knowledge as well as a visually guided heuristic within a validated cognitive framework. The goal behind this work is to provide an integrated model of visual- and non-visual planning, applicable to combined macro- and micro-planning tasks such as searching a building and all of its rooms. Such a model will be implemented in an existing cognitive architecture such as ACT-R (Anderson et al., 2004; Anderson, 2007) with perception-side heuristics developed for the task. The cognitive architecture leverages not just pre-defined, domain-specific symbolic knowledge, but also makes it clear how subsymbolic knowledge is acquired to optimize task performance. In this paper, we present two models that implement high-level and visual search within the ACT-R framework. We will discuss results of simulations with prototypical navigation problems. Our models, even though primarily predictive, show realistic results with respect to cognitive limits.

2 Recent work

Models of spatial navigation have addressed path finding as well as the representation mechanisms. For instance knowledge of locations may be arranged in two- or more dimensional arrays (e.g., Kosslyn, 1980; Glasgow & Papadias, 1998), or multiple layers of spatial representations (Kosslyn, 1994). A representation supporting path planning will not only need to store distances between locations, but also the affordances of connections between them: some routes may be unavailable or suboptimal. On the side of path finding algorithms, Fum & del Missier (2000) present data showing how humans develop spatial plans in a 2D environment that contained a varied number of obstacles. Subjects performance was measured by the time to find the path and the number of unnecessary steps taken compared to the optimal path to the goal (errors). The number of turns was found to be a crucial predictor of planning time. The model proposes that subjects choose locally optimal paths (a hill-climbing strategy) and minimize the number of turns; they do not develop a complete plan before committing to initial steps. Our model is similar in that the cognitive level also prefers to backtrack locally in order to avoid long-term memory needs. In the visual model presented in this paper, visually guided local planning results in long, straight lines with few turns.

The integration of cognitive architectures with path finding approaches represents a third field of work. (Chandrasekaran, 2006; Dye, 2007) attempt to blend cognitive and perceptual factor in navigation and planning applications in their work on implementing diagrammatic reasoning in cognitive architectures. Lathrop & Laird (2007) extend the Soar architecture with a visual system that is able to render imaginary drawings in tasks that involve reasoning about the relationships of geometric objects. As they show, these tasks can be carried out on a non-visual, largely symbolic (or arithmetic) level. Visual imagery, however, speeds up the process (in line with data). Within the context of the ACT-R architecture, much work has been done on the problem of spatial planning and representation, including issues of adaptivity in planning (Fu, 2003), encoding of spatio-temporal stimuli (Johnson et al., 2002), visualization capacity of spatial paths(Lyon et al., 2008), spatial perspective taking for planning (Hiatt et al., 2004) and architectural modules for neurally plausible navigation in 3D spaces (Schunn & Harrison, 2001). In this paper, we address the case of route planning based on information that is available externally, rather than purely held in memory.

3 Model overview

Our two models combine in planning a method to arrive at a goal point, given a current location and a partially observable environment, which constrains individual steps.

The representation in the first model is abstract and memory-based. Given the immediate surroundings, options for a next step can be determined. The abstract representation allows us to store locations, their local options (i.e., outgoing paths) and how useful these options are with respect to a given goal. The abstract situation is encoded as declarative knowledge, indicating the possibility to traverse from point *A* to point *B*. Such situations may be recalled with the help of cues: this could be a goal location *C*, but also a preceding traversal. This makes goal-directed planning possible and predicts that sequences of decisions rather than just individual steps are learned and combined during path planning. The traversal of a maze according to such abstract representations amounts to traversing a graph structure. The abstract representation is grounded in perceptual components of the architecture in visual concepts (landmarks). Landmarks may serve as cues to retrieve location-based information during planning. It should be noted that the abstract representation of affordances alone would be insufficient to explain classic mental imagery data (Shepard & Metzler, 1971).

The second model concerns the visual system, which has access to the part of the visual scene that it attends to at the time (cf., the visual representations in Glasgow & Papadias's (1998) model). Our visual model represents possible paths as largely straight lines from the point of attention to reachable (immediate) goals. Given a first-person perspective, such lines will equal lines of sight; given a two-dimensional (2D) representation of the maze (as in our experiments), possible paths are detected as straight lines that are uninterrupted by walls. In the latter (2D) case, knowledge of previous decisions is strongly grounded in the visual world. Memory that would otherwise be declarative on the abstract level is externalized in the visual scene. The role of the perceptual, visual module that the visual model depends on is to identify traversable shapes and select promising ones: the adopted heuristic is to choose the route that ends up as close to the goal as possible. Thus, the visual module is to convey a sense of distance from the end of straight lines to the goal. Naturally, the two models have strengths and weaknesses. Learning to navigate around a city, for instance, will be a memory task, while most navigation in a park unknown to the subject would be better suited to the visual approach. The two models are intended to combine; we will describe our approach for their integration.

4 High-level Planning Model

4.1 Model

The memory-based model primarily leverages the subsymbolic mechanisms of declarative memory in the ACT-R architecture. While subsymbolic mechanisms also play a large role in procedural memory, there were a number of reasons for focusing initially on declarative subsymbolic processes:

- Declarative memory provides a more direct integration path with symbolic and perceptual information since those sources of information are initially (and perhaps largely) stored in declarative memory.
- Decision-making typically follows a path by which it starts with declarative processes that are then ultimately (if possible) compiled into procedural structures. As such, a declarative account is an enabling account to a subsequent procedural one.
- Subsymbolic procedural processes largely consist of a utility calculus determining the selection of production rules. Since it bears a strong resemblance to reinforcement learning techniques that have been applied extensively to navigation and planning tasks, there are more limited possibilities for improvement there.
- Declarative subsymbolic mechanisms reflect more complex and discriminating statistical and semantic factors than procedural subsymbolic mechanisms that give them greater power in complex domains, as determined from experience in applying ACT-R to the game of Backgammon (Sanner et al., 2000).

Before describing the subsymbolic mechanisms, we will briefly sketch out the symbolic level of the model that leverages those mechanisms, more specifically the declarative representation of the problem and the production rules that manipulate them. Declarative memory items, *chunks*, may be of two basic types: location chunks that define states of the system, and path chunks that define transitions from one state to another. In addition, there is one basic goal chunk type representing the planning process that holds three pieces of information: the current state, the desired (goal) state and process information such as the current step of the process and ultimately an intermediate state determined by that level of planning. These goals are constructed during path planning and will enter declarative memory. There, they constitute a record for purposes of backtracking as well as learning across planning episodes that would allow previous partial solutions to be reused. As such, one can view specific paths between states as a special case of past planning solutions. Similarly, the productions that act on those representations are equally straightforward:

- The key production retrieves a path from memory this is the key step that leverages declarative subsymbolic processes.
- A subsequent production checks if the path starts at the current location if so it advances along the path and subgoals the process of moving from the endpoint of that path to the final destination.
- If that is not the case, another production subgoals the process of getting from the current location to the paths starting point, and changes the goal to get from that point to the final destination for later resumption.
- If a subgoal (such as the one set by the previous production) has been completed, then another production attempts to retrieve the next highest goal and resume it.
- If no such higher goal exists, the process terminates in success.
- If no path can be found from the current location that has not already been taken (to avoid looping), the process terminates in failure.

To focus on determining the ability of subsymbolic mechanisms to find the right path, we avoided in this basic model the use of backtracking mechanisms that could assure successful planning through exhaustive consideration of all possible routes. However, such a process could be easily integrated to leverage the record of previous (sub)goals in declarative memory, and would be triggered at the last step of the process described above. The key step is therefore the retrieval from memory of the next path to be considered, which attempts to maximize the activation of the chunk retrieved. Memory activation is a sum of three terms (not counting a stochastic noise term to be discussed later), each of which aims to capture another statistical factor in the path selection process. The first term (a.k.a base-level) attempts to capture frequency and recency, following the power laws of practice and forgetting, respectively, and can be interpreted as the log odds that a given path will be the right one. In contrast to this context-free term, the second term (a.k.a. spreading association) attempts to capture the log likelihood (in Bayesian terms) that this path is the right one given the current context elements. Technically, each context element (in practice, the current and destination locations) spreads activation to related chunks, in this case paths between locations. In practice, the additive spreading activation mechanism embodies the naïve Bayes assumption of independence between context elements, so there is no guarantee (and indeed it often fails) that a path related to (not necessarily connected to) both locations would be a good one to get from one to the other. Hence the necessity of a more explicit semantic factor reflected in the third term (a.k.a. partial matching), which imposes a penalty on chunks proportional to their dissimilarity with the pattern specified in the retrieval request. In practice, since the location chunks to be retrieved are represented with their starting and destination locations, we desire for the starting and destination locations to be as similar as possible to the current and final destination locations, respectively. These factors added together effectively balance the requirements that paths selected in the retrieval process reflect the constraints favoring common paths (a powerful heuristic reflected in all human actions), those often taken from and to the specific locations, and those that make progress toward the final goal.

While those factors should ultimately be learned from experience using architectural learning mechanisms, in the absence of a model integrating the various levels previously described, we have set these parameters to reflect idealized values reflecting their semantics and the convergence point of the learning processes. Accordingly, the base-level activation of a path chunk is set to the log odds that that path would be on the shortest path between two locations, averaged over all possible starting and ending location pairs. The strength of association from a location to a path is set to the log odds that the path is part of the shortest path from that location to or from another, averaged over all other locations. Finally, the similarities between locations using in matching the path starting and destination locations are set to reflect the semantics of the domain, reflecting a general sense of proximity, basically as a negative exponential function of distance between the points. Unlike the first two factors which reflect the statistics of previous problem-solving experience, the third factor most likely reflect the interaction with the domain itself such as resulting from the visual level. The use of these activation factors is similar to that of Lyon et al.'s (2008) model of 3D path planning.

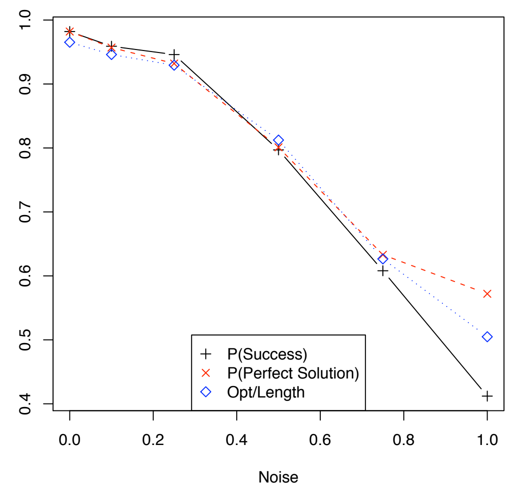

Figure 1: Model performance degrades with increased noise levels w.r.t. three measures: success rate, rate of perfect solution and optimal-to-obtained-path length ratio.

4.2 Simulation

We tested this high-level planning model in two idealized but naturally scalable environments: *trees* and *grids*. A tree structure is meant to approximate the hierarchical organization of structures such as buildings, with heavily travelled central connectors (elevators, main hallways) and increasingly localized destinations (wings, rooms). A grid structure is meant to approximate the regular pattern of city streets. We ran our model over structures of both types over a range of complexity, topping (mostly for incidental computational reasons) at trees of depth 4 (approximately the most levels of organization in hierarchical networks such as large buildings of road networks) and 5x5 grids (which might seem small but provides many combinatorial possibilities and could be applied in a self-scaling manner). The only parameter that we manipulated in the model was the amount of activation noise. This was designed both to reflect the stochastic nature of human cognition (which may seem like a purely limiting factor but is actually a powerful feature to avoid both predictability and local minima in search, e.g. West et al., 2005) and temper the assumption that the activation calculus parameters were set to idealized values that learning mechanisms would not perfectly reach.

Figure 1 displays the performance of the model as a function of activation noise, averaged over all environment structures and sizes, in terms of three measures. They are the probability of the planning process to successfully reach the goal (far from assured given the constraint of

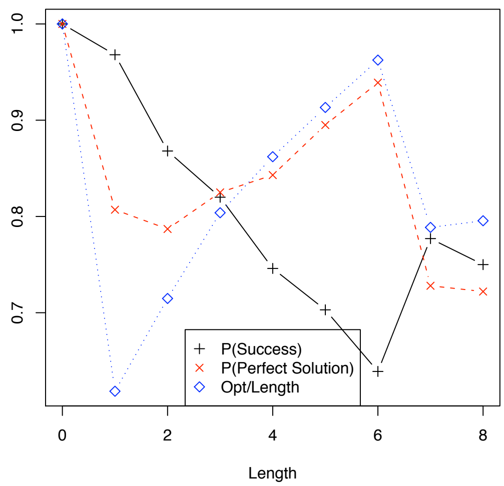

Figure 2: The model predicts lower success rates for longer solutions. When a solution is found, it is usually less than 25% longer than the optimal solution.

never using the same path twice and the lack of backtracking), the probability of finding the shortest path assuming success, and the ratio of shortest path to actual path length (again assuming success). While, as expected, all three performance measures decrease sharply (in remarkably correlated fashion) as a function of activation noise, a typical noise value of 0.25 used in many models of memory-based decision-making (e.g., Gonzales & Lebiere, 2005) provide over 90% performance on all measures.

Figure 2 displays the performance of the model as it scales to paths of increasing lengths, averaged over all noise values plotted in the previous figure. An interesting pattern arises for paths of length 2 to 6, displaying an inverse relationship between decreasing probability of success but increasing probability of finding a path of minimal length assuming success. This pattern is however not present for paths of length 7 and 8, which consist only (for artefactual reasons) of grid structures. In general, the probability of success seems to scale well (over this admittedly limited range) as a function of path length, especially considering the wide sampling of high noise values reflecting in this average.

Future experiments will involve somewhat more realistic environmental structures, including small world networks as a generalization of tree structures that better reflects the overall road network, and probabilistically introducing disabled or one-way paths in grid structures, to more accurately reflect traditional city environments (e.g. Manhattan).

5 Visual Navigation Model

5.1 Mazes

In line with our original motivation to model navigation tasks that are closer to real-life navigation over short distances, we also investigated path planning using *mazes*, which allows us to manipulate the complexity of the task along several dimensions. Our mazes are of size 10 by 10 squares or larger. They are relatively easy to solve: they do not generally contain many long dead-end routes, requiring the model to backtrack only occasionally. Other, more complex mazes, would define the task primarily as a memory problem: there, remembering previous branching points is the most important sub-task of solving algorithms.

Mazes were generated using a dynamic programming algorithm commonly known as *Eller's algorithm*. Start and end points were always located on opposites sides of the maze.

5.2 Model

The visual navigation model always chooses the best route along a line of sight from the current location; the best route is the one that transports us as close to the goal as possible. Routes that avoid bringing the model to previously visited parts of the territory are preferred. If a route brings us away from the goal, then we are careful to detect alternative routes along the way: the model inspects the areas left and right at each step, stopping when there is a way out. Such a way out is likely to be more useful than to retreat further from the goal.

In the case of mazes, visual navigation relies on the recognition of *passages*. Such passages are stretches of straight lines, translating to a line of sight in the corresponding first-person perspective environment. The visual model recognizes passages beginning at the location of visual attention. The model then commits to the passage that brings it the closest to the goal. We call the intermediate target location chosen this way the *next stop*.

While traveling to the next stop, the model notes its locations; the most recent ones are accessible in form of *visual finsts* (Pylyshyn, 1989). This mechanism allows the model to distinguish previously visited locations from novel portions of the maze. While making its way to the next stop, the model does not normally inspect the surroundings for possible alternative routes. We expect our model to be compatible with more recent, graded views of visual salience (Byrne, 2006), but the model presented here does not require any subsymbolic representation beyond the raw image of a maze.

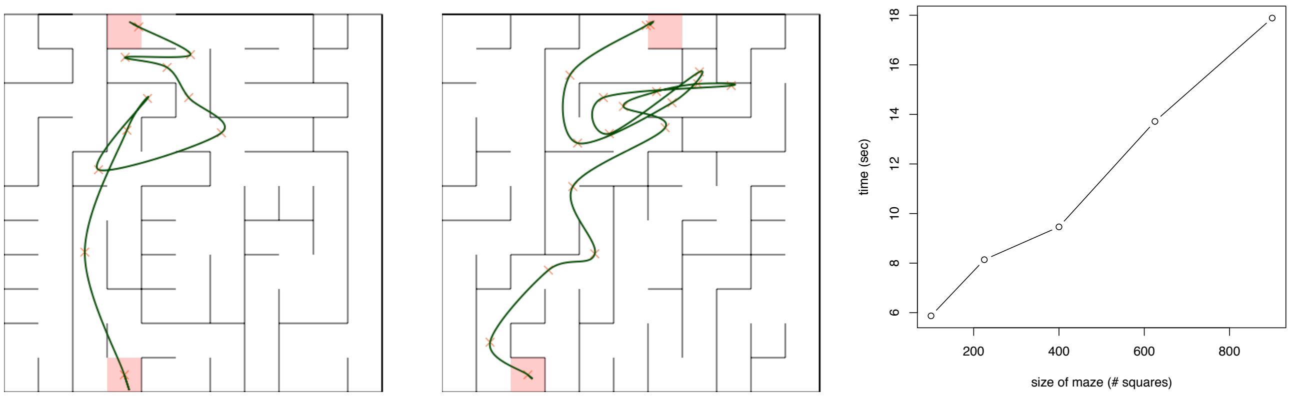

Figure 3: The visual model can find relatively straight paths easily, with limited capability to backtrack (left). Crosses mark decision points. The model fails where knowledge of the local surroundings would be helpful (center). Predicted solution times increase linearly with the size of the maze (right).

The model implements the following algorithm:

1. Set visual attention to the start point.
2. Identify straight lines (uninterrupted by obstacles) in all directions from the current point of visual attention.
3. Choose the line whose end point is closest to the goal.
4. Move visual attention along the chosen line towards the end point. While doing so: If entering a location that has been recently visited (visual finst), identify openings to the left and right, that is, uninterrupted, straight lines beginning at the current point of attention and extending orthogonally to the current track. If openings are found, abandon the earlier movement and continue with 3 (choosing the opening that is closer to the goal). If no openings are found, continue with 4.
5. Terminate if goal reached, or if number of steps exceeds the maximum.[1]

Following Salvucci (2001), the time to shift attention and encode the new location is determined as a function of location frequency f_i and eccentricity e_i (distance in units of visual angle) and constants K and k: $T_{enc} = K * [-\log f_i] * e^{ke_i}$. In the context of the maze, where most locations are novel, we assume $f_i = 0.5$. The model does not take additional motor activity into account.

The visual navigation model is conceived as a strategy applicable whenever the goal is deemed easy to reach. Such a goal could also be a subgoal, with the visual strategy tying in to the cognitive strategy to find micro-solutions and guide exploratory behavior. It should be noted that visual navigation does not require the declarative, explicit storage of a branching point. Any backtracking is visually guided and constrained by visual finsts. As a consequence, the visual model inherits the limitations of visual memory: primarily, there is only a small number of finsts available. Without a memory-based component storing branching points explicitly, the visual model can only identify 5 or so locations as previously visited (if we accept the default assumption of about 5 finsts). The visual model alone may, given a sufficiently complex navigation task, even get caught in a loop, visiting the same locations again and again.

5.3 Strengths and Weaknesses

For situations that do not need much backtracking, the visual model predicts efficient paths that are found quickly (Fig. 3, left). So, for many real-world situations that hardly compare to the pathological case of a maze with numerous dead-end paths, the visual model may offer a compelling explanation. It also provides a simply account for the correlation between turns and solution time found by Fum & del Missier (2000). At the same time, the model also predicts longer solution times for cases where backtracking is important. In particular, this part of the model predicts long solution times (or fails to provide a solution) in cases were solution paths first lead the attentional focus away from the goal, while offering more direct paths that ultimately lead to a dead end (Fig. 3, center).

5.4 Simulation

We ran the visual model using randomly generated mazes ($100 - -900$ squares), varying the size of the maze and also the number of visual finsts ($2 - -40$). As dependent measures, we show

[1] We use a heuristic of $w^{1.5}$, where w is the width of the maze in squares to determine the maximum number of steps.

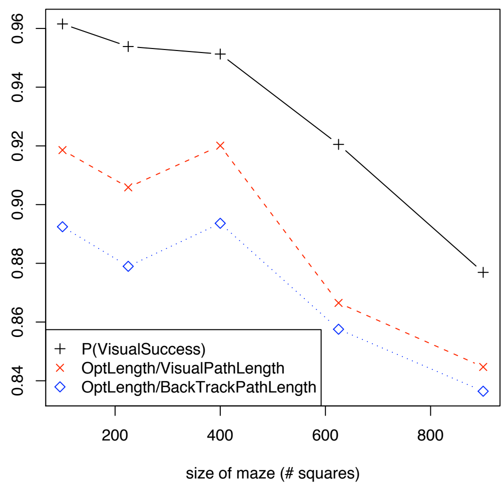

Figure 4: The performance of the visual model drops in mazes bigger than about 20 by 20 squares. Data aggregated over number of finsts. We show success rates, ratio of minimal vs. obtained path length, and, for comparison, ratio of optimal vs. backtracking-based path length. The backtracking baseline delivers consistently longer paths than the visual model.

- the proportion of successful runs (reaching the goal) within the allotted maximum number of steps (*P(VisualSuccess)*)
- the proportion of the minimal path length (from start to goal, *OptLength*) and the length of the path suggested by the visual model (*VisualPathLength*). A value of 1 would indicate that the visual model has chosen the shortest possible path.
- the proportion of the minimal path length and the length chosen by an alternative backtracking model, which will always succeed. It is a variation of the visual model: it moves square by square and backtracks to previously visited choice-points, but prefers squares that are visually closer to the goal.

The visual model predicts a mostly linear increase in time-to-solution with size (side length) (Fig. 3, right). This is expected as the model generally optimizes decisions locally rather than searching the whole maze. The model predicts good performance with respect to the length of the solutions (Fig. 4); for mazes of any size, the solutions resemble the optimal solution better than a simple backtracking algorithm that explores the maze square by square. Performance drops in mazes larger than about 400 squares. This drop in performance also holds with finsts held at 10. Even though the number of finsts is critical for the model to succeed, 5–10 finsts are sufficient to solve the mazes (Fig. 5), and performance does not improve beyond that.

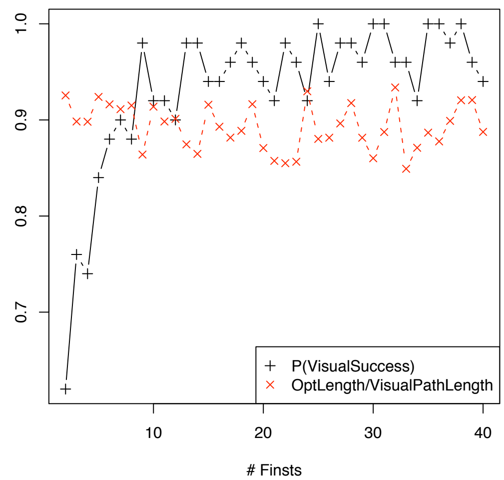

Figure 5: Model performance appears to stabilize after 5–10 visual finsts. Data aggregated over maze sizes.

Note that solution times for mazes predicted by visual navigation depend on the number of decision points. This is a property of the visual model inherent in the demands of attention shifts. This explains data described by Fum & del Missier (2000). Unlike in their model we define branching points as end points of straight lines, or points at which the visual model recognizes branching points explicitly while backtracking. When making progress toward the goal, a decision point will often imply a change of direction. The temporal penalty resulting from the decision point results from visual activity (search for a new straight line) rather than from storing the location in memory.

6 Integrating the Models

The combined model (not presented formally here due to space constraints) will attempt to retrieve any path from declarative memory. If none is found, visual search produces the next intermediate step. Even with the visual model in place, visited locations and exploited pathways are stored in declarative memory, and reinforced upon repeated presentation. They can be retrieved by the memory-based model should the need arise: either, when no visual information is available, or, when the visual system reaches its narrowly defined limitations. In particular where the need to identify past locations requires the use of memorized knowledge rather that visual finsts, the integrated model will abandon the visual strategy. When presented with an unknown maze, most search will be visual initially, progressing to memory-based navigation later.

Combining the visual and memory-based models, we see an opportunity for a third optimization. Consider again

Fig. 3 (center): the model chooses the wrong direction (East), backtracks West, and then attempts a similar direction again (North East) rather than jumping South West (leading it temporarily away from the goal). Once the large space adjacent to the goal is identified as a useful location to navigate too, the model can leverage knowledge of the local surroundings to circumnavigate the Western wall. Recognizing a pattern of 3 by 3 squares would be enough to see that the wall to the West is only one square long, and that there is an opening further South. We expect this type of knowledge to be retrieved from declarative memory, once the visual pattern has been recognized. Translated to practical tasks, the model has to *know* about the affordances of a door (access to a bigger space behind it) or a chair (can be climbed over if necessary).

7 Conclusion

In this paper, we have presented two models of navigation implemented using the ACT-R cognitive architecture. They address two aspects of the navigation problem: how to plan a good path efficiently, given knowledge about various portions of the route, and how to plan the path given visual information about the route. In the context of mazes, the visual model may provide a good heuristic for exploration, helping in the acquisition of declarative and subsymbolic knowledge. The evaluation of the models shows that the memory-based approach, guided by subsymbolic information, is able to find solutions within grid and tree layouts that are close to perfect. The visual approach yields good results, which begin to degrade at a critical maze size. In line with the literature, we infer a threshold of visual finsts at 5–10, beyond which the visual model does not gain in performance. Integration of the two models would provide a stepping stone toward the general goal of providing a general unified model of planning for a broad range of environments and requirements.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?*. Oxford, UK: Oxford University Press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Quin, Y. (2004). An integrated theory of mind. *Psychological Review*, *111*, 1036–1060.

Byrne, M. D. (2006). A theory of visual salience computation in ACT-R. In *Presentation at the Thirteenth Annual ACT-R Workshop*.

Chandrasekaran, B. (2006). Multimodal cognitive architecture: Making perception more central to intelligent behavior. In *Proc. AAAI National Conference on Artificial Intelligence*, (pp. 1508–1512).

Dye, H. A. (2007). A diagrammatic reasoning: Route planning on maps with ACT-R. In *Proc. Eigth International Conference on Cognitive Modeling*, Ann Arbor, MI.

Fu, W.-T. (2003). An ACT-R adaptive planner in a simple map-navigation task. In F. Detje, D. Doerner, & H. Schaub (eds.), *Proc. Fifth International Conference on Cognitive Modeling*, (pp. 99–104), Bamberg, Germany.

Fum, D., & del Missier, F. (2000). Climbing the mazes: A cognitive model of spatial planning. In *Proc. Third International Conference on Cognitive Modeling*, (pp. 126–133), Veenendaal, The Netherlands. Universal Press.

Glasgow, J., & Papadias, D. (1998). Computational imagery. In P. Thagard (ed.), *Mind Readings*. Cambridge, MA: MIT Press.

Gonzales, C., & Lebiere, C. (2005). Instance-based cognitive models of decision making. In D. Zizzo, & A. Courakis (eds.), *Transfer of knowledge in economic decision making*. New York: Palgrave McMillan.

Hiatt, L. M., Trafton, J. G., Harrison, A. M., & Schulz, A. C. (2004). A cognitive model for spatial perspective taking. In *Proc. Sixth International Conference on Cognitive Modeling*, (pp. 354–355), Pittsburgh, PA.

Johnson, T., Wang, H., Zhang, J., & Wang, Y. (2002). A model of spatio-temporal coding of memory for multidimensional stimuli. In *Proc. 24th Annual Meeting of the Cognitive Science Society*, Fairfax, VA.

Kosslyn, S. M. (1980). *Image and Mind*. Cambridge, MA: Harvard University Press.

Kosslyn, S. M. (1994). *Image and Brain*. Cambridge, MA: MIT Press.

Lathrop, S. D., & Laird, J. E. (2007). Towards incorporating visual imagery into a cognitive architecture. In *Proc. Eighth International Conference on Cognitive Modeling*, (pp. 25–30), Ann Arbor, MI.

Lyon, D. R., Gunzelmann, G., & Gluck, K. A. (2008). A computational model of spatial visualization capacity. *Cognitive Psychology*, *57*, 122–152.

Pylyshyn, Z. W. (1989). The role of location indexes in spatial perception: A sketch of the finst spatial-index model. *Cognition*, *32*, 65–97.

Salvucci, D. D. (2001). An integrated model of eye movements and visual encoding. *Cognitive Systems Research*, *1*, 201–220.

Sanner, S., Anderson, J. R., Lebiere, C., & Lovett, M. (2000). Achieving efficient and cognitively plausible learning in backgammon. In *Proc. Seventeenth International Conference on Machine Learning*, (pp. 823–830), San Francisco, CA.

Schunn, C. D., & Harrison, A. M. (2001). Act-rs: A neuropsychologically inspired module for spatial reasoning. In *Proc. Fourth International Conference on Cognitive Modeling*, (pp. 267–268), Mahwah, NJ.

Shepard, R. N., & Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science*, *171*, 701–703.

West, R. L., Stewart, T. C., Lebiere, C., & Chandrasekharan, S. (2005). Stochastic resonance in human cognition: ACT-R vs. game theory, associative neural networks, recursice neural networks, q-learning, and humans. In *Proc. 27th CogSci*, Mahwah, NJ. Erlbaum.