Encoding and Recall of Spatio-Temporal Episodic Memory in Real Time

Poo-Hee Chang and Ah-Hwee Tan
School of Computer Science and Engineering
Nanyang Technological University
Singapore 639798
{phchang, asahtan}@ntu.edu.sg

Abstract

Episodic memory enables a cognitive system to improve its performance by reflecting upon past events. In this paper, we propose a computational model called STEM for encoding and recall of episodic events together with the associated contextual information in real time. Based on a class of self-organizing neural networks, STEM is designed to learn memory chunks or cognitive nodes, each encoding a set of co-occurring multi-modal activity patterns across multiple pattern channels. We present algorithms for recall of events based on partial and inexact input patterns. Our empirical results based on a public domain data set show that STEM displays a high level of efficiency and robustness in encoding and retrieval with both partial and noisy search cues when compared with a state-of-the-art associative memory model.

1 Introduction

Memory is the basis of human intelligence, enabling us to react to current situations based on our past experiences. At the functional level, almost all cognitive processes, such as reasoning, planning, learning and problem-solving, require some form of memory. In this paper, we consider a special form of memory, known as episodic memory [Tulving, 1983], which refers to the record of temporal sequences of events associated with contextual information, in particular people, objects, activities, time, and places. Such type of memory is expected to be a core component of any cognitive system, be it biological or artificial, as it enables the system to recall past events and to adapt its actions from past experience.

The main challenge of modeling episodic memory is to build an efficient storage mechanism for encoding an incoming stream of episodic events consisting of multi-modal sensory as well as contextual information in real time. The episodic memory should allow generalization across events, when required and be scalable and remain plastic (adaptable) to new incoming events. On the other hand, the memory model should support recall of stored events in real time in response to partial or inexact search cues.

Although there has been great interests in the study of episodic memory, most early episodic memory models are largely based on symbolic representation. As they are designed to encode complex relationships between events, they are not able to handle recall using vague or incomplete cues [Ho *et al.*, 2003; Samsonovich and Ascoli, 2005; Nuxoll and Laird, 2007]. Also, by encoding the incoming events without generalization, scaling up the memory storage in a real time environment is a key issue.

Taking a biologically-inspired approach [O'Keefe and Dostrovsky, 1971], this paper presents a computational model of episodic memory, named the Spatio-Temporal Episodic Memory (STEM) model, based on a generalized self-organizing neural network model known as fusion Adaptive Resonance Theory (fusion ART) [Tan *et al.*, 2007]. Fusion ART is a generalization of the Adaptive Resonance Theory (ART) for pattern fusion and association across multimodal pattern channels. By inheriting the ART properties, it is designed to learn cognitive nodes, each encoding the associated information of an event represented across multiple pattern channels, in response to a continual stream of incoming patterns in an online and real time manner.

Although similar neural models of episodic memory, known as the Episodic Memory - Adaptive Resonance Theory (EM-ART) [Wang *et al.*, 2012; Subagdja *et al.*, 2012; Subagdja and Tan, 2015] and the general associative memory (GAM) [Shen *et al.*, 2013], have been developed, they function more as a sequential memory model and lack the explicit learning and representation of time and space. In contrast to the EM-ART and the GAM model, the proposed memory model does not employ a separate *episode* layer. Instead, the information of time is represented explicitly and learned through a *time* input channel. The design of time input channel is inspired from the autobiographical memory model [Wang *et al.*, 2016]. For retrieval of events based on time, we develop a novel memory search algorithm for identifying cognitive nodes, which encode events at a specific time or within a selected time interval. In addition, a dedicated "place" pattern channel is incorporated for learning spatial representation of real world position coordinates.

In an application context, the proposed STEM model is intended to augment a cognitive system equipped with a sensory front-end. In the most general case, the system may have multi-modal sensing capabilities, notably visual sensing, object recognition, and scene analysis. The incorporation of episodic memory will thus enable the cognitive system to memorize and maintain a record of the happenings on a scene with both temporal and spatial information, which can then be subsequently analyzed. For evaluation, we extract event-based information from a public domain video data set [Fisher *et al.*, 2005]. Compared with the GAM model, our experiments show that the STEM model is able to encode the over 40,000 extracted events in seconds and supports recall of the stored events using partial and noisy search cues.

2 Related Work

One common approach to modeling episodic memory is to store them as a trace of events and activities in a linear order, wherein some operations are designed specifically to retrieve and modify the memory to support particular tasks [Vere and Bickmore, 1990; Ho *et al.*, 2003; Mueller and Shiffrin, 2006]. These models however are limited to processing simple sequential trace structure and may not be able to learn complex relations between events or to retrieve memory with imperfect or noisy cues. Another work extends SOAR as an existing generic cognitive architecture with mechanisms to maintain sequential traces of production system operations [Nuxoll and Laird, 2007]. It makes use of SOAR built-in operations to conduct complex memory encoding and retrieval. One critical issue of this approach is the requirement of effective partial matching to deal with incomplete and possibly degraded cues for retrieval [Nuxoll, 2007].

Another approach using neural networks aims to model episodic memory with inherent support for partial matching and pattern generalization. Grossberg and Merrill combine ART (Adaptive Resonance Theory) neural network with *spectral timing* encoding to model timed learning in hippocampus [Grossberg and Merrill, 1996]. However, this model is only made specifically to handle learning timed responses in episodic memory. Shastri focuses on complex relational representation in SMRITI, a neural architecture for episodic memory [Shastri, 2002]. The model can handle role-entity bindings in which retrieval cues can involve transient values for retrieval using partial information. Although SMRITI has already supported complex inferences on top of the relational representation, it omits the temporal or sequential relations between items altogether.

Instead of modeling hippocampus, some other neural models focus on handling spatio-temporal or multi-modal patterns. For example, TESMECOR [Rinkus, 2004] rapidly stores spatio-temporal patterns in a distributed manner while providing a robust retrieval mechanism that supports complex sequential representation. Starzyk and He develop a neural network model of anticipation-based spatio-temporal learning that can store and retrieve complex sequences as units of episodes [Starzyk and He, 2009]. Based on a neural model for complex sequential learning and production [Wang and Arbib, 1993], the model can tolerate errors from search cues and support partial matching. Shen et al. propose the general associative memory (GAM) model [Shen *et al.*, 2013] that stores both static and temporal sequence information. GAM is also able to tolerate noisy and partial cues. However, these models function more like sequential memory and do not encode the time information explicitly.

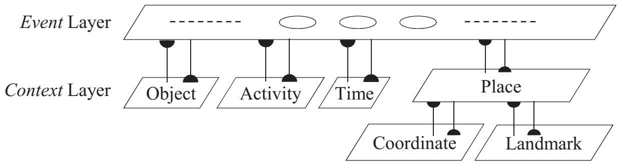

Figure 1: The STEM architecture

Another recent approach, including Neural Turing Machines [Graves *et al.*, 2014] and Memory Networks [Weston *et al.*, 2014], makes use of recurrent neural networks to store episodic memory of sequential inputs. However, these methods present a computational challenge for real-time learning as described in [Kaiser and Sutskever, 2015] and [Angelov and Sperduti, 2016].

3 The STEM Architecture

Using self-organizing neural networks as the building block, the proposed memory model is designed to integrate multimodal episodic memory involving audio, visual imagery, self and other contextual information.

Figure 1 shows the overall architecture of the Spatiotemporal Episodic Memory (STEM) model. It can be viewed as two fusion ART models connected in a hierarchical manner. The *context* layer consists of four input fields, namely the *object* field for representing the presence of specific people and objects in the event; the *activity* field for representing the occurred activity; the *time* field for representing the time of occurrence; and the *place* field for location representation.

The *place* field in the *context* layer receives inputs from two lower level fields, namely the *coordinate* field for representing the real world position, and the *landmark* field for representing a specific region, such as "entrance", "lift", and "reception" etc. The *place* field learns the spatial representation in the form of space category nodes by using the fusion ART algorithm. Each space category node learned represents a group of visited positions. The learned spatial representation is then used as an input into the *event* field in the *event* layer. Together with the *object*, *time* and *activity* input fields from the *context* layer, the *event* field learns recognition nodes in response to the presented sensory and contextual information.

3.1 Fusion ART

The episodic memory model proposed in this paper is based on fusion ART [Tan *et al.*, 2007] which can be viewed as an Adaptive Resonance Theory (ART) neural network [Carpenter and Grossberg, 2003] with multiple input fields (Figure 2). The network is designed to learn cognitive nodes encoding groups of multi-modal input patterns and support the recognition and recall of the stored patterns. By inheriting the ART properties, fusion ART performs fast and stable learning in response to a continual stream of input patterns, and learns new patterns incrementally. There is no separate phase of operations for learning and recall. For completeness, a summary of the network dynamics is given below.

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

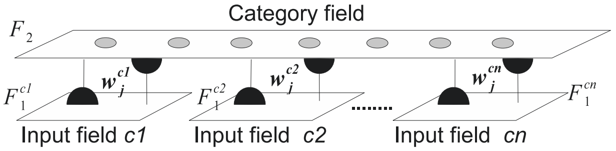

Figure 2: Fusion ART

$\mathbf{I}^k$ 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 field: Let F_2 denote the category field. 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

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

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

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

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 if the *match function* m_J^k of the chosen node J meets its vigilance criterion such that

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

If any of the vigilance constraints is violated, mismatch reset occurs and 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.

Template learning: Once a resonance occurs, for each channel k, the weight vector $\mathbf{w}_J^k$ is modified by the following learning rule:

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

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 $\mathbf{x}^{k(\text{new})} = \mathbf{w}_J^k$.

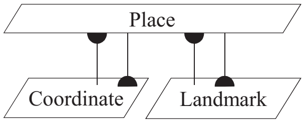

Figure 3: Spatial representation architecture

A fusion ART network, consisting of different input (output) fields and a category field is a flexible architecture. In this paper, we show how fusion ART can be used for learning spatial and episodic event representation.

3.2 Learning Spatial Representation

As shown in Figure 3, the place field learns spatial representation of the scene via the fusion ART mechanism with two input fields, namely the *coordinate* field and the *landmark* field, which represent the physical position and symbolic location respectively.

The *coordinate* field represents a position in the 2D map which can be represented with continuous values along the X-Y axis. These values are normalized to [0,1] range. The *landmark* field is a symbolic representation of the locations.

Using the fusion ART dynamics, the place field is able to learn and generalize spatial representation of the environment in response to coordinate positions and landmark symbols. The overall process is summarized in Algorithm 1.

Algorithm 1 Spatial representation encoding

```
Present the coordinate and landmark vectors
Perform code activation in the place field              ▷ see (1)
repeat
    Perform code competition and template matching      ▷ see (2)
until resonance occurs                                  ▷ see (3)
Perform template learning                               ▷ see (4)
```

3.3 Learning Event Representation

After the cognitive nodes in the place field are learned, we proceed to encode the events by presenting the input vectors into their respective input fields as shown in Algorithm 2.

Algorithm 2 Event encoding

```
for each event in the schedule do
    Present the coordinate and landmark vectors
    Perform code activation/competition in the place field ▷ see (1)
    Present the object, activity, time, and place vectors
    Perform code activation in the event field           ▷ see (1)
    repeat
        Perform code competition and template matching   ▷ see (2)
    until resonance occurs                               ▷ see (3)
    Perform template learning                            ▷ see (4)
end for
```

3.4 Retrieval of Events

After the events are encoded, we are able to retrieve the memory based on the retrieval cues. A retrieval cue is a stimulus to facilitate activation of memory. The content of a retrieval cue is similar to the input event vectors with possibly incomplete or erroneous attribute values. As presented in Algorithm 3, upon presenting a retrieval cue, the algorithm retrieves its winner(s) based on the maximum choice value. With the winner node(s) known, we are able to conduct activity readout of the event.

Algorithm 3 Event retrieval

```
Input: Retrieval cue, and the trained model
Activate the *place* field with *coordinate* and *landmark* fields
Select the winner code(s) in *place* field
Activate the *event* field with *object*, *activity*, *time* and *place* fields
Select a list of cognitive nodes in *event* layer
return List of winners with highest choice value
```

3.5 Temporal Representation

The time stamps of events are encoded and presented in the time input field of the STEM model for time representation. Time, represented in unit of seconds, is normalized into a real value between 0 and 1.

Through a direct memory access procedure, STEM searches for the cognitive node which best matches with the given search cue. Given a specific time stamp as the search cue, the following lemma shows the code activation and competition process of STEM will select the cognitive node encoding an event with a time stamp closest to the chosen time.

Direct Time-based Memory Access Property: During direct memory access, given the complemented coded temporal activity vector $\mathbf{x}^{ct} = (t, 1-t)$, the fusion ART code activation and competition process will select the cognitive node J with the weight vector $\mathbf{w}_J^{c1}$ closest to $\mathbf{x}^{ct}$.

Proof (by Contradiction): Given a complement coded temporal activity vector $\mathbf{x}^{ct} = (t, 1-t)$, suppose STEM selects a F_2 node J and there exists another F_2 node K of which the weight vector $\mathbf{w}_K^{ct} = (t_K, 1-t_K)$ is more similar to $\mathbf{x}^{ct}$ than $\mathbf{w}_J^{ct} = (t_J, 1-t_J)$. As $\mathbf{w}_K^{ct}$ is more similar to $\mathbf{x}^{ct}$ than $\mathbf{w}_J^{ct}$, it means $|t - t_K| < |t - t_J|$. Without loss of generalization, suppose $t < t_K < t_J$, we derive that the choice functions (1) of F_2 nodes K and J are

$$T_K = \frac{|\mathbf{x}^{ct} \wedge \mathbf{w}_K^{ct}|}{\alpha^{ct} + |\mathbf{w}_K^{ct}|} = \frac{t + 1 - t_K}{\alpha^{ct} + 1} \quad (5)$$

$$T_J = \frac{|\mathbf{x}^{ct} \wedge \mathbf{w}_J^{ct}|}{\alpha^{ct} + |\mathbf{w}_J^{ct}|} = \frac{t + 1 - t_J}{\alpha^{ct} + 1} \quad (6)$$

As $t_K < t_J$, the above condition implies that $T_K^c > T_J^c$, which means node K should be selected by STEM instead of node J (Contradiction). *[End of Proof]*

In addition to providing direct access to stored events based on a time stamp, STEM further supports memory access to stored events with time stamps falling into a selected time interval. The lemma below summarizes such a property.

Direct Time Interval Memory Access Property: During memory recall, given the complemented coded temporal activity vector $\mathbf{x}^{ct} = (t_1, t_2)$, the fusion ART code activation and competition process will select a set of cognitive nodes $J_1, J_2, \ldots, J_N$, each of which encodes an event with a time stamp $t \in [t_1, 1 - t_2]$.

Proof: Given a complement coded temporal activity vector $\mathbf{x}^{ct} = (t_1, t_2)$, suppose there exists a F_2 node J encoding weight vector $\mathbf{w}_J^{ct} = (t_J, 1 - t_J)$, where $t_J \in [t_1, 1 - t_2]$, we can derive that $t_1 < t_J$ and $t_2 < 1 - t_J$. As such, the choice function value of the F_2 node J is given by

$$\frac{|\mathbf{x}^{ct} \wedge \mathbf{w}_J^{ct}|}{\alpha^{ct} + |\mathbf{w}_J^{ct}|} = \frac{t_1 + t_2}{\alpha^{ct} + 1}, \quad (7)$$

which is the maximal possible given the temporal activity vector. Therefore, the node J will be selected by the memory access process. *[End of Proof]*

4 Experiments

4.1 The CAVIAR Data Set

We use the CAVIAR data set [Fisher, 2004; Fisher *et al.*, 2005]. It contains 28 videos of a lobby entrance, together with hand-labeled ground truths of the surveillance activities. Information is provided to map from pixels to real world positions using ground plane homography. For our purpose, we use the ground truth to provide the objects, activities and positional information to be encoded into our memory model. Since the data set is small (26,419 frames), we augment the data by treating each frame of the video as one second. As the data set does not contain time information, we label each video frame with a specific time in the day.

4.2 Encoding of Memory

In the CAVIAR data set, each frame may involve multiple objects or groups. To encode such events, we define a separate event for each object or group, in each frame instance. There are a total of 46,125 events. The encoding schemes of the input fields are summarized below.

1. The **Object vector** indicates the presence of up to nine persons or objects, each represented by a binary attribute. To emulate the distributed representation of visual objects in the human brain [Valdez *et al.*, 2015], it is further encoded with a (5,1) repetition code.
2. The **Activity vector** indicates the presence of one of the nine different activities, each symbolized with a 7x7 binary image (Figure 4). Each pixel is either 1 (black) or 0 (white). We use the data set's "situation" labels to identify the current activity of the object(s).
3. The **Time vector** contains a normalized real-valued attribute, obtained by dividing the current time of the day in seconds by the total number of seconds in a day.
4. The **Coordinate vector** represents the estimated real world position quantized to the nearest feet. The pixel position is first extracted from the bounding box provided in the data set. By using homography, the pixel positions are mapped to the real world points with respect to the X and Y axis. It is further normalized into a real value between 0 and 1 for each axis.
5. The **Landmark vector** represents one of the ten region of interest (Figure 5) for this domain. The ten different regions are represented by a vector of seven bits, encoded similarly to a seven-segment digit display.
6. The **Place vector** represents the winner node(s) of the place field activated by the coordinate and landmark vectors. The vector is constructed based on the indices of the learned place nodes.

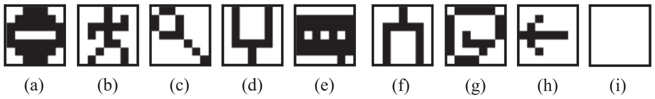

Figure 4: Iconic patterns for encoding types of activities: (a) inactive; (b) moving; (c) browsing; (d) joining; (e) interacting; (f) split up; (g) fight; (h) leaving object; (i) none.

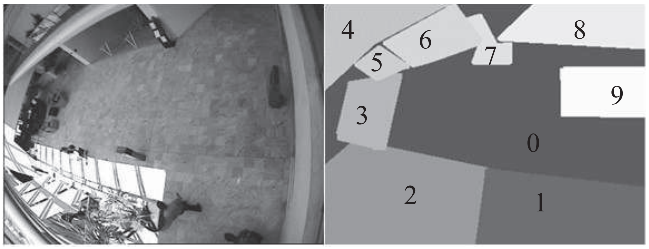

Figure 5: Left: A frame from CAVIAR data set, showing the entrance of a lobby. Right: Landmark regions with labels.

Before the STEM model encodes the event input patterns, the place field is first trained with the coordinates and their corresponding landmark labels. The spatial representation is learned with vigilance values of $\rho = 0.99$, the contribution parameter $\gamma = 0.5$ and choice parameter $\alpha = 0.001$ on both the coordinate and landmark fields. See Figure 6 for visualization of the learned spatial nodes.

After the cognitive nodes in the place field are learned, we proceed to encode the events by presenting the input vectors to their respective input fields. The events are presented sequentially based on the constructed schedule of a day.

For performance comparison, we have chosen the general associative memory (GAM) [Shen *et al.*, 2013], a recently proposed spatio-temporal memory model which has shown superior performance in comparison with traditional memory models in handling partial and noisy retrieval cues. As the GAM learns with a single pattern channel, it is trained by concatenating all contextual inputs of an event into one pattern. The GAM will require noisy training inputs to calculate the classification thresholds. However, due to the large number of events, only exact event inputs without noise are trained. During recall, thresholding is removed so the GAM will always recall at least one nearest matched event. In both models, aging and removal of the memory nodes are not used.

4.3 Retrieval of Events

After the events are encoded into the memory models, experiments are conducted to retrieve events based on retrieval cues. For a fair comparison, in each retrieval trial, we only consider the top recalled event returned by each model. A search is successful if the recalled event corresponds with the original event.

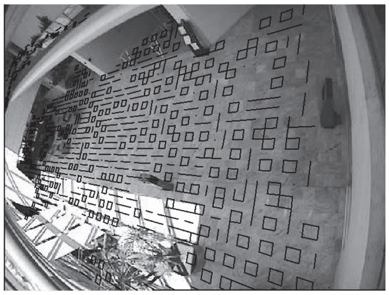

Figure 6: Visualization of the learned spatial nodes in place field.

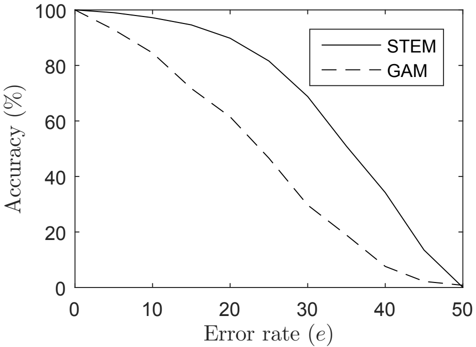

Figure 7: Retrieval accuracy based on noisy cues.

In our experiments, noisy and partial cues are tested for retrieval. For each cue type, 1,000 randomly selected events from the data set are used to form the cues and be tested. The following types of cues and the results are summarized below.

Noisy cue: For each event attribute, it is subjected to an error rate $\boldsymbol{e} \in [0, 100]$. For binary valued fields, there is a probability of $\boldsymbol{e}/100$ for its attributes to be toggled, such as $x_{noisy} = 1 - x$. For continuous attributes found in *time* and *coordinate* fields, there is a probability of $\boldsymbol{e}/100$ that its attributes are subjected to a Gaussian noise. Our experiments are conducted with error rates of 0 to 50. A noisy cue with $\boldsymbol{e} = 0$ is in fact an exact cue. As shown in Figure 7, while both models are able to achieve 100% recall with zero noise, the STEM model is highly robust with noisy cues up to 20% of error.

Partial cue: The partial cue contains missing attributes from one or more field vectors such as *object*, *activity*, *time*, *coordinate* and *landmark*. We have conducted our experiment with partial cues constructed from all five down to one field vector chosen randomly. For the STEM model, the missing field attributes and their complement values are assigned to zeroes. For the GAM model, the missing field attributes are set to zero. For evaluation, two performance measures are used, namely the retrieval accuracy of the recalled events with respect to the corresponding target events; and recall error in terms of the normalized Euclidean distance (ED) between the recalled event activity patterns and the corresponding target event activity patterns.

As shown in Figure 8, the STEM model also performs significantly better than GAM with partial field cues. This can be attributed to STEM's capability in handling multi-channel input patterns whereas GAM performs pattern matching of the aggregated input patterns as a whole. In addition, STEM represents the *place* field based on the coordinate and landmark inputs. Therefore the *place* field can still recall a place representation even with incomplete spatial information.

4.4 Retrieval of Episodes

For the STEM model, an episode can be dynamically retrieved based on the direct time interval memory access property (Section 3.5). Given a time interval between t_1 and t_2, the time cue with complemented coded vector is $[t_1, 1 - t_2]$. Experiments are conducted by randomly picking 1,000 pairs

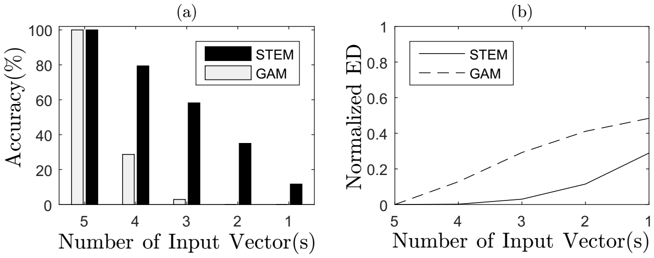

Figure 8: (a) Retrieval accuracy and; (b) Recall error in response to partial cues.

of t_1 and t_2 time-only cues. We are able to retrieve 100% accuracy on all events happened in between time t_1 and t_2.

GAM is also able to retrieve episodes with exact search cues at 100% accuracy. However, a successful episodic retrieval will be dependent on retrieving the first event correctly as the GAM will retrieve the event sequences thereafter. As GAM is unable to accurately retrieve the correct event based on time-only partial cues (shown in Figure 8), it is unable to retrieve episodes with time-only partial cues.

5 Analysis of Space and Time Requirement

The space and time analysis addresses the feasibility of the model to operate in real-time on a typical computer. For space requirement, we compute the estimated size of computer memory needed to store the model's data. For time, we regard the performance deadline for real-time requirement as one second, which is the intended sampling rate. We will compare the event encoding computation performance of our model with the original ART model.

5.1 Space Requirement

We make assumptions of the general behavior of the crowd in a specific place, to determine the number of events that the system will store per day. We have chosen the lobby as a scenario so that the parameters used in the assumption can be put into a realistic context.

Assumption for the Lobby Scenario

We assume that the daily human traffic in the lobby is **1,000 people**, with the average duration in the scene for each person at **30 seconds**. The physical area of the lobby is assumed to be **750 square feet**. The vector size in the *object*, *activity*, *time*, *coordinate* and *landmark* fields are 250, 49, 1, 2 and 7 respectively. The *object* field is represented by 50 attributes with (5,1) code repetition. The *activity* field is derived from the 7x7 image, while the *landmark* field is representing ten landmarks with seven-segment encoding.

Spatial Memory Estimation

We assume that the maximum number of nodes formed in the place layer is equal to the number of unique combinations of the coordinate and landmark values. As such, the maximum number of nodes in the spatial representation is also the size of the area, which is 750. We use two doubles (16 bytes) to represent an attribute with its complement. The memory size encoded by each spatial node is thus $\{2+7\}\times16 = 144$ bytes. The total memory size encoded by the place field is therefore $750 \times 144 = 108,000$ bytes or 0.103 MB.

Event Memory Requirement

We assume the number of events in a day is the number of times people appeared at one-second intervals. Therefore, the expected number of daily events is the product of the average daily human traffic and the average duration in the scene, i.e. $30,000$. The estimated memory size encoded by each event node, including complement, is $13,888$ bytes ($\{250 + 49 + 1 + 750\} \times 16$). Finally, the estimated memory size of the daily events is 480.65 MB.

Overall Memory Estimation Analysis

On the whole, the required computer memory needed to store all events, together with the spatial representation under the lobby scenario is 480.75 MB. As such, a typical consumer hard disk (1 TB) is sufficient to deal with the scenario over several years.

5.2 Time Requirement

In the standard Fusion ART model, the worst-case time complexity of encoding a new node (when no matching node is found) is $O(tn^2)$, where t is the number of attributes in the input fields, and n is the number of event nodes. The quadratic complexity is due to computing the winner nodes for generalization. With generalization, encoding over 22,928 out of 46,126 events, will take more than one second to store one event. In our implementation, we disabled the search process as no generalization of events are needed. Thus, its computation complexity is just $O(t)$. Consequently, the time taken to encode an event is 367.45 microseconds on average, which is no greater than the sampling rate of one second.

During retrieval of events, we use a vigilance of zero. This means that winner selection is only based on code activation, without the vigilance check. The computation complexity of this retrieval is $O(tn)$. It takes 341 milliseconds on average in our test to retrieve an event using retrieval cues, which is well adequate for real time use.

6 Conclusion

We have designed and implemented an episodic memory module called STEM that performs explicit encoding of time, space and contextual information of objects, people, and their activities. In this paper, we have demonstrated the encoding and recall capabilities of the model based on event-related information extracted from a public domain video data set. Our experiments show that the STEM model is able to support robust recall of the stored events in response to partial and noisy search cues, in comparison with a recent spatio-temporal associative memory model.

Going forward, we shall integrate the episodic memory model to a visual cognitive system deployed in a real time environment. In addition, we shall explore the use of code compression and pruning so that we can mitigate the issue of ever growing memory in the real world.

Acknowledgments

This research is supported by the DSO National Laboratories under research grant DSOCL16006.

References

[Angelov and Sperduti, 2016] Plamen Angelov and Alessandro Sperduti. Challenges in deep learning. In *Proceedings of the 24th European Symposium on Artificial Neural Networks (ESANN). i6doc. com*, pages 489–495, 2016.

[Carpenter and Grossberg, 2003] Gail A Carpenter and Stephen Grossberg. Adaptive Resonance Theory. In Michael A Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 87–90. Cambridge, MA: MIT Press, 2003.

[Fisher *et al.*, 2005] Robert Fisher, Jose Santos-Victor, and James Crowley. CAVIAR: Context aware vision using image-based active recognition, 2005.

[Fisher, 2004] Robert B Fisher. The PETS04 surveillance ground-truth data sets. In *Proceedings of the 6th IEEE international workshop on performance evaluation of tracking and surveillance*, pages 1–5, 2004.

[Graves *et al.*, 2014] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv:1410.5401*, 2014.

[Grossberg and Merrill, 1996] Stephen Grossberg and John W.L. Merrill. The hippocampus and cerebellum in adaptively timed learning, recognition, and movement. *Journal of Cognitive Neuroscience*, 8(3):257–277, 1996.

[Ho *et al.*, 2003] Wan Ching Ho, Kerstin Dautenhahn, and Chrystopher L Nehaniv. Comparing different control architectures for autobiographic agents in static virtual environments. In *International Workshop on Intelligent Virtual Agents*, pages 182–191. Springer, 2003.

[Kaiser and Sutskever, 2015] Łukasz Kaiser and Ilya Sutskever. Neural GPUs learn algorithms. *arXiv:1511.08228*, 2015.

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

[Nuxoll and Laird, 2007] Andrew M. Nuxoll and John E. Laird. Extending cognitive architecture with episodic memory. In *AAAI*, pages 1560–1564. AAAI Press, 2007.

[Nuxoll, 2007] Andrew M Nuxoll. *Enhancing Intelligent Agents with Episodic Memory*. PhD thesis, University of Michigan, 2007.

[O'Keefe and Dostrovsky, 1971] John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain research*, 34(1):171–175, 1971.

[Rinkus, 2004] Gerard 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.

[Samsonovich and Ascoli, 2005] Alexei V Samsonovich and Giorgio A Ascoli. A simple neural network model of the hippocampus suggesting its pathfinding role in episodic memory retrieval. *Learning and Memory*, 12(2):193–208, 2005.

[Shastri, 2002] Lokendra Shastri. Episodic memory and cortico–hippocampal interactions. *TRENDS in cognitive sciences*, 6(4):162–168, April 2002.

[Shen *et al.*, 2013] Furao Shen, Qiubao Ouyang, Wataru Kasai, and Osamu Hasegawa. A general associative memory based on self-organizing incremental neural network. *Neurocomputing*, 104:57–71, 2013.

[Starzyk and He, 2009] Janusz A Starzyk and Haibo He. Spatio–temporal memories for machine learning: A long-term memory organization. *IEEE Transactions on Neural Networks*, 20(5):768–780, 2009.

[Subagdja and Tan, 2015] Budhitama Subagdja and Ah-Hwee Tan. Neural modeling of sequential inferences and learning over episodic memory. *Neurocomputing*, 161:229–242, 2015.

[Subagdja *et al.*, 2012] Budhitama Subagdja, Wenwen Wang, Ah-Hwee Tan, Yuan-Sin Tan, and Loo-Nin Teow. Memory formation, consolidation, and forgetting in learning agents. In *Proceedings, Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 1007–1014, 2012.

[Tan *et al.*, 2007] Ah-Hwee Tan, Gail A Carpenter, and Stephen Grossberg. Intelligence through interaction: Towards a unified theory for learning. In *International Symposium on Neural Networks*, pages 1094–1103. Springer, 2007.

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

[Valdez *et al.*, 2015] André B Valdez, Megan H Papesh, David M Treiman, Kris A Smith, Stephen D Goldinger, and Peter N Steinmetz. Distributed representation of visual objects by single neurons in the human brain. *Journal of Neuroscience*, 35(13):5180–5186, 2015.

[Vere and Bickmore, 1990] Steven Vere and Timothy Bickmore. A basic agent. *Computational intelligence*, 6(1):41–60, 1990.

[Wang and Arbib, 1993] DeLiang Wang and Michael A Arbib. Timing and chunking in processing temporal order. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(4):993–1009, 1993.

[Wang *et al.*, 2012] Wenwen Wang, Budhitama Subagdja, Ah-Hwee Tan, and Janusz A Starzyk. Neural modeling of episodic memory: Encoding, retrieval, and forgetting. *IEEE transactions on neural networks and learning systems*, 23(10):1574–1586, 2012.

[Wang *et al.*, 2016] Di Wang, Ah-Hwee Tan, and Chunyan Miao. Modeling autobiographical memory in human-like autonomous agents. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 845–853, 2016.

[Weston *et al.*, 2014] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv:1410.3916*, 2014.