# ROBOTIC DREAMS: A COMPUTATION JUSTIFICATION FOR THE POST-HOC PROCESSING OF EPISODIC MEMORIES.

TROY DALE KELLEY

*Human Research and Engineering Directorate, U.S. Army Research Laboratory,*
*Aberdeen, MD 21005, USA*
*troy.d.kelley6.civ@mail.mil*

1. **As part of the development of the Symbolic and Sub-symbolic Robotics Intelligence System (SS-RICS) we have implemented a memory store to allow a robot to retain knowledge from previous experiences. As part of the development of the event memory store, justification for an off-line, unconscious memory process was tested. Three strategies for the recognition of previous events were compared. The first strategy stored all memories and searched all of the memories for a match to the current event. The second strategy searched memories while an event was taking place and started the search with the most recent memory first. Finally, a third strategy post-processed all memories using pruning, abstraction, and cueing. Pruning removed memories, abstraction used categories to reduce metric information and the cueing process provided pointers for the recognition of episodes. We found that post-processing memories as an unconscious process was the most efficient strategy. This computational implementation provides a justification for the post-processing of memories as an efficient means of memory retrieval.**

*Keywords*: Episodic memory, robotics, dream, sleep, novelty.

## 2. Introduction

The Human Research and Engineering Directorate (HRED) of the U.S. Army Research Laboratory (ARL) is developing a robotics system called the Symbolic and Sub-symbolic Robotics Intelligence Control System (SS-RICS) [Kelley, 2006]. The Cognitive Robotics Team of HRED is conducting a series of field tests to implement SS-RICS onto a variety of robotics systems. We have based SS-RICS on the Adaptive Character of Thought-Rational (ACT-R) [Anderson and Lebiere, 1998] cognitive architecture. Implementing

SS-RICS on a robotics platform required changes and additional functionality that expanded the learning and developmental capabilities of the existing architecture. Specifically, while ACT-R included modules for handling declarative memories and procedural memories, it did not include a specific episodic memory store. We developed such an episodic memory store to allow a robot to learn from its experiences. As part of this development process, we discovered that consolidating episodic memory as a post-hoc process was an advantageous methodology as compared to other real-time strategies. During the post-hoc processing of episodic memories, we added an associative cue to anticipated episodes which proved to be more beneficial, in terms of speed, than trying to remember episodes in real time without the cue. In this manuscript we will discuss a computational justification for the unconscious, post-hoc, memory process we implemented.

## 3. Background

Why do we sleep? Why would an organism allow itself to lapse into an unconscious state - removing itself from real world interactions, leaving itself susceptible to attack from predators, unable to react to potentially dangerous situations - unless the process was extremely beneficial? From a cognitive perspective, there must be some benefit to sleep which outweighs the hazards associated with such a dangerous unconscious state.

Humans exist or experience the world in three types of consciousness: waking, Non-Rapid Eye Movement (REM) sleep, and REM sleep [Siegel, 1996]. Sleep in humans has primarily two stages to include REM sleep (where dreaming occurs) and non-REM sleep [Aserinsky and Kleitman, 1953]. While sleep in humans appears to be a non-conscious state, the brain is actually very active during the sleeping process [Hobson, 1988]. The dreaming mind is processing information while perceptual information from the outside world is simultaneously being blocked [Crick and Mitchison, 1983]. This exclusion of outside stimuli would appear to be a dangerous unconscious state were it not for some kind of resultant beneficial processing. What is the benefit from this unconscious state?

Early theories concerning the function of sleep characterized sleep as "unlearning" or that, "we dream in order to forget" [Crick and Mitchison, 1983]. Crick and Mitchison [1983] proposed that noisy or irrelevant dream content was activated within the brain during a dream so that it could be "unlearned." This hypothesis that random events were activated during sleep as dream content so that they could be "unlearned" was consistent with the "activation-synthesis" model of sleep [McCarley & Hobson, 1977; Hobson, 1988; Hobson, et al., 2000]. This hypothesis was supported by studies of dream content, which appeared to suggest that dreams were essentially random thoughts [Wolf, 1994]. However, other researchers noted that dreams are not completely random and that dreams were often repetitive or contained the same general themes [Domhoff, 1993]. Moreover, an extensive study of 58 young adults noted that most dreamers were able to give, "a detailed account of realistic situations" and "people caught up in very ordinary activities"

[Snyder, 1970]. So data indicated that dreams are reenactments of everyday situations and less random than might have been suggested by Freudian theory or the later activation-synthesis theory [Domhoff, 1993].

Episodic memories have been identified as an important component to cognition and learning [Tulving, 1983] as well as being involved in dreaming and sleep [Fosse et al., 2003]. Studies indicate that sleep plays an important role in memory reorganization, especially episodic memories stored in the hippocampus [Pavlides & Winson, 1989; Wilson & McNaughton, 1994]. Animal studies indicate that REM sleep increases following training sessions for both cats and rats; and that sleep deprivation after training sessions impairs the retention of previously presented information [Pearlman, 1979]. Several studies of human sleep support the conclusion that episodic memories are replayed during dreaming usually from the preceding day's events (Cavallero & Cicogna, 1993; Arkin & Antrobus, 1978; De Koninck & Koulack, 1975].

In 1995, Squire and Alvarez noted that memory consolidation might occur during sleep and that "slow wave sleep" contained some "interesting properties" that would make it a candidate for memory consolidation during sleep. They also noted that "consolidation occurs when the neocortical representations are repeatedly co-activated by the medial temporal lobe." Additionally, they note that memory consolidation is a process by which memory becomes independent of the hippocampus (which contains episodic information) and that episodic information is abstracted away from the hippocampus during memory consolidation. This abstraction process of episodic memory was the computational goal for this research.

## 4. Computational implementations of sleep

Unfortunately, the post processing of episodic memory has not been extensively studied or implemented in computational cognitive models, however, Nuxoll and Laird [2007] implemented an episodic memory system in Soar. This implementation was designed to increase the speed of retrievals and show the functional usefulness of episodic memory as a cognitive structure. Similar to our study, they improved memory retrieval times by using cues and pointers for the indexing of episodic memory retrieval. However, their methodology was not executed as a post-hoc process.

Others studies, most notably, Alvarez and Squire [1994], developed a simple neural network model of memory consolidation which used two interacting networks to simulate working memory and Long Term Memory (LTM). The working memory components (hippocampus) of the model had its connection strengths change very quickly, and were short lasting, while the LTM components (neocortical) of the model had its connection strengths change slowly and were long lasting. From this relatively simple model of

memory consolidation the authors were able to reproduce results similar to animals studies and produce temporally graded retrograde amnesia [Alvarez and Squire, 1994].

In Zhang's [2009] neural network model of sleep episodic sequences are replayed either in order, or in random sequences, to simulate the random nature of dream sequences. The results replicated functional aspects of dreams and sleep. Using episodic memories can also be seen as a form of case based reasoning. There have been case based reasoning models which performing learning tasks in "batch mode" during "off-peak" hours [Ram and Santamaria, 1997], however, newer case based reasoning models operate in real time to avoid the delay from learning in batch mode [Sharma et al., 2007].

As part of computational implementation of cognition for SS-RICS, we were interested in using episodic memory as part of a learning process. We were especially interested in quickly retrieving memories during events in order to match previous memories to current events. We were also interested in remembering certain events as being novel. From this initial set of criteria we developed three strategies for memory retrieval. These strategies included 1) *Complete Memory Search*: store every event and retrieve the best match while a novel event was taking place  2) *Recent Memory Search*: storing only novel events and searching recent novel events first while a novel event was happing 3) *Episodic Indexing*: Storing both novel and boring events and post process these events to improve the retrieval process. Additionally, searching the most recent novel events first.

In order to process episodic memories from the robot, we were initially interested in identifying the memories as novel or unusual, based on an algorithm developed for landmark identification [Kelley & McGhee, 2013]. The novelty algorithm captures a window of data of a specific length and uses a histogram of correlations to determine the amount of change over the course of the window. If there was a significant amount of change across the window the histogram registers very few correlations around 1 - signifying a novel event. If there was very little change, the histogram will show a large number of correlations around 1 - indicating boredom. However, this algorithm was too slow in determining change in the environment so instead we used a movement detection algorithm which indicated change much faster and was better suited for our first set of experiments. The movement algorithm sensed movement immediately while the novelty algorithm took several seconds (a window) worth of data to determine a novel scene. Ideally, we would like to transition from the use of the movement algorithm in the current study to the use of the novelty algorithm in future studies as more information is learned and gathered as events.
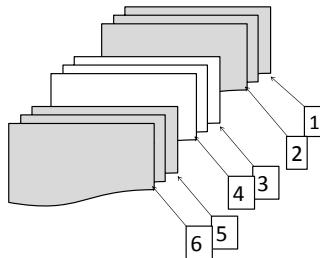
## 5. Procedure



Fig. 1. A sample event. The beginning and ends of different episodes are referenced by the numbers and elaborated in the text as Episodic Segment Numbers (ESNs) to aid in the explanations.

We defined an event as a collection of episodes (see Figure 1). An episode was considered an instance of data within an event. Any number episodes could add up to a single event. Figure 1 is one event with a collection of nine episodes. The numbers in Figure 1 correspond to important starting and ending points for the entire event. The critical points are referenced in the manuscript as Episodic Segment Numbers (ESN) to aid in explanation. In Figure 1, the dark segments represent static episodes (no movement) and the light segments represent movement episodes. Specifically, ESN 1 is the beginning of a static episode; ESN 2 is the end of the first group of static episodes; ESN 3 is the beginning of the movement episode; ESN 4 is the end of the group of movement episodes; ESN 5 is the beginning of the next group of static episodes and ESN 6 is the end of the last group of static episodes.

## 5.1.  *Strategy Comparison*

The task for the robot was simple.  The robot was to recall the presentation of a target (a yellow flashlight) following the presentation of a different target (a red tool case) (see Figure 2).  Note that the robot was not making a semantic determination; instead the robot was determining whether or not it had seen some particular object, with some particular set of features, at a particular point in time.  The semantic aspects were not important for the event to be remembered or anticipated.  We wanted this capability so that the robot could accomplish episodic consolidation autonomously without prior semantic knowledge.

We converted the corr2 2-D correlation algorithm from MathWorks MATLAB into C# code for this work.  The algorithm compares the currently viewed image to a previously stored image on disk and returns a correlation value between 1 and 0 as output.

Additionally, the robot utilized an attention loop similar to the OODA loop [Ford, 2010] with the addition of an Evaluation segment.  As part of the OODA loop the robot Observed the world; Oriented to the world (based on inputs from the world (i.e. motion)); Decided on an action (retrieve a memory or pay attention to the movement episode) and finally executed an Action (remember the object or pay attention to a movement episode). The last Action was also Evaluated for reinforcement.  In other words, was SS-RICS successful in remembering the object?  So Evaluation was added to the end of the OODA loop creating the OODAE loop.

The robot was shown each object in sequence (object 1 then object 2, object 2 then object 1) by moving a monocular camera back and forth between the two objects.
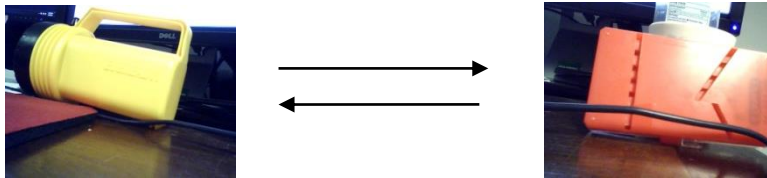
Fig. 2.  The two objects that were used for the task. The robot extracted color, proximity, and size features from each object and detected the motion of the camera.

### 5.1.1.  *Strategy 1 – Complete Memory Search*

We obviously knew that a complete memory search was not an efficient or practical strategy for memory retrieval, however, it served as a baseline for additional comparisons.

The robot collected images (metric data) and generated symbolic representations (declarative data) as we moved the camera back and forth between the two objects shown in Figure 2. The robot created 600 directories of images, each containing a JPEG image of approximately 100KBs, and generated the corresponding declarative information. The Complete Memory Search Strategy retrievals were executed while the motion event was occurring and were both declarative and metric. This strategy led to a linear increase in retrieval times with the addition of new memories (Figure 3).
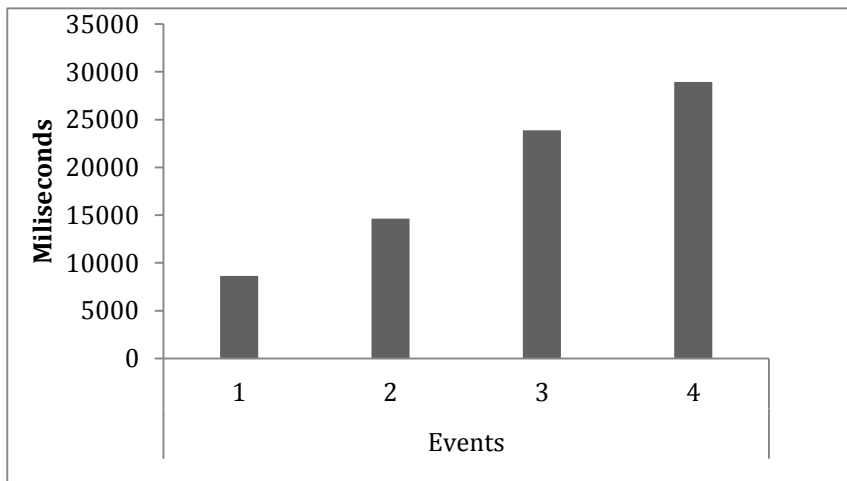
Fig. 3. Number of milliseconds for each retrieval following a movement event. We found a linear increase for each retrieval.

### 5.1.2. *Strategy 2 – Recent Memory Search*

Unlike the Compete Memory Search Strategy, which stored all the information (static and movement), the second strategy, the Recent Memory Search, *only stored the movement episodes* (from ESN 3 to ESN 4). The initial idea was that only movement episodes were important and subsequently only those episodes should be stored. Additionally, the Recent Memory Search Strategy used activation [Anderson and Lebiere, 1998] and searched from the most recent declarative memory backwards, essentially concentrating on the most recent memories for a possible match; creating the recency effect (where the most recent facts are more likely to be remembered).

The data gathered for the Recent Memory Search was done in a similar fashion as the Complete Memory Search, using the same image correlation retrieval algorithm for a metric match as well as a declarative memory match. One data set is presented in Figure

4.  The data pattern shown in Figure 4 was representative of the other runs that were conducted.

The data gathered for the test of the Recent Memory Search allowed for the OODAE loop to execute 400 times.  Note that the OODAE loop was executing while the camera was moving back and forth and while the camera was still.  Therefore the OODAE loop could execute several times during one event, producing several retrievals within one event (each episode might involve a retrieval).
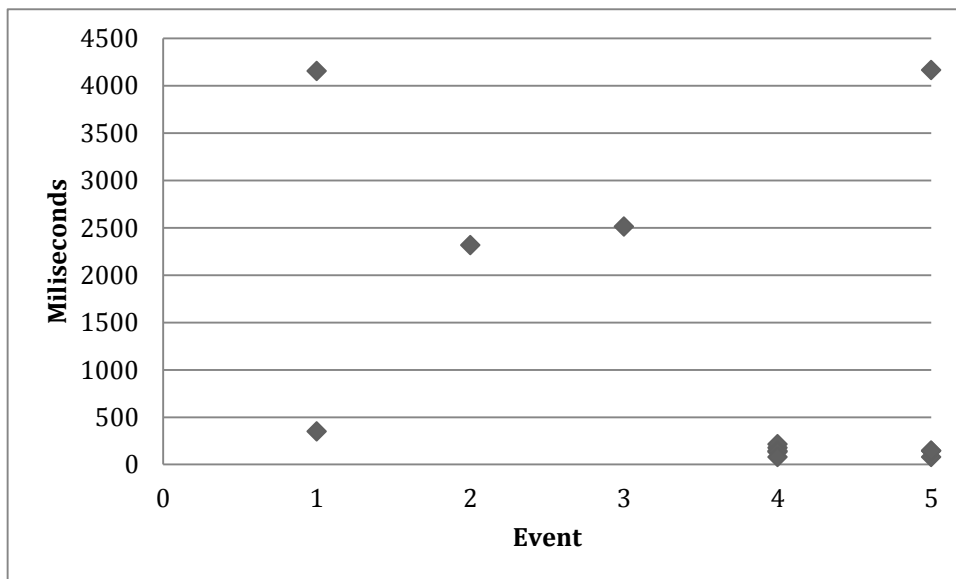


Fig. 4.  Number of milliseconds for metric matches using the correlation algorithm within five events.  We found a gradual improvement until an adequate match could not be found, at which point the retrieval times increased as the search space increased.

The times listed in Figure 4 for each event occurred between ESN 3 and ESN 4 (see Figure 1).  The data pattern shown in Figure 4 was typical of the type of data we found using the Recent Memory Search.  The retrieval times in Figure 4 are within one event (x axis), which involved multiple retrievals to find the best match.  Note that event 1 is not at time 00:00, but rather, it is the first movement event that occurred after some number of static events.

In Figure 4, the first retrieval took more than 4,000 milliseconds (because there was not a good match to begin with), but then quickly got reduced to less than 500 milliseconds after more data was gathered and the most recent matches could be used.  By event four, the data seemed to suggest that all the retrieval times for each event were getting faster

and the Recent Memory Search Strategy was a useful strategy because it was matching the most recent data. Event four showed that the time required by all the retrievals in the event was less than 500 milliseconds. However, by event five, we again saw an increase in times for a retrieval to around 4,000 milliseconds because SS-RICS could not find a suitable match even in the most recent data.

The data showed in Figure 4 were typical of Recent Memory Search across several runs. In general, the system was constantly matching to the most recent set of memories and the retrieval times for a match would decrease up to a point at which a good match could no longer be found, at which point more memory was searched to find an adequate match and the general pattern would repeat again.

### 5.1.3. *Strategy 3 – Episodic Indexing*

The Episodic Indexing Strategy inserted a pointer at the end of a static event (ESN 2) *which served as a cue for the subsequent movement events* (ESN 3). This modification was conducted as post-hoc processing on each episode and on both the declarative and metric data. The entire process involved three main components:

1) Pruning - Static memories were removed from memories to produce a reduced data set.
2) Abstraction – Declarative memories containing metric data were abstracted as symbolic categories to reduce storage and reduce retrieval times.
3) Cues – Cues were inserted into the episodes to create one complete event. Specifically at ESN 2, 3, 4 and 5 (see Figure 1).

As part of the data gathered for the Episodic Indexing Strategy each object was shown in the pattern outlined in Figure 2 – a camera being moved back and forth between two objects.

For the pruning process, the static declarative information and static metric information were removed during a post-hoc process. This occurred from ESN 1 up to ESN 2, but ESN 2 was kept. The critical factor for this strategy *was that the last static episode served as a cue for the next movement episode*. This meant that both movement episodes and static episodes had to be stored initially so that the static episodes could serve as a cue for the movement episodes. The process of refining both static episodes and movement episodes appeared to be best accomplished as a post-hoc process.

For the abstraction process, the declarative information contained at ESN 2, 3, 4, and 5 was processed post-hoc and stored as a complete new event. For example, as part of the abstraction process, new categorical declarative information was created to reduce the amount of metric information in the original declarative memory. Specifically, a green

object with an area of 2120 pixels and a red object with an area of 7500 pixels was reduced to "biggest object=RED" (see Table 1 and Table 2). These same principles of abstraction were applied to the other declarative memories representing the event.

For pruning, ESN 2, 3, 4 and 5 were kept and information between ESN 1 and ESN 2 was removed, with ESN 2 being kept to serve as a cue. Additionally, static information after ESN 5 was removed (see Figure 1). This created a reduced declarative memory set (see Table 2).

As previously mentioned episodes were created during each OODAE loop. Before the post-hoc processing, when an episode was created, it contained both metric and declarative information (this was the same for all the strategies). In contrast to the metric data, the declarative memory was stored in working memory and ordered by activation while the metric data for each image was stored on the disk. All of the episodic memories were then reduced to the example in Table 2 following post-hoc processing.

Table 1. An example of episodic declarative memories before the Episodic Indexing Strategy.

| Name | Type | GreenArea | YellowArea | RedArea | Pointer | Movement |
|---|---|---|---|---|---|---|
| 1 | Episode | 0 | 1223 | 7500 | 14 | False |
| 2 | Episode | 0 | 9000 | 0 | 15 | True |
| 3 | Episode | 0 | 0 | 0 | 16 | True |
| 4 | Episode | 0 | 0 | 0 | 17 | True |
| 5 | Episode | 0 | 9788 | 0 | 18 | True |
| 6 | Episode | 0 | 9989 | 1250 | 19 | False |

Table 2. An example of the declarative memories representing one complete event following the Episodic Indexing Strategy.

| Name | Type | Pointer | Event | Cycle | Color |
|---|---|---|---|---|---|
| 1 | Dream | 14 | 1 | 1 | Red |
| 2 | Dream | 15 | 1 | 2 | Yellow |
| 3 | Dream | 18 | 1 | 3 | Yellow |
| 4 | Dream | 19 | 1 | 4 | Yellow |

The information contained in Table 2 is an example of the declarative memories produced *after* the Episodic Indexing Strategy. It represents one event that was created as the result of the post-hoc processing of episodes.

The amount of abstraction across multiple runs was fairly consistent. For example, with 400 episodes captured during the training run after post-hoc processing we were left with

112 declarative memory elements and corresponding metric information (112 directories containing image data). Those 112 declarative memory elements represented a total of 28 events (4 declarative memory elements per event (see Table 2)). So we had a reduction of about a forth by getting rid of the static episodes using pruning. We used four declarative memories to represent one abstracted event. Each new memory contained a unique name, a type, a pointer, the event number, the cycle number, and the color of the biggest object in the scene. Note that the pointer was the essential information contained in the newly created declarative memory. It was a pointer to the metric data stored on the disk. In other words, in Table 2, the metric information for the declarative memory element named "4" of type "dream" was contained in directory "19." *This pointer allowed the system to quickly retrieve metric information once a symbolic declarative match had been made.*

In order to implement the Episodic Indexing Strategy we used the following steps:
1) While SS-RISCS was viewing an object, *and there was no movement,* the system would set a retrieval goal. The goal retrieved a declarative memory that matched the current scene *at the symbolic level.* Specifically, the camera was viewing the world, which in this case was a large red object, and SS-RICS was retrieving declarative memories which matched "large red object" (i.e. declarative memory 1 from Table 2). This type of working memory symbolic retrieval based on activation can be done extremely quickly and was further enhanced by the fact that it was done during moments of static activity.

2) Following the retrieval goal, assuming SS-RICS found *symbolic* match, the pointer in the declarative memory was then utilized. The pointer was used to compare the current scene to the *metric* data (using image correlation) stored on disk. The metric data was not in working memory (other than as a pointer) and these correlation comparisons were slower than *symbolic* working memory comparisons in the production system. However, this *metric* information was critical because it served to bolster the information at the abstracted *symbolic* level and confirmed the scene as a match. For example, this type of *metric* match allowed SS-RICS to discriminate between two difference types of red objects, since the *metric* information contained more detailed data than the *symbolic* data. To summarize, the first match was a working memory *symbolic* match and the second match was a more detailed LTM *metric* match.

3) Following a match of both the metric data and the symbolic data, SS-RICS then retrieved a declarative memory representing the *result* of the event. If it had been viewing a red object, the result would have been to expect a yellow object (see Table 2). At this point SS-RICS waited for the event to occur (movement) and was primed to anticipate a certain object (yellow object).

4) After the event completed, the LTM *metric* information was retrieved at the end of the event to confirm the result of the event.

Results showed that the metric match for the Episodic Indexing Strategy, as compared to Complete Memory Search and Recent Memory Search, was faster.  Below is a trace from SS-RICS of what the Episodic Indexing Strategy produced.

1) Waking up…
2) Attending to world
3) Remember Dream Episode Default
4) Cognitive Time is 1
5) Remember Dream Episode Default
6) Cognitive Time is 2
7) Subject Directory/43
8) Correlation .9115
9) Cognitive Time is 3
10) Subject Directory/43
11) Correlation .9115
12) Cognitive Time is 4
13) Motion State is True
14) Motion State is True
15) Motion State is True
16) Subject Directory/56
17) Correlation is .8162
18) Duration time is .208

As you can see from the trace, the SS-RICS took a few cycles of inactivity (Remember Dream Episode Default, lines 3 and 5), before it found a sufficient metric and symbolic match to the current scene - which occurred on line 8.  At that point it had acquired a pointer to the appropriate metric information (located in directory 43, Line 7) and it had retrieved the appropriate declarative memory information for the entire event (See Table 2).  Once the event started (movement), SS-RICS was essentially free to allocate resources (pay attention) the incoming information since it knew what to expect and was not doing retrievals.  This was different from the Recent Memory Search which was doing retrievals during movement episodes.  Also note that SS-RICS had an expectation as to how long the event would take place (about 5 attention cycles, from 14 to 19, see Table 2).  Once the movement of the camera stopped, the final image was correlated with the image in directory 56 and reported (correlation value .8162, Line 17).  Since the cue was already set this process was almost instantaneous (.208 seconds, Line 18). Additional data collected using this strategy produced similar results (see Figure 5).
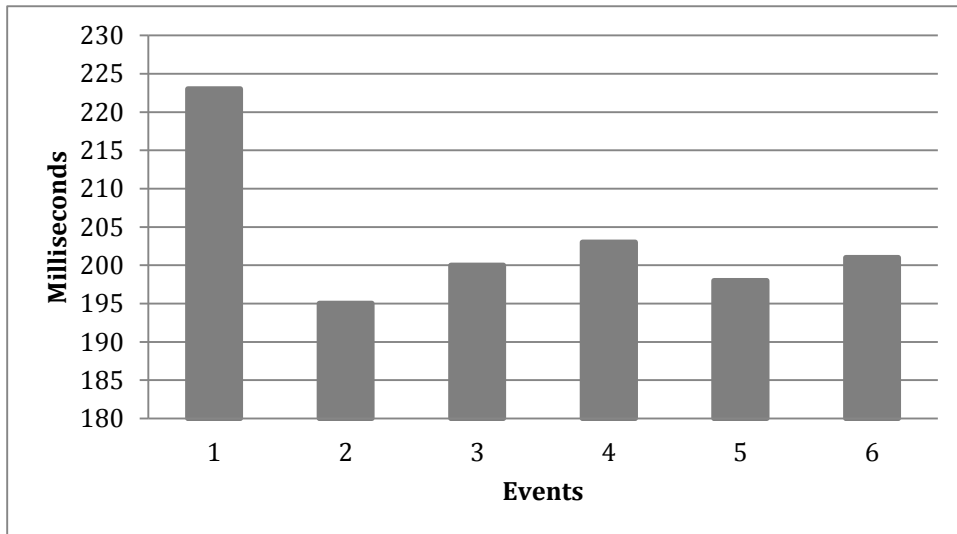
Fig. 5. Number of milliseconds for a final metric retrieval following the post-hoc processing of events. Each bar is the time associated with the last metric retrieval, using the image correlation algorithm, at the end of the event (ESN 5) to positively confirm an expected result.

The amount of time the post-hoc processing phase took after the data were collected was dependent upon the number of working memory items in the episodic trace. Additionally, the post-hoc process was not optimized to be as efficient as possible because it was written as a set of goals within SS-RICS instead of in C#. However, for this implementation of 400 memories, which accounted for approximately 30 seconds of real-time experience, the post-hoc process took approximately 3 minutes.

## 6. Future Work

This research has given us the opportunity to explore many possible directions for future work. Our most immediate goal is to integrate this process with our current novelty algorithm [Kelley & McGhee, 2013]. This would allow SS-RICS to transition from movement based events to novelty based events which are collections of movement events. Following this transition, novel events would become less and less novel as the robot gained more and more experience. Ideally, the robot would become more accustomed and familiar with its environment and find less information novel.

Our initial implementations of this memory retrieval task assumed that movement episodes were the most important information to store. And that the retrieval of the metric information, while movement was happening, would not be a problem computationally (Recent Memory Search Strategy). However, we found that both movement and static episodes needed to be stored (Figure 1) *so that the static episode could provide a cue to the movement episode*. Computationally, we found that attempting to retrieve a movement episode while an object was moving meant that some of the current movement information might be missed - and that the retrieval was not efficient.

For further research we need to incorporate an expectation model. The model would be based on the percentage of times the robot encountered some expected outcome. This can be handled as an expectation utility algorithm and will be further explored. If different outcomes are encountered, that are not part of the current expectations, the utility functions would be updated.

We would also like to relate our instances of events to conceptual primitives which would allow for generalization from conceptual templates [Mandler, 1992]. In this case the conceptual primitive would be something similar to, "one object replaced another object." Upon the completion of an event, the event would be matched to the conceptual primitive and the concept could potentially change. This could be done as a post-hoc process or in real-time.

Also, notice that each event is made up of four episodes (see Table 2) which represent the end of the static episode (ESN 2) the beginning of the movement episode (ESN 3), the end of the movement episode (ESN 4) and the beginning of the next static episode (ESN 5). It could be that these four bits of information are all that are needed to abstract an event into an efficient declarative memory element which represents the entire event, but this could be further researched.

Additionally, metric information should be stored in working memory so that the metric retrievals are much faster. In a sense, we are storing the metric information in working memory but only as a pointer to make it as small as possible, and to keep it essentially symbolic. In fact, it would be difficult to store the image based metric information in

working memory of a production system due to the amount of information, so using a pointer to the metric information seems to be a solution for now.

Finally, there is a question for us as to why this post-hoc processing has to be accomplished as an unconscious state instead of being accomplished in real time following an event. For example, why cannot the associative memory cues, pruning, and abstraction be set immediately following a novel event? For humans, there is some indication that learning can be accomplished during restful situations with no REM sleep. For example, goal based learning has been found to be improved and consolidated during REM sleep, but procedural motor learning has been shown to be improved and consolidated during restful periods. [Cohen et al., 2005]. Other studies have found that procedural learning is also improved during REM sleep [Gais et al., 2000]. From our simulations, it would appear that procedural learning has a clear advantage for post-hoc processing, since motor learning can be especially time critical and procedural tasks would presumably benefit from extremely fast retrievals of associated information. Computationally there is also some advantage to generating an unconscious state and shutting off the perceptual system to allow for memory consolidation to occur. This unconscious state allows the events to be replayed and consolidated without being confused with real time events. However, there could be some computational problems if the post-hoc process was interrupted by outside perceptual processes and required to start over again. This would lead to the computational equivalent of "grogginess." Assuming the brain is an analog system there might be some additional benefits during sleep to "rewind" events to get to the appropriate cue locations – thus requiring an effortful unconscious state. Indeed, it is clear that in humans, and in most other animals, sleep is an effortful, unconscious process and there is a general need for it to occur during periods when there is reduced external stimuli. For SS-RICS, we will continue to explore both an unconscious state for the robot during moments of inactivity as well as post-hoc processing in real-time during uneventful periods.

## 7. Conclusions

We implemented a post-hoc memory process that was more advantageous, in terms of efficient memory retrievals, than other strategies that were not implemented as a post-hoc process. This computational implementation appears to provide a computational justification for a post-hoc memory reorganization process seen in most mammals as sleep.

For robots it would appear that the post-processing of events is beneficial for the improvement of retrieval times by using cues, abstraction, and pruning. Further research needs to be done to determine the specifics of this process to optimize it for robotics applications.

# References

Alvarez, P., & Squire, L. R. [1994] *Memory consolidation and the medial temporal lobe: a simple network model*. Proceedings of the National Academy of Sciences, *91*(15), 7041-7045.

Anderson, J. R., & Lebiere, C. [1998] *The Atomic Components of Thought.*  Mahwah, N. J: Lawrence Erlbaum Associates.

Arkin, A., & Antrobus, J. S. (Eds.) [1978] *The mind in sleep*. Hillsdale, NJ: Erlbaum.

Aserinsky and Kleitman [1953] *Regularly occurring periods of eye motility, and concomitant phenomena during sleep*. Science, 1953 118 273-274.

Cavallero, C., & Cicogna, P. [1993] *Memory and dreaming*. In C. Cavallero & D. Foulkes (Eds.), Dreaming as cognition. Harvester Wheatsheaf.

Cohen, D. A., Pascual-Leone, A., Press, D. Z., & Robertson, E. M. [2005] *Off-line learning of motor skill memory: a double dissociation of goal and movement.* Proceedings of the National Academy of Sciences of the United States of America, 102(50), 18237-18241.

Crick, F., & Mitchison, G. [1983] *The function of dream sleep*. Nature, *304*(5922), 111-114.

De Koninck, J. M., & Koulack, D. [1975] *Dream content and adaptation to a stressful situation.* Journal of Abnormal Psychology, 84(3), 250.

Domhoff, G. William Moffitt, Kramer Alan. (Ed); Kramer, Milton (Ed); Hoffmann, Robert (Ed), [1993] *The functions of dreaming*. SUNY series in dream studies., (pp. 293-320). Albany, NY, US: State University of New York Press, x, 610 pp.

Ford, D. [2010] *A Vision So Noble: John Boyd, the OODA Loop, and America's War on Terror*. Daniel Ford.

Fosse, M.J., Fosse, R., Hobson, J.A., & Stickgold, R.J. [2003] *Dreaming and episodic memory: a functional dissociation?* J. Cogn. Neurosci. 15:1-9.

Gais, S., Plihal, W., Wagner, U., & Born, J. [2000] *Early sleep triggers memory for early visual discrimination skills*. Nature neuroscience, 3(12), 1335-1339.

Hobson, J. A. [1988]. *The dreaming brain*. Basic Books (AZ).

Hobson, J.A., Pace-Schott, E., & Stickgold, R. [2000] *Dreaming and the Brain: Towards a Cognitive Neuroscience of Conscious States*. Behavioral and Brain Sciences 23:793-842.

Kelley, T. D., & McGhee, S. [2013, May] *Combining metric episodes with semantic event concepts within the symbolic and sub-symbolic robotics intelligence control system (SS-RICS)*. In SPIE Defense, Security, and Sensing (pp. 87560L-87560L). International Society for Optics and Photonics.

Kelley, T. D. [2006] *Developing a psychologically inspired cognitive architecture for robotic control: the Symbolic and Sub-symbolic Robotic Intelligence Control System (SS-RICS).*

International Journal of Advanced Robotic Systems, Vol. 3, No. 3. p 219-222.

Nuxoll, A. M., & Laird, J. E. [2007, July]  E*xtending cognitive architecture with episodic memory.* In Proceedings of the National Conference on Artificial Intelligence (Vol. 22, No. 2, p. 1560). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Mandler, J. M. [1992] *How to build a baby: II. Conceptual primitives*. Psychological review, 99(4), 587.

McCarley, R. W. Hobson, J. A. [1977] *The neurobiological origins of psychoanalytic dream theory*. The American Journal of Psychiatry, Vol 134(11), Nov 1977, 1211-1221.

McCarley, R.W. & Hobson, J.A. [1979] *The form of dreams and the biology of sleep*, In: B.B. Wolman (Ed.) Handbook of dreams. Research, theories and applications [pp. 76-130). New York: Van Nostrand Reinhold Co.

Pavlides, C., & Winson, J. [1989] *Influences of hippocampal place cell firing in the awake state on the activity of these cells during subsequent sleep episodes*. Journal of Neuroscience 9:2907-2918.

Pearlman, C. A. [1979] *REM sleep and information processing: evidence from animal studies*. Neuroscience & Biobehavioral Reviews, 3(2), 57-68.

Sharma, M., Holmes, M. P., Santamaría, J. C., Irani, A., Isbell Jr, C. L., & Ram, A. [2007, January]. *Transfer Learning in Real-Time Strategy Games Using Hybrid CBR/RL.* In *IJCAI* (Vol. 7, pp. 1041-1046).

Siegel, J. M. [1999] *The evolution of REM sleep* (pp. 87-100). Boca Raton, FL: CRC Press.

Snyder, F. [1970] *The phenomenology of dreaming.* In L. Madow & L. Snow (Eds.), The psychodynamic implications of the physiological studies on dreams (pp. 124-151). Springfield, IL: Thomas.

Ram, A., & Santamaria, J. C. [1997]  *Continuous case-based reasoning*. Artificial Intelligence, 90(1), 25-77.

Squire, L. R., & Alvarez, P. [1995] *Retrograde amnesia and memory consolidation: a neurobiological perspective*. Current opinion in neurobiology, 5(2), 169-177.

Tulving, E. [1983]  Elements of episodic memory (p. 123). Oxford: Clarendon Press.

Wilson, M.A., McNaughton, B.L. [1994] *Reactivation of hippocampal ensemble memories during sleep*. Science 265:676-679.

Wolf, F. A. [1994] *The dreaming universe. Psychological* Perspectives, 30(1), 36-41.

Zhang, Q. [2009] *A computational account of dreaming: learning and memory consolidation.* Cognitive Systems Research, 10(2), 91-101.