Spatial Working Memory in the LIDA Cognitive Architecture

Tamas Madl (tamas.madl@gmail.com)
School of Computer Science, University of Manchester, Manchester M13 9PL, UK
Austrian Research Institute for Artificial Intelligence, Vienna A-1010, Austria

Stan Franklin (franklin@memphis.edu)
Institute for Intelligent Systems, University of Memphis, Memphis TN 38152, USA

Ke Chen (chen@cs.manchester.ac.uk)
School of Computer Science, University of Manchester, Manchester M13 9PL, UK

Robert Trappl (robert.trappl@ofai.at)
Austrian Research Institute for Artificial Intelligence, Vienna A-1010, Austria

Abstract

Human spatial representations are known to be remarkably robust and efficient, and to be structured hierarchically. In this paper, we describe a biologically inspired computational model of spatial working memory attempting to account for these properties, based on the LIDA cognitive architecture. We also present preliminary results regarding a virtual reality experiment, which the model is able to account for, and the quantitative properties of the representation.

Keywords: Spatial memory, LIDA, cognitive architecture, grid cell code

Introduction

In this paper, we describe a partially implemented spatial memory model based on the LIDA (Learning Intelligent Distribution Agent) cognitive architecture, focusing on allocentric representations and short-term working memory, and attempt to verify the assumptions underlying this model by accounting for error patterns made by human subjects in spatial memory tasks in a 3D virtual reality.

Since Baddeley and Hitch's influential working memory model (A. D. Baddeley, Hitch, et al., 1974; A. Baddeley, 2003), a number of computational cognitive models of working memory have been developed, many of which include spatial representations. Most of such models focus on processing spatial stimuli in two dimensions, such as arrays of objects on a screen (Winkelholz & Schlick, 2007; Fischer, 2001). To our knowledge, however, short-term memory in three dimensions has received little attention from the cognitive architectures community.

LIDA (Franklin, Strain, Snaider, McCall, & Faghihi, 2012; Franklin & Patterson Jr, 2006) is a conceptual and partially implemented computational cognitive architecture based on Baars' Global Workspace Theory (GWT) (Baars, 2002) of functional consciousness in brains. Three properties of LIDA make it well-suited to modelling spatial memory: specific and detailed subdivisions of memory systems, cognitively plausible memory representations based on Sparse Distributed Memory (SDM[1]) (Baars & Franklin, 2003; Snaider & Franklin, 2012), and a functional consciousness mechanism central to LIDA's cognitive cycles, making use of a short-term memory capacity which enables access between cognitive functions that are otherwise separate, and which includes spatial representations.

Working Memory in LIDA

LIDA's cognitive cycles, corresponding to the action-perception cycles in neuroscience (Freeman, 2002; Fuster, 2002), consist of three phases. The *understanding* phase includes sensing the environment, detecting features, recognizing objects and categories, and building internal representations. The *attending* phase is responsible for deciding what portion of this representation should be attended to and broadcast to the rest of the system, making it the current contents of consciousness. This portion allows the agent to choose an appropriate action to execute in the *action* phase. These phases, their timing, and empirical support have been described elsewhere (Madl, Baars, & Franklin, 2011; Franklin, Madl, D'Mello, & Snaider, in review).

During the understanding phase, percepts are recognized based on LIDA's perceptual knowledge base, the Perceptual Associative Memory (PAM), which is a semantic net containing nodes with activation connected by links. Recognized objects, categories, etc. are stored in LIDA's preconscious *Working Memory*, and are represented by structures of PAM nodes and links between them (Franklin & Patterson Jr, 2006). These PAM node structures - parts of the PAM semantic net - are hierarchical, modal representations similar to Barsalou's perceptual symbols (Barsalou, 1999).

Since in a complex environment, the internal representation of the current situation could be too complex to process by the action selection system at once, LIDA agents need to select the most important or urgent portions of the representations. Following GWT, this selection is implemented as a 'competition for consciousness'. The most salient (important, urgent) PAM node structures win this competition, are broadcast globally - bringing their contents to 'consciousness'[2] -, and allows agents to select the most appropriate ac-

[1] SDM is a mathematical model of human long-term memory based on large, sparse vectors, proposed by (Kanerva, 1988)

[2] Percepts which have been broadcast to other modules of LIDA,

tion. The number of representations (PAM node structures) that the broadcast can contain is limited to C, a model parameter - such as e.g. $C = 4 \pm 1$ in accordance with human working memory capacity limits (Cowan, 2010) - which plays a role in the recall of spatial information.

Spatial Working Memory

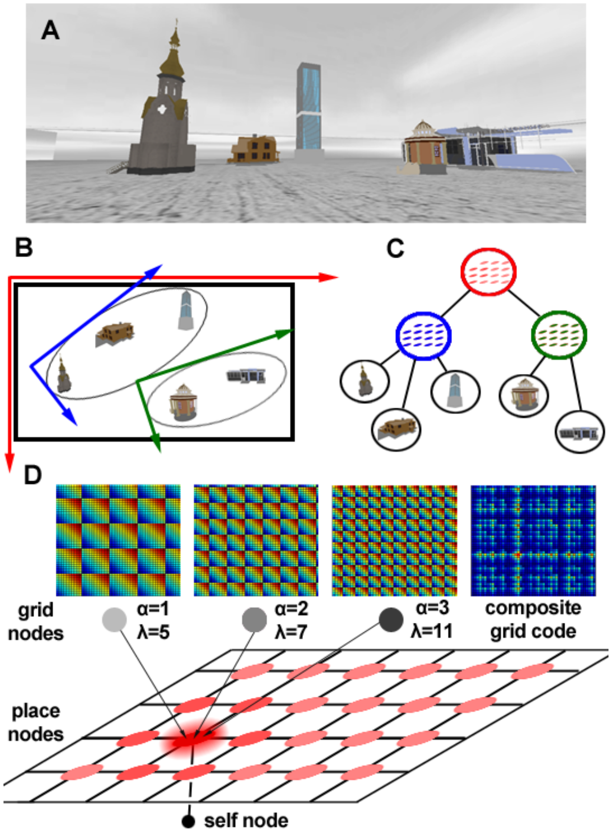

Figure 1: Cognitive map representations. A: screenshot of the virtual reality environment. B: Top view of the environment. Thin ellipses: sub-maps. Red, blue, green axes: reference frames for the entire environment (aligned to boundaries), and for the two sub-maps (aligned to intrinsic structure). C: The hierarchical cognitive map structure. Red, blue, green circles: map and submaps; black circles: objects. Links convey containment and rotation information; place nodes contain position information. D: Location representation by grid and place nodes. Top row: three grid nodes, their phases at each point (hot colors - high values), and the value of the composite code (right). Grid nodes convey activation to all place nodes. The 'self' node (bottom) monitors all place nodes and attaches to the winning node with the highest activation.

Neural basis. Spatial memory in LIDA is inspired by the computational properties of the hippocampal-entorhinal complex (HEC), the neural structure responsible for spatial memory in brains (Moser, Kropff, & Moser, 2008). Two types of neurons are important for this purpose: place cells, hippocampal neurons firing in specific spatial locations and thus suggested to encode a 'cognitive map', and grid cells, entorhinal neurons firing at multiple regularly positioned locations.

The HEC is uniquely suited for representing space. Grid cells employ a surprisingly precise and noise-robust, self-correcting, modular code, which can produce exponentially small error at asymptotically finite information rates and is thus a 'good code' in the sense of Shannon's theorem (Sreenivasan & Fiete, 2011). The grid cell code resembles a residue number system (RNS), which allows very fast, parallel, carry-free computation and recall (since each digit of the representation can be processed independently from the others) and degrades gracefully (Yang & Hanzo, 2001). Our high-level model of this grid code attempts to exploit these properties, which have the further advantage of fitting neatly into LIDA's SDM-based memory systems - in which error tolerance and graceful degradation are important, because of possible interference between memories, and an RNS with small parts can reduce storage requirements (Snaider & Franklin, 2012), and, finally, the carry-free code is easily implementable on parallel hardware.

The readout network for making the grid cell representation explicit is suggested to be implemented by hippocampal place cells, neurons with spatially localized firing. Most place cells provide unambiguous location estimates; however, they encode locations via Classical Population Codes, and thus their capacity grows only linearly with the number of neurons, instead of exponentially as is the case with grid cell coding (Fiete, Burak, & Brookings, 2008).

Finally, the hippocampal-entorhinal map contains representations on different scales, and is fragmented into independent representations or reference frames (Derdikman & Moser, 2010; Sato & Yamaguchi, 2009).

Location Representation. For representing a LIDA agent's spatial location, we use N 'grid nodes', which are types of PAM nodes employing modular representation. Each grid node α has a period λ_α (the same for all dimensions) and encodes a spatial vector $\boldsymbol{x}$ as a spatial phase with respect to this period: $\chi_d = x_d \, mod \, \lambda_\alpha$, where x_d is the position in dimension d - based on the formalization by (Fiete et al., 2008). Each grid node represents a grid lain over the environment (see Figure 1D for a plot of the spatial phases χ_d for three grid node periods λ_α). However, one grid node cannot unambiguously represent a location, having multiple maxima.

In order to represent unambiguous locations, N grid nodes are used to represent each location as a composite grid code: $L(\boldsymbol{x}) = \{\chi_1, ..., \chi_N\}$. At initialization, each grid node is randomly assigned a period λ_α and a phase $\boldsymbol{\chi}_{\alpha,0}$. The phase is relative to the reference frame provided by the boundaries of the immediate environment (similarly to biological grid cells). When the agent moves by a vector $\boldsymbol{v}_t$ in any direction at timestep t, the phase of each grid node is updated: $\boldsymbol{\chi}_{\alpha,t} = (\boldsymbol{\chi}_{\alpha,t-1} + \boldsymbol{v}_t) \, mod \, \lambda_\alpha$. This mechanism resembles rep-

and can be acted upon, are called functionally conscious, as in GWT resenting a large number by separately storing each digit in a register, as in a fixed-base code (FBC) such as the decimal system. In this case the location representation $L(\boldsymbol{x})$ would be a set of registers together representing a decimal number. However, there are some key advantages of this modular RNS coding (see previous section and Results).

Grid nodes alone could not represent a 'cognitive map' - although the composite code using multiple nodes can represent unique locations, it is aperiodic, non-Euclidian, and cannot easily be used for metric comparisons. For representing a map, the model uses a number of 'place nodes', readout nodes arranged over the environment, with a density/resolution depending on the size of the map. Multiple such maps can be built in a hierarchical fashion. To be able to decode the modular grid code, all grid nodes are connected to each place node via PAM links. Each place node i is assigned a constant phase $\boldsymbol{\chi}_{pn,i}$ calculated from the position it is representing in space, and an activation of 0. Upon each update, grid nodes asynchronously send an activation depending on the phase difference between the grid node and the place node: $\Delta a = (1 - |\chi_{\alpha,t} - \chi_{pn,t}|/\lambda_{\alpha})N^{-1}$. The LIDA-agent's 'Self'-node (Ramamurthy, Franklin, & Agrawal, 2012) implements a winner-take-all mechanism, and will be linked to the place node with the highest activation[3]. This is an optimized mechanism for the maximum likelihood position estimation in (Sreenivasan & Fiete, 2011), sacrificing some biological plausibility[4] for computational efficiency.

Distances within place nodes can be estimated either by propagating activation through the place node network over links (see Results), or by structure building codelets creating explicit spatial links between object nodes if required (McCall, Franklin, Friedlander, & DMello, 2010).

Environment Representation. To represent an environment, PAM node structures representing objects (obstacles, landmarks, goals, etc.) in LIDA's working memory need associations with spatial locations. This is implemented by high-level feature detectors connecting such representations to the appropriate place node using a PAM link, similarly to the connection with the 'Self' node to a place node. These feature detectors find the right place node to connect to based on the objects perceived distance [5].

As described in the previous section, LIDA's attention and functional consciousness mechanism lets multiple representations compete against each other in order to select the most important information to act upon. The broadcasts contain one to five (see capacity limit below) node structures, which in turn are hierarchical, tree-like representations containing objects and associated spatial information, i.e. arrays of place nodes. Such a node structure containing spatial information is called a 'cognitive map' structure in LIDA, and is allocentric (relative to aspects of the environment, instead of the agent).

Multiple cognitive map structures can be used within the same environment in a hierarchical fashion (there can be maps and sub-maps on different scales, and containment relations between them). There is some evidence that the human cognitive map is hierarchical, organized according to clustered objects or landmarks (Hirtle & Jonides, 1985; Sato & Yamaguchi, 2009). Hierarchical representations are a crucial part of the model, as is the limited information carried by the conscious broadcast. We strengthen these assumptions using human behaviour data in the Results section.

Containment relations between map structures can be seen as reference frames. Sub-maps have to be anchored to the environment structure of the parent map, such as to boundaries. Relative position information is given by which place node the link between the two maps is connected to, whereas relative rotation is stored within the link - see Figure 1. If information from multiple sub-maps is needed for problem solving (e.g. shortcut or multi-goal route planning), the sub-maps are rendered onto the parent map in order to be able to use a common reference frame.

Methods

Participants. Sixty subjects were recruited on the online crowd-sourcing platform Amazon Mechanical Turk, which has been argued before to be well-suited as a source of subjects for experimental and behavioural research (Paolacci, Chandler, & Ipeirotis, 2010; Mason & Suri, 2012). Subjects indicated their consent on the form, and received a small monetary reward for participation.

Procedure, Materials, and Design. Participants were free to perform the experiment on their own computer (the only requirement was a WebGL-enabled browser). After reading instructions, they were asked to complete 15 trials, lasting about an hour in total. At the start of each trial, a 3D virtual environment[6] was displayed within the browser which subjects could freely explore, consisting of a large square horizontal floor patterned like concrete and constrained by four walls, and a number of buildings within the walls. Each trial consisted of four phases: 1) free exploration, 2) questions about the distances between buildings, 3) a map recall task in which subjects had to move icons of the buildings into their remembered positions, creating a schematic 2D map, and 4) the travelling salesman problem (TSP), in which they were asked to mark all buildings using the shortest path visiting all buildings and returning to the first one. The 15 trials consisted of five blocks of three, with the number of displayed buildings starting at four and increasing after each block up to a total of eight displayed buildings. Within a block, there were three trials, a *random trial* in which the buildings were placed randomly (but with a minimal distance ensuring that no random clusters emerged), a *visually clustered trial* in which build-

[3] The 'Self' node receives links from all place nodes, but only projects back to the winning place node. This winning link is updated whenever a different node exceeds the activation of the winning node. This is more efficient than lateral inhibition

[4] Apart from the simplified connectivity, we have also removed the circular normal and Gaussian functions, and constrained the RNS variables to integers in order to fit the integer SDM

[5] For now, feature detectors receive the true distances explicitly.

[6] Using the CopperLicht engine, http://ambiera.com/copperlicht

ings were also placed randomly, but visually similar buildings were grouped (all buildings of a type, e.g. skyscrapers or churches, were adjacent), and a *distance clustered trial* in which some buildings were placed closely next to each other to build a random number of clusters.

Participants estimated building distances, and their recalled building positions on the map recall task, were recorded after each round. Since these building positions were drawn on a separate 2D screen and used arbitrary pixel distance units instead of 3D distance, they were linearly transformed (translated, scaled and rotated) to best conform to the real positions using procrustes analysis. Fourteen participants were excluded from the analysis, because their estimated distances had negligible correlations to the real distances (the correlation threshold was set at $r >= 0.2$). All reported subject errors are squared distances (or the sum of squared distances for multiple buildings) in the 3D engine's coordinate system. Model parameters were fitted using coordinate descent.

Results and Discussion

Robustness and performance

The outlined implementation of the grid code inspired RNS can represent an exponentially large range of distances, similarly to the range in a FBC. The maximum representable distance is the least common multiple of all phases λ_{alpha}, or in the best case $D_{max} = (\prod_{\alpha=1}^{N} \lambda_{alpha}) - 1$ (Fiete et al., 2008). However, introducing redundancy by only using a subinterval within this representable range, the modular code becomes highly robust to noise, because the representation of each location $\boldsymbol{x}$ within the subinterval maps to highly different location representations L. A representation L' in which each phase χ_{λ} is corrupted by independent noise - e.g. interference effects in SDM - can be fully recovered, as long as the difference is smaller than the largest correctable error - see Figure 2A for numerical results of error robustness in our model, and (Sreenivasan & Fiete, 2011) for an analytical treatment of error correction capability of the grid cell code.

The second advantage is performance - RNS codes can perform addition, subtraction, and multiplication in linear time. In contrast, FBCs need polynomial time for multiplication. For addition - the most frequently required operation on spatial information (e.g. for path integration or multi-goal path planning) -, both RNS and FBC require linear time, but the latter require carry operations, which makes the computation slower. Figure 2B compares the number of operations required for adding two randomly generated numbers in the range of 0 - 10.000m in the two representation systems.

A third advantage of the described RNS representation is the ease of parallelization - since each phase can be independently processed. LIDA's SDM has been implemented both for parallel and for GPU computation.

Distance errors

Judgement (and recall) of spatial distance information is imperfect. To be able to interpret our data, we will assume the noise or error to depend on the absolute distance. We will assume a linear relationship; although this is just an approximation, it provides a reasonably good fit for most distance judgement data in virtual environments (Lampton, McDonald, Singer, & Bliss, 1995; Waller, 1999).

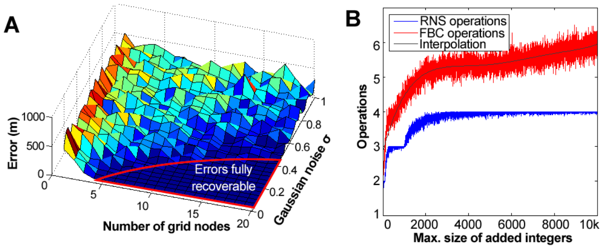

Figure 2: Robustness and performance of RNS-like representations. A: errors of location representations in the described grid node representations for different grid node numbers and levels of multiplicative Gaussian noise (in a square environment of $1km^2$). B: number of required operations to add two integers of various sizes in an RNS and FBC representation

Except for the 14 excluded subjects, participants were able to judge distances in the virtual environment: the average correlation between real and estimated distances was $r_{dist} = 0.82$ ($\sigma = 0.10$). The distance errors were correlated with the real distance, with an average $r_{err} = 0.56$ ($\sigma = 0.18$), and a linear model $error = c_1 * dist + c_2$ could explain $R^2 = 0.35$ ($\sigma = 0.19$) proportion of the variance on average. Note that there was a very high amount of random variance due to the guesses entered when subjects did not remember the distance - exponential models did not perform significantly better than the linear model ($error = c_1 * dist^{c_2} + c_3$: average $R^2 = 0.39$).

Position errors

The hierarchical cognitive map model predicts that errors should depend linearly on map size. They should be smaller on small maps, since the positions are stored relative to the map reference frame instead of a global reference frame (if the latter was the case, the cluster sizes would not matter, only the distances from the walls).

Figure 3 shows the dependence of the squared position errors on cluster sizes (approximated as the area of the smallest rectangle enclosing all buildings within the cluster), which is approximately linear. The position errors are strongly correlated with cluster size ($r = 0.82$), and a linear model explains $R^2 = 0.67$ proportion of the variance. Together with the distance errors depending linearly on the distance, this is an indication of building positions being stored relative to intrinsic cluster structure, not relative to the global reference frame.

The green line in Figure 3 shows the moving average of the errors of a simulated agent, representing environments in the way described in the Introduction. Three parameters were used for fitting this dataset, two defining an additive Gaussian error $\mu_{\varepsilon}, \sigma_{\varepsilon}$ assigned to the place nodes and designed to model readout error, and a random error added at the moment of conscious recall in the Global Workspace, in the range defined by recall error ξ. Since Gaussian noise is added to each readout place node, the total error depends on the number of place nodes involved in the representation, which depends on the cluster (map) size. The blue line shows a moving average of subject errors (with the smoothing parameter $S = 100$ for both). Since the errors are random, the exact correspondence between the two changes. The linear interpolation of the model errors explains $R^2 = 0.38$. The correspondence between the data and the model substantiates the hierarchical storage of positions instead of a global reference frame.

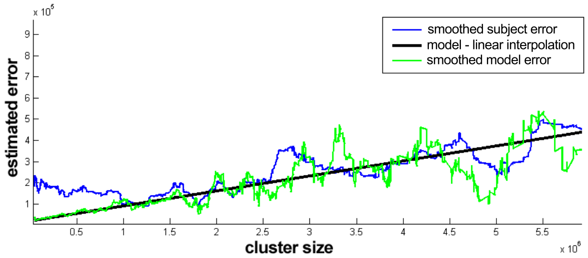

Figure 3: The average squared building position errors, depending on the area of the cluster the building is part of. Blue line: moving average of position error data from all subjects. Green line: moving average of model position errors (with the same smoothing parameter $S = 100$ for both). Black line: linear interpolation of model errors

Working memory capacity

The model predicts that map errors should increase linearly with each new building (after each block), since the addition of a new position to be recalled adds a new source of error, and the total error is a linear function of the individual building errors. However, the working memory capacity limit of 4 ± 1 (Cowan, 2010) implies that there should be a non-linear increase in errors when the number of entities on the map exceeds four. In this case we would expect a guessing behaviour which ensures guesses to be farther than a minimum distance D from other buildings. Thus, with each guess, the space available for further guesses is constrained by a constant factor, leading to an exponential increase in error.

Figure 4 shows the map errors depending on the number of buildings. Additionally, the following model formalizing the constrained guessing idea is fitted to this data using a least-squared errors approach (x being the number of buildings):

$$M(x) = \begin{cases} k * x + d & \text{if } x <= 5 \\ a * x^b + c & \text{if } x > 5 \end{cases} \tag{1}$$

To aid visualization, the linear part is plotted in red, and the exponential part in green. The fitted model explains $R^2 = 0.982$ proportion of the variance on average ($\sigma = 0.017$). This suggests that the spatial working memory capacity limit used in the model is realistic (and that constrained guessing might explain the behaviour beyond the limit).

In LIDA, conscious broadcasts are limited to C node structures (a model parameter). The C node structures with the highest activation are broadcast consciously. Node structures beyond C get omitted; no spatial information is available to the model beyond C buildings, forcing it to guess (the constrained guessing strategy is not part of the LIDA framework implementation). This is illustrated by the horizontal black lines in Figure 4, which show the error rates under random guessing (approximated numerically). Performance is statistically indistinguishable from random guessing when the number of buildings exceed 5, in accordance with previously suggested working memory capacity limits (Cowan, 2010).

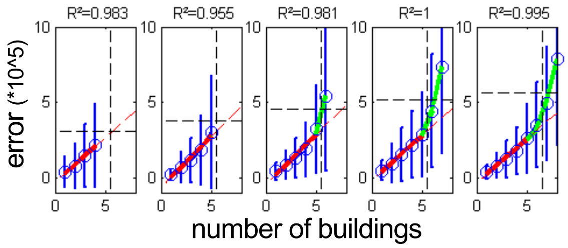

Figure 4: Map errors depending on the number of buildings. The blue errorbar plot and blue circles show the map error (sum of squared position errors of buildings), the other colors show the fitted nonlinear function (eq. (1)), consisting of a linear component (red; increasing because with more buildings there are more distances to misjudge) and an exponential component (green; increasing superlinearly because above 4 buildings, the spatial information exceeds the capacity limit of spatial working memory). Horizontal dashed line: random guessing performance. Vertical dashed line: limit of the number of recallable buildings with better than random accuracy

Travelling Salesman Problem errors

The described model can solve the TSP with its simple navigation heuristic, the nearest neighbor strategy, by spreading activation from the self node and from the place nodes associated with buildings, across the network of place nodes (Figure 1). The links connecting the place nodes propagate a fraction of the activation determined by the link weight. Thus, place nodes being on the shortest path between two buildings, or between a building and the self node, receive the highest amount of activation, can be selected by an attention codelet, and be included in the conscious broadcast for action selection (Franklin et al., 2012). Although this process would yield a better performance if performed hierarchically (starting with a rough solution and then refining it), the simple, flat nearest neighbour strategy explained the subjects data well, with $R^2 = 0.92$ ($\sigma = 0.05$) across trials (the Percentage-Above-Optimal metric was used to compare the subjects - and the models - performance against the optimal solution). The success of this simple heuristic might be due to the TSP being more difficult in navigation space than in the usual visual setting (Wiener, Ehbauer, & Mallot, 2009). There was no significant performance difference between the trial types.

Conclusion

We have presented a partially implemented, biologically inspired model of allocentric spatial working memory, argued for its robustness, and strengthened its major assumptions (robust modular coding, hierarchical representation using intrinsic reference frames, and limited capacity) using a simulation and subject data from a virtual reality experiment.

Acknowledgments

This research was supported by EPSRC (Engineering and Physical Sciences Research Council) grant EP/I028099/1, and FWF (Austrian Science Fund) grant P25380-N23.

References

Baars, B. J. (2002). The conscious access hypothesis: origins and recent evidence. *Trends in cognitive sciences*, *6*(1), 47–52.

Baars, B. J., & Franklin, S. (2003). How conscious experience and working memory interact. *Trends in Cognitive Sciences*, *7*(4), 166–172.

Baddeley, A. (2003). Working memory: Looking back and looking forward. *Nature Reviews Neuroscience*, *4*(10), 829–839.

Baddeley, A. D., Hitch, G. J., et al. (1974). Working memory. *The psychology of learning and motivation*, *8*, 47–89.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and brain sciences*, *22*(04), 577–660.

Cowan, N. (2010). The magical mystery four how is working memory capacity limited, and why? *Current Directions in Psychological Science*, *19*(1), 51–57.

Derdikman, D., & Moser, E. I. (2010). A manifold of spatial maps in the brain. *Trends in cognitive sciences*, *14*(12), 561–569.

Fiete, I. R., Burak, Y., & Brookings, T. (2008). What grid cells convey about rat location. *The Journal of Neuroscience*, *28*(27), 6858–6871.

Fischer, M. H. (2001). Probing spatial working memory with the corsi blocks task. *Brain and Cognition*, *45*(2), 143–154.

Franklin, S., Madl, T., D'Mello, S., & Snaider, J. (in review). LIDA: A Systems-level Architecture for Cognition, Emotion, and Learning. *IEEE Transactions on Autonomous Mental Development*.

Franklin, S., & Patterson Jr, F. (2006). The lida architecture: Adding new modes of learning to an intelligent, autonomous, software agent. *pat*, *703*, 764–1004.

Franklin, S., Strain, S., Snaider, J., McCall, R., & Faghihi, U. (2012). Global workspace theory, its lida model and the underlying neuroscience. *Biologically Inspired Cognitive Architectures*.

Freeman, W. J. (2002). *The limbic action-perception cycle controlling goal-directed animal behavior* (Vol. 3).

Fuster, J. M. (2002). Physiology of executive functions: The perception-action cycle. In D. T. Stuss & R. T. Knight (Eds.), *Principles of frontal lobe function* (p. 96108). New York: Oxford University Press.

Hirtle, S. C., & Jonides, J. (1985). Evidence of hierarchies in cognitive maps. *Memory & Cognition*, *13*(3), 208–217.

Kanerva, P. (1988). *Sparse distributed memory*. MIT Press.

Lampton, D. R., McDonald, D. P., Singer, M., & Bliss, J. P. (1995). Distance estimation in virtual environments. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 39, pp. 1268–1272).

Madl, T., Baars, B. J., & Franklin, S. (2011). The timing of the cognitive cycle. *PloS one*, *6*(4), e14803.

Mason, W., & Suri, S. (2012). Conducting behavioral research on amazons mechanical turk. *Behavior Research Methods*, *44*(1), 1–23.

McCall, R., Franklin, S., Friedlander, D., & DMello, S. (2010). Grounded event-based and modal representations for objects, relations, beliefs, etc. *FLAIRS-23, Daytona Beach, FL.*

Moser, E. I., Kropff, E., & Moser, M.-B. (2008). Place cells, grid cells, and the brain's spatial representation system. *Annu. Rev. Neurosci.*, *31*, 69–89.

Paolacci, G., Chandler, J., & Ipeirotis, P. (2010). Running experiments on amazon mechanical turk. *Judgment and Decision Making*, *5*(5), 411–419.

Ramamurthy, U., Franklin, S., & Agrawal, P. (2012). Self-system in a model of cognition. *International Journal of Machine Consciousness*, *4*(02), 325–333.

Sato, N., & Yamaguchi, Y. (2009). Spatial-area selective retrieval of multiple object–place associations in a hierarchical cognitive map formed by theta phase coding. *Cognitive neurodynamics*, *3*(2), 131–140.

Snaider, J., & Franklin, S. (2012). Integer sparse distributed memory. In *Twenty-fifth international flairs conference.*

Sreenivasan, S., & Fiete, I. (2011). Grid cells generate an analog error-correcting code for singularly precise neural computation. *Nature neuroscience*, *14*(10), 1330–1337.

Waller, D. (1999). Factors affecting the perception of interobject distances in virtual environments. *Presence*, *8*(6), 657–670.

Wiener, J., Ehbauer, N., & Mallot, H. (2009). Planning paths to multiple targets: memory involvement and planning heuristics in spatial problem solving. *Psychological Research PRPF*, *73*(5), 644–658.

Winkelholz, C., & Schlick, C. (2007). Modeling human spatial memory within a symbolic architecture of cognition. *Spatial Cognition V Reasoning, Action, Interaction*, 229–248.

Yang, L.-L., & Hanzo, L. (2001). Redundant residue number system based error correction codes. In *Vehicular technology conference, 2001. vtc 2001 fall. ieee vts 54th* (Vol. 3, pp. 1472–1476).