further conduct the retrieval test with noisy data. Two types of errors are applied in the test as follows:

- Error in individual event's attributes;
- Error in event ordering within a complete sequence.

In this section, we test the model's robustness in dealing with the first type of noise. The corresponding noisy data set is directly derived from the original data set using the method described in Algorithm 4, with specified error rate.

Algorithm 4 (Generation of noisy events):

```
Input: Error rate r ∈ (0, 1)
FOR EACH event in the original data set
    FOR EACH attribute A in the event
        IF A is boolean value
            Toggle A value with a probability of r
        IF A is real value
            A.value = A.max − (A.value − A.min)
            with a probability of r
```

We test the model with various error rates on event representation and the results are shown in Figures 9 to 12. We observe that, to achieve a high retrieval accuracy with noisy cues, the model requires a high vigilance ρ^e for event recognition in the F_2 layer, but a low vigilance ρ^s during sequence recognition in the F_3 layer. With $\rho^s = 0.5$ and $\rho^e = 1.0$, the model can achieve 100% retrieval accuracy with an error rate as high as 10%. The results show that during event recognition, the higher vigilance (ρ^e) is required to distinguish the highly similar but conceptually different events; In contrast, episode recognition should be able to

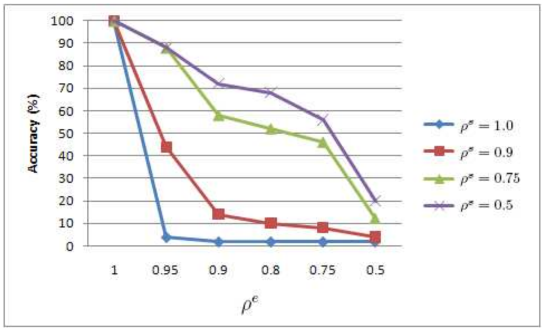

Fig. 9. Accuracies of retrieving with original data set.

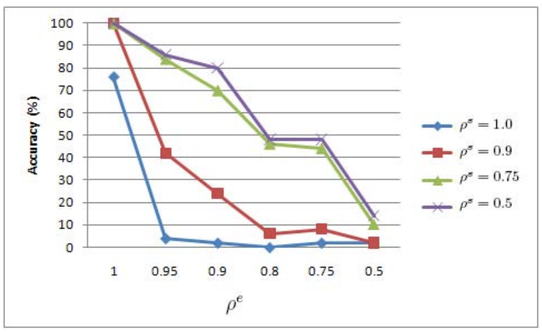

Fig. 10. Accuracies of retrieving with 2% error on event representation.

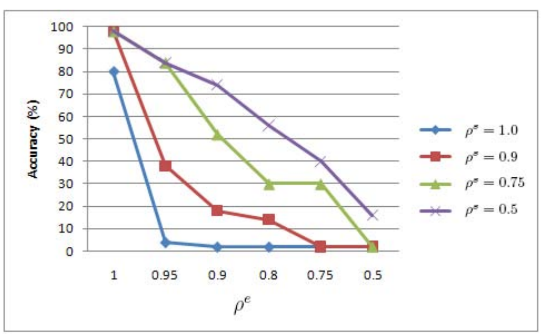

Fig. 11. Accuracies of retrieving with 5% error on event representation.

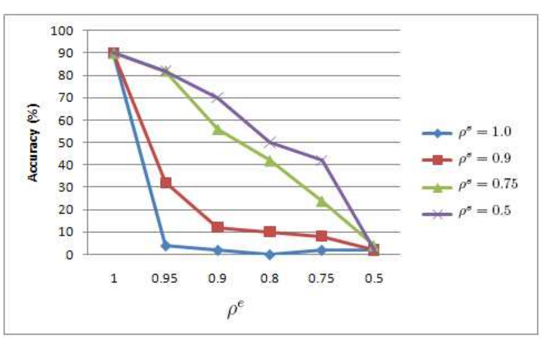

Fig. 12. Accuracies of retrieving with 10% error on event representation.

tolerate minor changes within events and their temporal orders, which is achieved by lowering its vigilance (ρ^s). By tuning vigilance values, the model tackles the challenge of building an efficient memory storage matching scheme as stated in Section II.

F. Retrieving with Noisy Episodes

In this section, we test the model reliability in dealing with the second type of noise. The corresponding noisy data set is derived from the original data set using the method described in Algorithm 5, given the desired rate of noise.

Algorithm 5 (Generation of noisy episodes):

```
FOR EACH sequence S_1 stored
    Randomly select another stored sequence S_2
    Randomly set the value of x, (0 < x < S_1.length)
    Set y = x + n, where n/S_1.length is the desired error rate
    Replace S_1s partial sequence in S_1 indexed by [x, y]
    with the corresponding partial sequence of S_2
```

We test the model with various error rates on sequence representations and the results are shown in Figure 13. Similar to the previous results, to achieve tolerance to high level of noise, the model requires a relatively low vigilance (ρ^s). With a vigilance of 0.5, the model can achieve 100% retrieval accuracy with an error rate as high as 20%. All the tests presented in this session were conducted with $\rho^e = 1.0$.

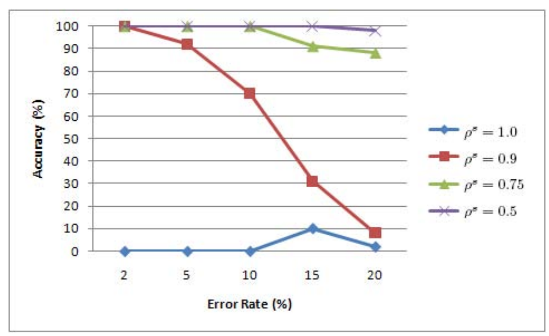

Fig. 13. Accuracies of retrieving with various error rates on sequence representation.

VII. Related Work

Many prior systems model episodic memory as a trace of events and activities stored in a linear order, wherein some operations are designed specifically to retrieve and modify the memory to support specific tasks (e.g. [6]). These approaches are limited to encoding simple sequential trace structure and may not be able to learn complex relations between events and retrieve episodes with imperfect or noisy cues. Our proposed model addresses this issue by representing events as multi-channel activation patterns allowing retrieval based on partial matching. Furthermore, the fusion ARTs fuzzy operations and the complement coding technique enable patterns to be generalized, so that irrelevant attributes of an event can also be suppressed through learning.

Another approach of episodic memory modeling uses the tree structure of a general cognitive architecture (SOAR) to store episodes instead of the linear trace (e.g. [7]). Each node in the memory tree includes some temporal information about its occurrence so that more complex representation can be expressed and episodes can be retrieved based on partial match. However, as it requires to store every snapshot of working memory, the system may not be efficient due to possible large storage of snapshots. In contrast, our EM model clusters both individual events and their sequential patterns based on similarities instead of holding all incoming information in a trace buffer. Our approach inherently comprises more compact storage and efficient processing.

Other models also regard the plausible neural embodiment of episodic memory. Most neural network models of EM use associative networks that store relations between attributes of events and episodes (e.g. [9], [4]). Although they can handle partial and approximate matching of events and episodes with complex relationships, the associative model may still be limited in recalling information based on sequential cues. Some of the existing episodic memory models have attempted to address these challenges, in particular episode formation. SMRITI encodes events as relational structures composed of role-entity bindings [10]. The model attempts to address temporal associations among events within an episode by strengthening the links between events that are "close" in time. The simple sequential learning scheme is insufficient in learning complex sequential information, that may involve repeating events and episodes of varying length. The model also views individual events in separation and does not address the fundamental relations among various events, including possible event clustering and temporal ordering. Our proposed model tackles these issues by employing two levels of fusion ART. The first level deals with repetition by growing separate categories whenever repeated events occur. The second level clusters sequential patterns formed at the first level including repetitions so that various lengths of complex sequential patterns can be learnt at once. Another model called TESMECOR [8] captures complex spatio-temporal patterns and supports retrievals based on degraded cues. Using two neural layers consisting nearly complete horizontal connections, the model distributedly captures events and episodes without clustering. Although it may be the most comparable architecture to our model, our approach offers modularity and flexibility by employing two levels of clustering that may be used by other systems.

Compared with pior work, our proposed EM model handles cues as input patterns which may contain noises or errors. The correct episode can still be recognized and recalled accurately regardless of these imperfect input patterns from the possible noisy or erroneous cues. This capability is achieved by employing the natural decay process on the activations of the selected events, which inherently exhibits a continuous search process towards the best match.

VIII. Conclusion

We have presented a new episode memory model, based on a class of self-organizing neural networks known as fusion Adaptive Resonance Theory and the technique of invariance principle. The model encodes the potentially complex conceptual and spatio-temporal relations among past situations. The stored information can be retrieved with various imperfect cues containing noises and errors.

We have conducted empirical experimental evaluation on the proposed model using a first-person shooting game. Various tests are performed on the built memory model to access its efficiency of possible memory retrieving during the games. Our experimental results show that the model is able to provide a robust level of performance in encoding and recalling events and episodes even with incomplete and noisy cues. This is mainly due to its approximate retrieval using resonance search.

This paper has focused on the learning and retrieval functions within the episodic memory model. As discussed, episodic memory requires interactions with other related cognitive components to reveal its crucial roles and functionalities. Therefore, one immediate extension of our work is to explore its interaction with other memory systems, especially semantic memory.

References

[1] G. Bradski, G. A. Carpenter, and S. Grossberg. STORE working memory networks for storage and recall of arbitrary temporal sequences. *Biological Cybernetics*, 71:469–480, 1994.

[2] G. A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37:54–115, June 1987.

[3] M. A. Conway. *Exploring episodic memory*, volume 18, chapter 1.2, pages 19–29. Elsevier, Amsterdam, 2008.

[4] M. A. Gluck and C. E. Myers. Hippocampal mediation of stimulus representation: a computational theory. *Hippocampus*, 3(4):491–516, October 1993.

[5] S. Grossberg. Behavioral contrast in short-term memory: Serial binary memory models or parallel continuous memory models? *Journal of Mathematical Psychology*, 3:199–219, 1978.

[6] S. T. Mueller and R. M. Shiffrin. REM-II: a model of the development co-evolution of episodic memory and semantic knowledge. In *Proceedings of International Conference on Development and Learning*, volume 5, 2006.

[7] A. Nuxoll and J. E. Laird. Extending cognitive architecture with episodic memory. In *AAAI*, pages 1560–1564. AAAI Press, 2007.

[8] G. J. Rinkus. A neural model of episodic and semantic spatiotemporal memory. In *Proceedings of the 26th Annual Conference of Cognitive Science Society*, pages 1155–1160, Chicago, 2004. LEA.

[9] A. B. Samsonovvich and G. A. Ascoli. A simple newural network model of the hippocampus suggesting its pathfinding role in episodic memory retrieval. *Learning and Memory*, 2005.

[10] L. Shastri. Episodic memory and cortico-hippocampal interactions. *TRENDS in Cognitive Sciences*, 6(4):162–168, April 2002.

[11] B. Subagdja and A.-H. Tan. A self-organizing neural network architecture for intentional planning agents. In *AAMAS (2)*, pages 1081–1088, 2009.

[12] A.-H. Tan, G. A. Carpenter, and S. Grossberg. Intelligence through interaction: Towards a unified theory for learning. In *ISNN (1)*, pages 1094–1103, 2007.

[13] E. Tulving. *Elements of episodic memory*. Oxford University Press, 1983.

[14] D. Wang, B. Subagdja, A.-H. Tan, and G.-W. Ng. Creating human-like autonomous players in real-time first person shooter computer games. In *Proceedings, Twenty-First Annual Conference on Innovative Applications of Artificial Intelligence*, volume to appear, 2009.