Towards modeling false memory using virtual characters: a position paper

Michal Čermák and **Rudolf Kadlec** and **Cyril Brom**[1]

Abstract. This position paper presents our approach to development a long term episodic memory model featuring the false memory effect. We will explain motivation for the model, data structures used in the model and algorithms working over these structures. Finally we will present a prototype of an agent embodied in a 3D virtual world equipped with our model.

1 INTRODUCTION

Human memory is fallible: we do not remember everything we perceive, we forget, we may fail to retrieve information. A less traditional trait of memory fallibility - related to errors of commission rather than omission - is false memory [3]. Generally, false memory refers to "circumstances in which we are possessed of positive, definite memories - although the degree of definiteness may vary - of events that did not actually happen to us" [3, pp. 5]. Examples include remembering a *gist* of experience that may actually not correspond exactly to what has happened [3, 2, 20]; implanting distressing childhood memories, either accidentally at a psychotherapy [3, ch. 8][1, pp. 150] or in a laboratory experiment [18]; enhancing memories by or blending them with post-event information [17, ch. 4]; or fabrication of non-existing details of a criminal event by an eyewitness [3, ch. 6].

False memory is characteristic of normal rather then pathological remembering [3, 23]. Yet many people have only limited knowledge of false memory or may neglect its importance [28][1, pp. 151]. This can be particularly troublesome at courts and during psychotherapies. Therefore, the research on false memory phenomena (and increasing awareness of them) is important.

Computational approaches to memory modeling have become increasingly important in the past decade [21, 9]. In silico simulations enable a researcher to specify hypothetical mechanisms in precise detail, systematically explore the model and manipulate its parameters, and generate new predictions [26, 19, 25, 9]. It is known in computational cognitive sciences for some time that computational (neuro-)psychological episodic memory models, predominantly sub-symbolic ones, can produce some false memory-like phenomena (see [21], for a review of these models). However, to our knowledge, the issue of false memory has never been studied systematically in that field. At the same time, development of mathematical models of false memory by the community studying false memory directly is in its early stages [3, pp. 426-447].

In this position paper, we present our approach to computational modeling of false memory. We have been developing for about a year a generic episodic memory model featuring false memory characteristics, a model extending our previous episodic memory models [7, 4]. Of course there are some false memory characteristics that are out of our scope. The model is intended for acquisition, retention and retrieval of complex everyday events, such as cooking dinner (as opposed to events from laboratory tasks, e.g. presentations of lists of words). The memory representation is organized around memories of single objects (but not their features, e.g. not features of faces) and hierarchically nested events/episodes lasing from seconds to hours (e.g. knocking a door, opening the door, a visit) (see [5] for details). Our present aim is to develop architecture for false memory models rather than a single model fitting data from a particular experiment. Still, we believe that in future, when the model is stable enough, it can be used for the purpose of computational cognitive sciences. Additionally, because the underlying platform on which we test the model is a virtual character inhabiting a complex 3D virtual environment (see [6] for more on using VR for development of high-level cognitive models), the model can be also used in virtual reality applications. For instance, think of a serious game explaining to jurors limitations of eyewitness testimony with respect to false memory phenomena.

The rest of the paper is organised around the following points: 1) psychological underpinnings, 2) architecture of the model, 3) problems stemming from validating the model against human data, including human data acquisition.

2 GENERAL APPROACH

Our false memory model capitalizes on the fuzzy-trace theory [3, 12]. In a nutshell, this theory posits two parallel mechanisms that encode incoming information: *verbatim* and *gist*. While the former encodes the surface-form of the information in detail, the latter encodes the meaning in a coarse-grained way [12]. Of course, it may not be always clear what exactly a gist is. In our approach, the gist resembles the notion of a script [24], a knowledge structure about a stereotypical situation, including typical events that will occur and the most common deviations. The verbatim corresponds to a detailed log-based hierarchical representation of a particular flow of events as we used in [7]. The overall representation can be also linked to the event segmentation theory [29] and parts of the Conway's self-memory system, namely to episodic memories and general events [10].

Concerning recollection and familiarity, verbatim and gist mechanisms may operate in opposition to each other. For instance, when a memory trace for a particular detail is not strong enough, this detail may be replaced during recall by a different information "fabricated" based on the respective gist memory trace.

[1] Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic, email: mikajel@yahoo.com, rudolf.kadlec@gmail.com, brom@ksvi.mff.cuni.cz

3 ARCHITECTURE OF THE MODEL

Our cognitive architecture integrates a decision making module, a memory system, a perception module and an emotion generator. The architecture is detailed in [6]. The next section provides a brief introduction into already existing decision making module. Then the extended long term memory module of our architecture will be described.

3.1 Decision making module

Our agent is driven by the existing decision making module based on AND-OR trees [7]. Terminology used in our model largely comes out from the structures of this module. An AND-OR tree is a tree consisting of two types of nodes: AND nodes, also called actions in our model; and OR nodes, also called goals. The property of AND nodes (actions) is that in order to accomplish it, all its children must be performed. On the contrary the OR nodes (goals) can be completed by performing any of its direct children. AND nodes not containing any child can be performed directly and are also called *atomic actions*. The root of a tree is always an OR node and it is usually referred to as a *top-level goal*. All goals and actions can also have *affordance slots*, that are placeholders for objects, places, etc. that provide resources for a node's execution, i.e. they define the roles of missing objects. The term affordance [14] was coined by Gibson. The set of all AND-OR trees specifying an agent's behavior is denoted as D. The agent also has a short term memory module [7] that keeps a track of its current goals.

3.2 Elements of the episodic memory

Our memory structure for storing episodic memories is a pair $\langle C, S \rangle$ where C is a set of *chronobags* and S is a *schema bag*. A chronobag is a unit of memory representing a certain period of time, it stores the *episode structures* that model the verbatim of episodes experienced in that period. The term chronobag was first used in our paper [4]. A schema bag holds the gist of a typical episode of a certain type. The gist is represented by a statistics about co-occurrences of goals and their satisfying actions together with objects used by the agent. The following sections describe these components more closely. Figure 1 shows the structure of both C and S components of our episodic memory model.

3.2.1 Episode structures

An *episode structure* E is a tree-like structure consisting of *episodic nodes*, objects in affordance slot and time pointers. It incorporates all the actions performed while trying to satisfy one top-level goal. The root of the episode structures is always an episode node representing one top-level goal. Its children are actions that were performed in order to satisfy it.

Episodic nodes can represent either action, sub-goal or atomic action in the decision tree and the whole episode structure represents *action/goal traces* from the top-level goal to the atomic actions performed when trying to satisfy one top-level goal. If a node has more than one child, an order of execution of child nodes is stored in a *time pointer*.

When the agent performs an atomic action, the episode with the root node corresponding to the current top-level goal is located and episodic nodes reflecting the action/goal trace are added to the episode. If the agent performs the same action several times in a row, new nodes are stored in the memory only once. All objects used during execution of an action are linked with appropriate affordance slots. Instances of these object nodes are shared among all the episodes. Note that this structure is a core of our previous models [7, 6].

3.2.2 Chronobags

A chronobag is a structure for holding episodes experienced by the agent in a given time period. The memory can contain any number of chronobags, but will always contain at least one chronobag for episodes from the current day, this chronobag is called the *present chronobag*. Anytime a new episode is experienced by the agent, it will be stored in this chronobag. In all the chronobags, there is an ordered list of episode structures belonging to it. Moreover in the present chronobag, there is also a separate list for episodes that are not finished yet.

The action selection algorithm allows for temporary interruption of the top-level goal the agent is trying to accomplish. The agent can interrupt the current episode (i.e. performing actions satisfying the current top-level goal), experience another episode (accomplish another top-level goal) and return to the original episode (and original top-level goal) later. The present chronobag can therefore contain several opened episodes. Each time the top-level goal of an episode is successfully satisfied or the agent abandons its top-level goal, the particular episode is marked as finished and moved from opened episodes to finished episodes. This also happens to all opened episodes during the agent's sleep.

Chronobags are organized in a layered structure. In the lowest level there are chronobags for episodes from single days, in higher layers there are multiday chronobags that integrate episodes from lower level chronobags. The multiday chronobags hold episodes belonging to the period of time of its subordinate chronobags. Currently the model divides chronobags into four different layers, the most abstract layer incorporating episodes from 7-day period.

3.2.3 Schema bag

Specific part of our model is so called schema bag corresponding to the gist trace from the fuzzy-trace theory. It incorporates all the events the agent experienced during its existence and helps to determine how often the agent performed specific actions and how often it used specific objects. Any action, goal or atomic action from AND-OR trees experienced by the agent will have the associated node in the schema bag. These nodes are called *schema episode nodes*. Apart from these nodes, the schema bag also keeps separate nodes for each object the agent used during its lifetime. These are called *schema object nodes*. Schema bag also includes representatives of affordance slots and special nodes that connect object node with affordance slot it was used in. These special nodes are called *slot content nodes*.

Probably the most important component of the schema bag are *schema counters*. Schema counters keep track of how many times a set of schema nodes was executed/used by the agent. This set can contain schema episode nodes, slot content nodes, or both node types. The maximum set size is currently set to 3 due to combinatorial explosion problem. Schema bag not only provides information how often a specific node is executed or used, it also provides conditional probabilities $P(X|Y)$ where X and Y can be any set of schema nodes provided the combined size of sets X and Y is not larger than 3. Information deducible from the schema can

be for example: the agent visited a cinema 6 times so far; when commuting to work the agent used a bus 6 times out of 10.

Nodes in the schema bag and all the counters are updated on-line as the agent performs atomic actions.

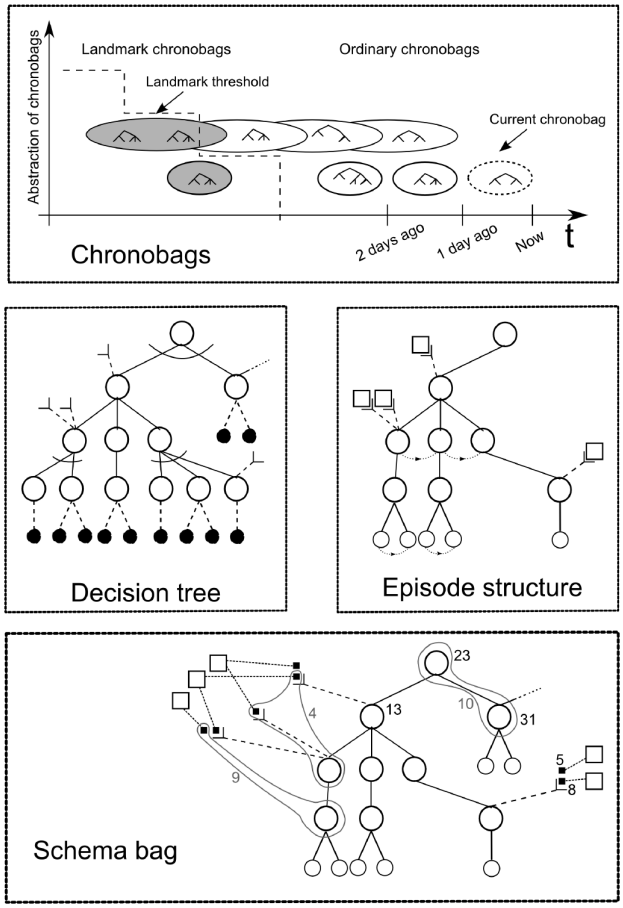

Figure 1. Different representations used by the memory model. The set of decision trees D represents procedural knowledge, it stores hierarchy of goals and actions satisfying those goals together with affordance slots (L-shaped figures) representing resources, and atomic actions (black circles) that can be performed directly in the agent's environment. The episode structure E represents actual experience. It is similar to one decision tree, but slots are already filled with object (squares). It also contains the sequence of episode nodes performed by the agent. Chronobags C hold the sets of episodes of similar age. As chronobags get older, details of episodes can be forgotten. When a chronobag gets old enough it becomes a landmark chronobag and it contains *fossilized* episodes that could not be forgotten. The boundary between ordinary chronobags and landmark chronobags is shown as a dashed line. Besides forgetting there is another process of creating more abstract chronobags for longer time periods. Details of lower level chronobags are merged into a higher level, more abstract chronobag spanning longer time period. The distinct chronobag on the right is the present chronobag used for storing current episodes. The last representation is schema bag S - schema bag is similar to the decision tree but it is extended with counts of how often each node was selected, how often each object was used in all affordance slots (black squares) and also it keeps a track of how many times different nodes and objects appeared in an episode together (there are three aggregate counts shown on the figure).

3.2.4 *Example of filling the memory*

For clearer conception of how new memories are added into the memory structures consider the following illustrative case. The agent starts with empty memory structures and he will try to fulfill the top-level goal *dinner* by performing following action *eating at restaurant*. To perform this action, he will have to complete the following subgoals: travel to a restaurant, order something to eat, eat it, and pay for the food.

When the agent starts following a new top-level goal, a new episode in the present chronobag will be created. The root of this episode will be the top-level goal *dinner*. This node will have one child node (*eat_at_restaurant*) and four grandchildren nodes (*travel*, *order*, *eat*, *pay*). Objects used will be also part of the episode: for example a *lobster* can be associated with the affordance slot *food* on the *eat* node. Each of these goals has to be completed by performing an action consisting of atomic actions executed by the agent in the virtual environment.

Apart from the episode structures, the schema bag is also being updated each time the agent performs an action. Imagine that the agent is sitting in the restaurant. The set of all schema nodes relevant to schema counter updating in this scene will be $S = S_{episode} \cup S_{slot\_content}$ where $S_{episode} = \{dinner, eat\_at\_restaurant, eat\}, S_{slot\_content} = \{lobster\_in\_food\_slot\}$. Then for each $X \subseteq S, |X| \leq 3$ the value of a schema counter will be increased by 1.

3.3 Processes maintaining the memory structure

One process behind maintenance of memory data structures deals with the acquisition of new information and it was explained in the previous section. Other processes described in this section are triggered during the agent's sleep and are more complicated. Some of these processes still have to be calibrated.

3.3.1 *Shifting of chronobags*

Shifting of chronobags simulates aging and generalization of episodic memories. There are two mechanisms working behind the chronobag shifting process each night:

1. Forgetting – as time passes chronobags are continuously being shifted back to the past. Age of chronobags is increased by one day every night. The present chronobag is moved to the set of past chronobags and new empty present chronobag is created. During every shift, some details of the episode can be gradually forgotten, as described later. This happens until the chronobag reaches age $t_l^{Landmark}$ when it becomes one of a *landmark* chronobags for the l-th level of chronobags. After this point no more details are forgotten from this specific chronobag. In literature this is referred to as a *flash bulb* memory, *flash bulb* memories are for example attacks from 9/11, birth of a child etc [8].
2. Episode merging – this process takes episodes from (non-landmark) consecutive chronobags, creates a chronobag representing union of time intervals of the chronobags being merged and copies all the contained episodes to it. This mechanism causes creation of several levels of abstraction of chronobags, with the daily chronobags being the least abstract chronobags. When the more abstract chronobag already contains a similar episode to the one being added, details of those episodes are merged, creating an "average" of the two. This is one of mechanisms for induction of false memories.

3.3.2 *Deriving an episode from the schema*

Existence of some nodes in the episode structure E can be deduced from other nodes in the episode with the use of schemas. If the conditional probability of existence of node n_1 given the existence of node n_2, that is $P(n_1|n_2)$ is close to 1, it means node n_1 does not have to be stored in episode E as long as node n_2 is not forgotten.

For example consider an agent that always goes to work by bus. Then the episode of *going to work by bus* happens every work day and it has the highest count among all ways of transport in the schema bag associated with going to work. It will be easily derivable from the schema (nodes *travel* and *work* will imply the existence of node *bus*). But when the agent oversleeps, it may use its car instead of the bus. Then this episode will not be derivable from the schema and its details should be remembered in a particular episode structure.

This mechanism helps to reduce the memory size and it would not cause any side effects if the derivability of nodes stayed constant during the existence of the agent and only derivable episodes would be forgotten. But in our model, even details that are not derivable can be forgotten, and in reality, the derivability of nodes can also change (because schemas are constantly updating). This process is another mechanism capable of inducing a false memory. Consider for example *going to work* episode mentioned above. If the node *car* is forgotten, the model will derive the node *bus* instead and the agent will not be able to distinguish this false memory from any other stored memory.

3.3.3 *Details of forgetting*

The forgetting of episode nodes is performed using the node's $score$. Each node is assigned a numeric score:

$$score = \sum_{a \epsilon Attributes} weight_a \cdot value_a \qquad (1)$$

based on the following $Attributes$ set: the user defined salience, the frequency of executing the node, the ability to derive its existence with the use of schema bag, the salience of objects attached to the node, the number of subnodes. Generally the score is higher for more interesting nodes: those more salient, less frequently executed and those that cannot be derived from the schema. Weights of all attributes will be fine-tuned during more complex testing of the model.

An important feature of the model is that the scores do not change in time. However, each chronobag has only limited capacity based on its age and the saliency of nodes in it. The capacity is currently calculated according to the formula:

$$capacity = MaxCapacity \cdot \frac{1}{a_l * t + 1} + b \qquad (2)$$

where t is the chronobag's age, a_l is a coefficient based on the chronobag's level of abstraction and b is a parameter used to increase capacity of chronobags with many salient nodes. The node scoring mechanism (Eq. 1) together with the limited capacity of chronobags (Eq. 2) should result in a believable forgetting process.

4 IMPLEMENTATION

The memory model is being developed as a standalone Java library independent of the agent's decision making system (DMS). The current implementation is divided into three separate projects:

- Bot – this library includes the DMS of the agent (in this case AND-OR trees) and it controls the agent's body through the Pogamut platform [13]. Pogamut is a tool for programming agents in virtual 3D environment.
- Memory ↔ Pogamut interface – a lightweight layer translating events originating in the agent's DMS into representation used in the episodic memory.
- Episodic memory model – a standalone library implementing the core of the model, that is: chronobags, schema bag, the chronobag shifting algorithm (see Section 3.3) and a GUI for exploring the content of the memory (see Figure 2). There is a clearly defined API used to insert information into the memory. AND-OR trees are the default formalism used by the model but any other DMS with hierarchical nature can be connected to the memory module too. The model works with the notion of more and less abstract actions (or goals), it does not matter whether those actions are implemented in the DMS as Hierarchical Finite State Machines, AND-OR trees etc.

This modular architecture makes it possible to connect our model to any other source of data without much effort in the future. For new environments, only the lightweight interface translating events to the format expected by the core memory model has to be implemented. The core episodic memory model can remain unchanged.

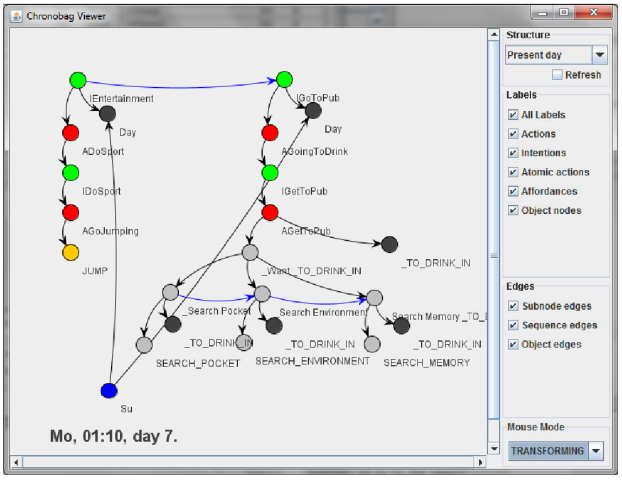

Figure 2. The GUI of the Episodic memory module showing a content of one chronobag. It shows two very simple episodes executed by the agent. The GUI is able to display contents of any chronobag, decision tree or schema structures. JUNG library [22] was used for visualization of the graph.

5 VALIDATION OF THE MODEL

Natural question is how the model will be validated and parameterized. Research on human memory provides only limited data about function of memory outside psychological laboratory. There are many experiments with memorizing lists of words, non-sense syllables and figures, but fewer results about working of memory in daily life on a scale of months, years or an entire life. For purposes of episodic memory modeling it would be best to have data with:

1. Input of the memory - e.g. all events, objects and other actors that the subject was exposed to.

2. Content of the memory - which of the inputs were remembered and the depth of detail that can be recalled.

Concerning Point 1, for longer time scales there is no such data yet, however this may change in a near future. Dawn of devices like Microsoft SenseCam [15] can generate tons of data about inputs of memory that we are currently lacking. Considering already existing published data, Wagenaar's six year diary study [27] and video based study of real life events [11] seems to be the closest matches to our requirements.

Concerning Point 2, the exact content of memory remains unknown, we can study it only through recollection and recognition experiments. Our aim is to fit data from this kind of experiments with human subjects.

In our methodology we want to create simulation on the scale of several months (and later a life time simulation) of our agent to obtain inputs of the memory. In a 3D simulated world we can log every subtle detail of the environment. After we obtain this log of information, we will use it as an input of our memory model and try to fit data dealing with false memories reported in [3, 16] and the data dealing with forgetting curves and retention intervals reported in [1, 11, 17].

We plan to perform several experiments:

1. The first is to prove that the model can recall episodes that did not happen but are compliant with the schemas. To do this we will perform simulation of three weeks with one set of plans the agent will be following and then one additional week with slightly modified plans. We expect to find reasonable parameters of our model, where a false memories will appear. We will try to fit the data reported in [16].
2. The second experiment should find parameters for a model that will approximate the retention curve of remembered memories. We will try to fit the real life data for several retention periods going from one day to several weeks, as reported in [1, 11, 17].
3. In the next experiment we will try to find out if our memory model is able to support a hypothesis that memory dating errors peak at multiples of seven days, as reported in [17].
4. We also consider creating a setting for the experiment where the agent's recollections of different events and items will be ordered. We want to parameterize the model so that less errors will be made in items recollected earlier, as reported in [17].

6 CONCLUSION

We have presented our computational model of long term episodic memory that aims to model false memory effect. The model capitalizes on our previous work [7, 4] and extends it with a notion of a schema bag and a chronobag shifting algorithm (Section 3.3). The chronobag shifting algorithm combining both gradual forgetting and episode merging was briefly described. We believe that these two mechanisms together with node derivability (Section 3.3.2) can result into emergence of false memory effects well known from psychological literature. However our model is currently a work in progress, the validation of the model against data from psychology will be the next step.

ACKNOWLEDGEMENTS

This work was partially supported by the student research grants GA UK 44910 and 21809, by the grant GACR 201/09/H057 and by the research project MSM0021620838 of the Ministry of Education of the Czech Republic.

REFERENCES

[1] A.D. Baddeley, M. Eysenck, and M.C. Anderson, *Memory*, Hove: Psychology Press, 2009.
[2] F.C. Barlett, 'Remembering: A study in experimental and social psychology', *University Press, Cambridge*, (1932).
[3] C.J. Brainerd and V.F. Reyna, *The science of false memory*, Oxford University Press, 2005.
[4] C. Brom, O. Burkert, and R. Kadlec, 'Timing in Episodic Memory for Virtual Characters', in *Proceedings of CIG 2010*, (2010).
[5] C. Brom and J. Lukavský, 'Towards virtual characters with a full episodic memory II: The episodic memory strikes back', in *Proc. Empathic Agents, AAMAS workshop*, pp. 1–9, (2009).
[6] C. Brom, J. Lukavský, and R. Kadlec, 'Episodic Memory for Human-like Agents and Human-like Agents for Episodic Memory', *International Journal of Machine Consciousness*, **2**(2), 227–244, (2010).
[7] C. Brom, K. Pešková, and J. Lukavský, 'What does your actor remember? towards characters with a full episodic memory', *Virtual Storytelling. Using Virtual Reality Technologies for Storytelling*, 89–101, (2007).
[8] R. Brown and J. Kulik, 'Flashbulb memories', *Cognition*, **5**(1), 73–99, (1977).
[9] N. Burgess, 'Computational models of the spatial and mnemonic functions of the hippocampus', in *The Hippocampus Book*, Oxford University Press, (2006).
[10] M.A. Conway, 'Memory and the self', *Journal of Memory and Language*, **53**(4), 594–628, (2005).
[11] O. Furman, N. Dorfman, U. Hasson, L. Davachi, and Y. Dudai, 'They saw a movie: long-term memory for an extended audiovisual narrative', *Learning & Memory*, **14**, 457–467, (2007).
[12] D.A. Gallo, *Associative illusions of memory: False memory research in DRM and related tasks*, Psychology Press, 2006.
[13] J. Gemrot, R. Kadlec, M. Bída, O. Burkert, R. Píbil, J. Havlíček, L. Zemčák, J. Šimlovič, R. Vansa, M. Štolba, et al., 'Pogamut 3 Can Assist Developers in Building AI (Not Only) for Their Videogame Agents', *Agents for Games and Simulations*, 1–15, (2009).
[14] J.J. Gibson, *The ecological approach to visual perception*, Lawrence Erlbaum, 1986.
[15] S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, and K. Wood, 'SenseCam: A retrospective memory aid', *UbiComp 2006: Ubiquitous Computing*, 177–193, (2006).
[16] J.M. Lampinen, S.M. Copeland, and J.S. Neuschatz, 'Recollections of things schematic: Room schemas revisited', *Journal of Experimental Psychology: Learning, Memory and Cognition*, **27**, 1211–1222, (2001).
[17] E. Loftus, *Eyewitness Testimony*, Harvard University Press, 1979.
[18] E. Loftus, 'Creating false memories', *Scientific American*, **277**, 70–75, (1997).
[19] S.C. Marsella and J. Gratch, 'EMA: A process model of appraisal dynamics', *Cognitive Systems Research*, **10**(1), 70–90, (2009).
[20] U. Neisser, 'John Dean's memory: A case study', *Cognition*, **9**(1), 1–22, (1981).
[21] K.A. Norman, G.J. Detre, and S.M. Polyn, 'Computational models of episodic memory', *The Cambridge handbook of computational cognitive modeling*, 189–225, (2008).
[22] J. O'Madadhain, D. Fisher, and T. Nelson. Java Universal Network/Graph Framework. `http://jung.sourceforge.net`, 2011.
[23] D.L. Schacter, *The seven sins of memory: How the mind forgets and remembers*, Mariner Books, 2002.
[24] R.C. Schank and R.P. Abelson, *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*, volume 2, Lawrence Erlbaum Associates Hillsdale, NJ, 1977.
[25] R. Sun, *The Cambridge handbook of computational psychology*, Cambridge University Press, 2008.
[26] T. Tyrrell, *Computational mechanisms for action selection*, University of Edinburgh, 1993.
[27] W.A. Wagenaar, 'My memory: A study of autobiographical memory over six years', *Cognitive psychology*, **18**(2), 225–252, (1986).
[28] R.A. Wise and M.A Safer, 'A Survey of Judges' Knowledge and Beliefs About Eyewitness Testimony', *Court review*, 6–16, (2003).
[29] J.M. Zacks and K.M. Swallow, 'Event segmentation', *Current Directions in Psychological Science*, **16**(2), (2007).