TOPICS
TOPICS IN COGNITIVE SCIENCE

Topics in Cognitive Science 3 (2011) 796–818

ISSN: 1756-8757 print / 1756-8765 online
DOI: 10.1111/j.1756-8765.2010.01130.x

Exploring the Functional Advantages of Spatial and Visual Cognition From an Architectural Perspective

Scott D. Lathrop,[a,]* Samuel Wintermute,[b] John E. Laird[b]

[a]*Department of Electrical Engineering and Computer Science, United States Military Academy*
[b]*Department of Electrical Engineering and Computer Science, University of Michigan*

Received 4 December 2009; received in revised form 28 April 2010; accepted 20 September 2010

Abstract

We present a general cognitive architecture that tightly integrates symbolic, spatial, and visual representations. A key means to achieving this integration is allowing cognition to move freely between these modes, using mental imagery. The specific components and their integration are motivated by results from psychology, as well as the need for developing a functional and efficient implementation. We discuss functional benefits that result from the combination of multiple content-based representations and the specialized processing units associated with them. Instantiating this theory, we then discuss the architectural components and processes, and illustrate the resulting functional advantages in two spatially and visually rich domains. The theory is then compared to other prominent approaches in the area.

Keywords: Cognitive architecture; Mental imagery; Spatial cognition; Visual cognition

1. Introduction

Space and vision are prominent in our experiences as humans. We live in a richly visual world and are constantly and acutely aware of our position in space and our surroundings. In addition to this seemingly precise awareness, we are also able to reason abstractly, use language, and construct arbitrary hypothetical scenarios. In this contrast, we see important questions for cognitive science: How can information from different senses, at different levels of abstraction, be fluidly used in decision making? What functional role does specialized spatial and visual processing play in cognition?

Correspondence should be sent to Scott D. Lathrop, Department of Electrical Engineering and Computer Science, United States Military Academy, West Point, NY 10996. E-mail: scott.lathrop@usma.edu

*Views expressed are those of the author and do not reflect the official position of the U.S. Military Academy, the U.S. Department of the Army, the U.S. Department of Defense, or the U.S. Government.

This article presents a comprehensive account of an implemented cognitive architecture tightly integrating spatial and visual processing with symbolic processing, making perceptually grounded information a first-class participant in higher-level cognition. This article brings together two related lines of research (Lathrop & Laird, 2009; Wintermute & Laird, 2009) focusing on the design of the architecture, the capabilities gained by integrating spatial and visual information with symbolic abstractions, and the relationship to existing cognitive theories.

While this architecture is psychologically inspired, it is not a precise model of human behavior. Rather, the goal is a functional explanation of visuospatial cognition, in terms of what representations and processes can support human-level capabilities and performance. To this end, we limit what we borrow from psychological theories to those aspects that provide a clear functional benefit and have not incorporated details that solely improve model fidelity (the match to human data). This approach benefits the construction of AI systems; however, we believe it has the potential to reflect back and provide novel insights to psychology. Focus on functionality over fidelity allows us to provide clear arguments for the core components of the system that are likely to be valid independently of what commitments are made about the details required for a high-fidelity model.

That said, implementation is an important aspect of our approach, and many details must be specified before the architecture can actually produce behavior. In the discussion here, we try be clear about which choices we consider to be implementation details rather than theoretical commitments.

To constrain the higher-level cognitive aspects to an established theory, our architecture is designed as an extension of the Soar cognitive architecture (Laird, 2008; Laird, Newell, & Rosenbloom, 1987). The extension is called the Spatial/Visual System, or SVS.[1] One of our key claims is that mental imagery is functionally essential to spatial and visual cognition. Accordingly, previous research in mental imagery (Kosslyn, Thompson, & Ganis, 2006) influences the multiplicity of representations in SVS: *symbolic*, *quantitative spatial,* and *visual depictive*. Together these representations form a basis for spatial and visual processing from which spatial and visual cognition emerge. Although psychologists have debated for years over the types and details of mental imagery representations (Kosslyn, 1994; Kosslyn et al., 2006; Pylyshyn, 1973, 1981, 2002), there has been less emphasis on the functional value that mental imagery provides to human cognition and how such functionality can be realized in a general computational system, which is the focus of this research.

In addition to cognitive functions, the architecture accounts for many higher-level aspects of perception and action, because imagery, perception, and action share computational machinery in our architecture. While a complete architecture capable of humanlike perception and action control is far beyond the state of the art, we argue that including perceptual and motor processes in cognition can provide functional benefits, as has often been argued by theorists studying ‘‘embodied,’’ or ‘‘grounded’’ cognition (Barsalou, 2008; Grush, 2004), even when those systems are incompletely implemented. There is a large body of existing research from which we borrow in our work, and upon which our work can provide an interesting perspective. Some of these connections are discussed as they come up, and others are addressed in the discussion section.

2. Spatial and visual representations in cognition

Our hypothesis is that three distinct representations support spatial and visual cognition: *amodal symbolic*, *quantitative spatial*, and *visual depictive*, all of which are shown in Fig. 1. The amodal symbolic representation is useful for general reasoning (Newell, 1990). From a spatial and visual perspective, symbols may denote an object, visual properties of an object, and spatial relationships between objects. In general, these symbols are qualitative properties, rather than quantities. They are sentential, in that their meaning is dependent on context and interpretation rather than their spatial arrangement in memory. For example, the right-hand column in the first row of Fig. 1 represents two objects, a tree and a house with

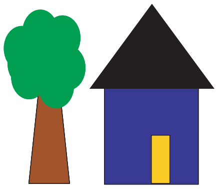
Representation	Information	Processing	Example
Symbolic	▪ Object identities ▪ Qualitative spatial and visual properties ▪ Non-perceptual information	Symbolic manipulation	object (tree) color (tree, green) left-of(tree, house)
Quantitative spatial	▪ Object labels ▪ 3D Spatial Properties (explicit) ○ General shape ○ Location ○ Orientation ▪ 3D Spatial Properties (implicit) ○ Size ○ Topology ○ Direction ○ Distance	Mathematical manipulation	tree: location <-2,4,0> orientation 0 shape coordinates <1,3,1>;<2,8,1>;<1,3,0>.. house: location <9,4,0> orientation 0 shape coordinates <8,3,1>;<2,3,1>;<4,3,0>..
Visual depictive	▪ Object labels ▪ 2D Visual Properties (explicit) ○ Shape ○ Texture ○ Empty space ▪ 2D Spatial Properties (implicit) ○ Location ○ Size ○ Topology ○ Direction	Mathematical manipulation Depictive manipulation	

Fig. 1. Multiple representations supported in Spatial/Visual System (SVS).

symbols denoting visual (e.g., color(tree, green)) and spatial (e.g., left-of(tree, house)) properties. In addition to spatial and visual properties, symbols can represent nonspatial or nonvisual content, which is necessary for associating an object with other modalities and concepts such as love, justice, or peace.

The quantitative spatial representation is also amodal but is perceptual-based in that it is an interpretation of visual, auditory, proprioception, and kinesthesis senses asserting the location, orientation,[2] and rough shape of objects in space. Computationally, the structure uses three-dimensional Euclidean space with symbols to label objects. Spatial processing is accomplished with sentential, mathematical equations. Motion can be simulated through linear transformations (i.e., translating, rotating, and scaling) or with nonlinear dynamical systems (e.g., Wintermute, 2009a). The second example in Fig. 1 represents the metric location, orientation, and rough shape of the tree and the house. Direction, distances between objects, size, and rough topology can be inferred from this information.

In contrast to the symbolic and spatial representation, both of which are sentential structures, space, including empty space, is inherent in the visual depictive representation. The depiction is from a privileged viewpoint, and the pattern structure resembles the objects in a perceived or imagined scene. Computationally, the depiction is a bitmap where the processing uses either mathematical manipulations (e.g., filters or affine transformations) or specialized processing that takes advantage of the topological structure. This imagery processing can be used to extract cognitively useful visual features (e.g., lines, curves, enclosed spaces) or for spatial reasoning where details of specific shapes are inherent to the problem. Similar to spatial imagery, visual imagery can manipulate the depiction to simulate physical processes.

Each representation has functional and computational trade-offs that specific tasks often highlight. For example, given appropriate inference rules and the symbolic representation in Fig. 1, one can infer that the green object (tree) is to the left of the blue object (house). However, one cannot infer the distance between the tree and the house or that the top of the house is shaped like a triangle. One can infer these properties from a symbolic representation only when the relevant property is encoded explicitly or when task knowledge supports the inference (e.g., if three lines intersect, then there is a triangle). Even if equivalent information is present in each representation, processing efficiency may vary across them. Mental imagery theorists (Kosslyn et al., 2006) have made similar arguments. However, their focus is primarily on the use of depictions, whereas we place equal emphasis on quantitative spatial representations.

These trade-offs can be characterized on a scale between discretion and assimilability (Norman, 2000) or scope and processing cost (Newell, 1990). The symbolic representation is high in discretion, as it conveys just enough information required for general reasoning. For example, the predicate description, on (apple, ground), is sufficient for general inferences such as ‘‘if the apple is on the ground, then grasp it.’’ Symbols have greater scope compared with the spatial and depictive representations in that they can represent incomplete knowledge such as negation and uncertainty, as in the statement, ‘‘If the apple is *not* in the tree but is on the ground *or* on the table, then grasp it.’’ Reasoning in this context does not have to be concerned about the exact location or shape of the objects.

At the other extreme, the spatial and depictive representations are low in terms of discretion and scope, as they provide many details but are limited to spatial and visual information. However, for spatial and visual properties, they have lower processing costs and are easier to assimilate. For example, from the image in Fig. 1, information such that the roof of the house looks like a triangle and overhangs the frame of the house is directly accessible. What is lost in scope is gained in efficiency.

3. Spatial and visual domains

To motivate the architectural discussion, we will use examples from two domains, Pegged Blocks World (Fig. 2; Wintermute & Laird, 2009) and Scout (Fig. 3; Lathrop & Laird, 2009). Pegged Blocks World problems are very simple but require precise spatial reasoning and broad generalization for success. The domain has been designed to be as simple as possible while still having sufficient complexity to demonstrate the usefulness of imagery. In contrast, the Scout domain is relatively complex, allowing for comprehensive agents to be created which demonstrate the broad capabilities of the architecture to include the use of multiple (i.e., symbolic, spatial, and visual) representations. Agents using the architecture have been developed for both of these domains and presented in previous work (Lathrop & Laird, 2009; Wintermute & Laird, 2009). In this work, the focus is on how the architecture supports the capabilities needed by these agents, rather than evaluating the performance of the agents in their domains.[3]

An agent in the Pegged Blocks World domain (Fig. 2; Wintermute & Laird, 2009) perceives blocks and can move them from place to place. Unlike similar domains, however, the blocks cannot be placed freely on a table. Instead, there are two fixed pegs, and each block must be aligned to one of the pegs—essentially, there can only be two towers, and their positions are fixed. The agent is presented with a simple goal, for example, to stack A on top of B on top of C on top of D, all on peg2. Blocks can be moved from the top of one tower to the other; however, the blocks vary in size, and the pegs are close enough that

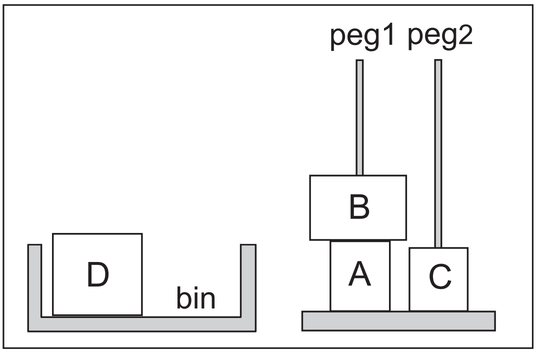


Fig. 2. A pegged blocks world state.

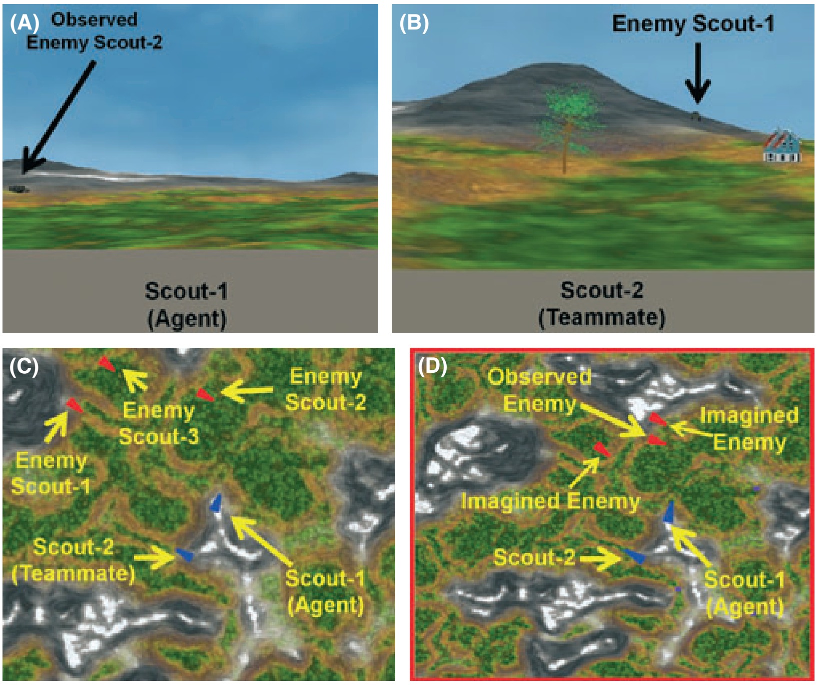


Fig. 3. Army scout domain. (A) Lead (agent) scout's view, (B) Teammate's view, (C) Actual situation (displayed on a map), (D) Agent's perceived map/imagined situation.

blocks can collide, depending on the exact sizes of the other blocks in the towers. Blocks can also be moved out of the way to a storage bin. The task of the agent is simply to build its goal stack while avoiding collisions and using the bin as little as possible.

The agent must solve a series of these problems, where the exact shapes of the blocks may vary, so the agent cannot perform well simply by memorizing an action sequence. The key means by which our agent solves this problem is by imagining the consequences of its actions in order to determine which actions would cause collisions.

The Scout domain (Fig. 3; Lathrop & Laird, 2009) is motivated by the U.S. Army's efforts in developing autonomous, robotic scouts for reconnaissance missions. A two member scout team's goal is to keep their higher command informed of the opposing force's movements by periodically sending observation reports (through the lead scout) of their best assessment of the enemy's location. To provide this information, the team must continually improve their positions in order to gain and maintain visual observation of the approaching enemy.

Consider the problems encountered by an agent performing the task of the lead scout. The agent perceives a certain view of the world (Fig. 3A), and its teammate perceives another (Fig. 3B). Both reflect some actual situation (Fig. 3C). The agent must combine information it directly perceives, information communicated from its teammate, and background knowledge (e.g., enemy tactics) to form hypotheses about where the enemy is located. To help with this analysis, the agent can look at a terrain map of the area and use this as a context in which to imagine these hypotheses (Fig. 3D).

Fig. 4 shows imagery supporting this processes in our system (the exact mechanisms will be elaborated shortly). In Fig. 4(A), the agent recognizes a likely path between an enemy location and a hypothetical destination. In Fig. 4(B), the agent determines what portion of the hypothetical path it could view by generating a visual depiction of its view area (dark blue) and overlaying it on the recognized path (orange). This reasoning, combined with similar analysis of other likely enemy paths from both the agent's current and simulated alternative perspectives and the teammate's current and simulated alternative perspectives, enables the agent to decide whether and where to reorient itself, its teammate, or both. Similar to Pegged Blocked World, the agent solves these problems by imagining the situation and the outcome of its actions before choosing an action to execute.

4. Architectural design

The overall design of SVS is shown in Fig. 5. Combined with the existing Soar architecture, this results in a comprehensive cognitive architecture for spatial and visual processing. A cognitive architecture is a set of fixed structures and mechanisms proposed to support general cognition (Anderson, 2007; Langley, Laird, & Rogers, 2009). A model of human behavior or an artificial agent is realized in such a system by adding task-specific knowledge.

The existing Soar architecture is shown at the top of Fig. 5. Soar contains a symbolic working memory, through which processes within Soar communicate. Connections to an external environment flow through SVS, where input and output occurs via changes to working memory. Other cognitive architectures, such as ACT-R (Anderson, 2007), use similar structures to connect to an external environment; the concepts behind SVS could be adapted to any system with such an interface.

Working memory serves as the locus of processing in Soar. Symbolic rules in long-term procedural memory match and modify its contents, mediated by operators (deliberate choices of internal or external actions) selected by a fixed decision procedure based on

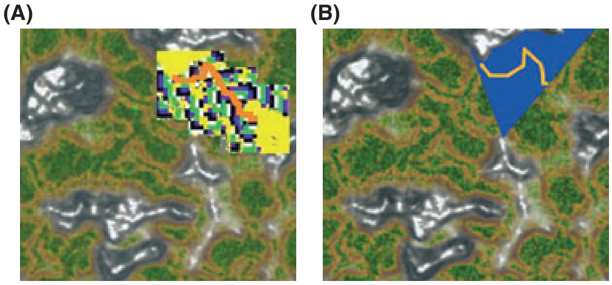


Fig. 4. Visual generation and recognition processes create visual depictions on a perceived map.

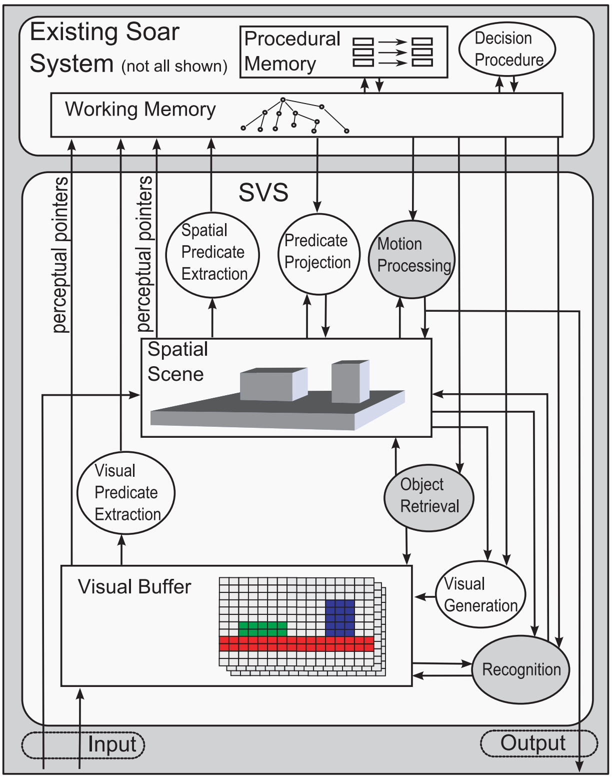


Fig. 5. Soar/SVS architecture. Boxes are short-term memories; circles are processes. Gray circles involve access to information in perceptual long-term memory (knowledge). There are implicit control lines (not shown) between working memory and all of the processes shown.

working memory. Further details of symbolic Soar processing will not be covered here, but they can be found in Laird (2008). Working memory fulfills the role of the amodal symbolic representation in our theory (Fig. 1). SVS adds a quantitative spatial representation, in the spatial scene (center of Fig. 5), and a visual depictive representation, in the visual buffer (bottom of Fig. 5). In addition to the two short-term memories, there is a long-term memory in SVS for visual, spatial, and motion data, called Perceptual LTM. To simplify the diagram, this memory is not explicitly shown, but it is accessed by object retrieval, motion processing, and recognition (discussed below).

Theoretically, all information in the system can be derived from depictive information added to the visual buffer by low-level vision. Conceptually, processes in Soar/SVS segment and recognize objects and estimate 3D spatial structure based on 2D visual information. However, complete domain-independent computer vision is beyond the state of the art. In practice, SVS is used in virtual environments without a complete visual system; many simulated environments represent the world in structures that can be directly input to the spatial scene. However, as will be explained, visual processing still plays a prominent role in our system. Visual imagery is cognitively useful, and it can be implemented without true perception. In addition, some aspects of object recognition can be fruitfully implemented, even though the broader problem remains unsolved.

In the following sections, each of the processes and memories inside SVS are briefly discussed.[4]

4.1. Perceptual pointers

While the memories in SVS contain primarily nonsymbolic information, there are symbols through which Soar can refer to elements within these memories. These identifying symbols, called *perceptual pointers*, are similar to the visual indices described by Pylyshyn (2001) for short-term visual memory, since the system ''... picks out a small number of individuals, keeps track of them, and provides a means by which the cognitive system can further examine them in order to encode their properties... or to carry out a motor command in relation to them'' (p. 130). That theory limits the number of objects to four or five, a limitation we do not model.

Fig. 6 shows pegged blocks world information represented in Soar/SVS. Soar's working memory is shown on the left. Structures in working memory are represented by a directed graph of symbols, which are matched and manipulated by rules. A portion of this graph is shown in the figure.

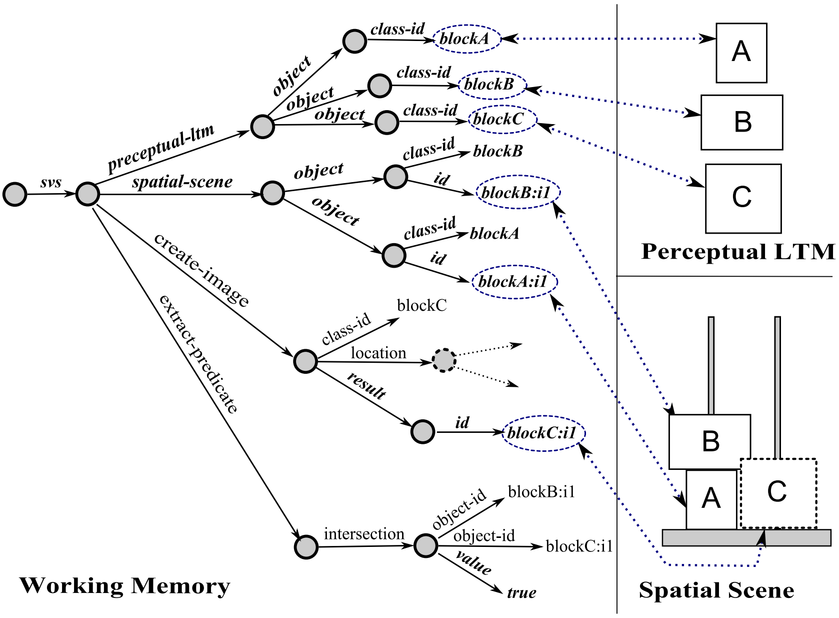


Fig. 6. Pegged blocks world information in working memory, perceptual LTM, and spatial scene. Working memory structures in bold italics are created by Spatial/Visual System (SVS).

Dotted arrows in the figure show perceptual pointers, represented in working memory as symbols. Only those created by SVS have arrows in the figure, but other instances of the same symbols also are pointers to the same objects. Note that the visual and spatial details of the objects in SVS (e.g., their coordinates in space or their pixel values) are *not* represented in working memory. Working memory instead holds the perceptual pointers, along with qualitative information available through the predicate extraction processes outlined below. As discussed in detail later, this constraint is a key difference between this architecture and other approaches (Larkin & Simon, 1987) where visual or spatial details are represented in the same symbolic working memory as abstract information.

4.2. Memory encodings

Internally, the spatial scene is a set of 3D objects grounded in continuous coordinates. Symbolic perceptual pointers to the objects in the scene are represented in Soar's working memory, organized as a part of hierarchy tree of objects and their constituent parts. Only the leaves of this tree correspond to primitive polyhedrons, but nodes at every level are considered objects, enabling reasoning over the whole or individual parts. For example, the house in Fig. 1 might be encoded as two polyhedrons, one each for the roof and the frame, both as part of a ''house'' object. The agent can reason at either level, considering the roof independently, or the house as a whole. Between each object node in the hierarchy is a transformation node. Each transformation node contains a perceptual pointer to the relationship between the two objects, such as the way the roof is related to the house as a whole. For simplicity, Fig. 6 only shows the object nodes at one level of the hierarchy in working memory.

The internal encoding of the visual buffer is a set of bitmaps. Each bitmap in the buffer is called a *depiction*, and there exists a perceptual pointer in working memory for each depiction. Individual pixels in the depiction can be set to a color, or to a special value indicating emptiness. Typically, there is at least one depiction in the set representing the perceived scene from an egocentric viewpoint, but others may be created through imagery processing. Having a set of depictions allows multiple objects to exist at the same visual location facilitating topological predicate extraction (discussed next).

The internal representation of perceptual LTM is more heterogeneous than the other parts of SVS. It stores spatial objects and transformations, visual textures, and motion patterns. Long-term perceptual pointers are available to Soar for all of these constructs.

4.3. Predicate extraction

The predicate extraction processes serve to provide symbolic processing in Soar with qualitative properties of the contents of the spatial scene and visual buffer. These processes are architectural; there is a fixed set of properties that can be extracted, which are not learnable by the agent. In contrast to perceptual pointers, qualitative predicates are created in working memory only when requested by Soar. There is a substantial amount of qualitative information implicit in the memories of SVS, each of which can take considerable computations to derive, so this top-down control is needed to determine which qualitative structures are made explicit in order to make the system computationally tractable.

For the spatial system, there are three important kinds of relationships between objects that can be queried for: topology, direction, and distance, as illustrated in Fig. 7. Topological relationships describe how the surfaces of objects relate to one another. In the current implementation of SVS, this is simply whether or not two objects intersect.

Distance is similarly simple. Currently the system can query for the distance between any two objects in the scene, along the closest line connecting them. This information is non-qualitative, although it is certainly ‘‘less quantitative’’ than the contents of the spatial scene, as it reduces three-dimensional information to a scalar quantity. However, it is extremely useful in practice. For example, the closest obstacle to the agent might be detected by extracting the distance from the agent to all of the obstacles, and comparing to determine the closest.

Direction queries are implemented following the approach of Hernández (1994). For each object, a set of surrounding acceptance regions are defined, which roughly correspond to concepts like left, right, etc. An object is in that direction if it lies within the acceptance region. Every object has an associated ‘‘front’’ vector, defining an intrinsic frame of reference, upon which the regions are based; however, this could easily be extended to allow queries based on frames of reference of other objects, or a global coordinate frame.

The visual system also supports predicate extraction. For example, there is a predicate that reports whether or not a given depiction has any nonempty pixels. Typically, the result of visual generation and top-down visual recognition processes (discussed below) is a depiction that will have some pixels filled in if some property is true (e.g., an object exists), and none if it is not.

Fig. 6 shows a simple example of predicate extraction in Pegged Blocks World. The agent has created an image of block C on peg2 (a process which will be described in the next section) and used predicate extraction to detect that this imagined block intersects block B. To do this, rules created a query structure in working memory, which SVS processing detects and responds. The working memory representation has been simplified in the figure, but the essentials are the same.

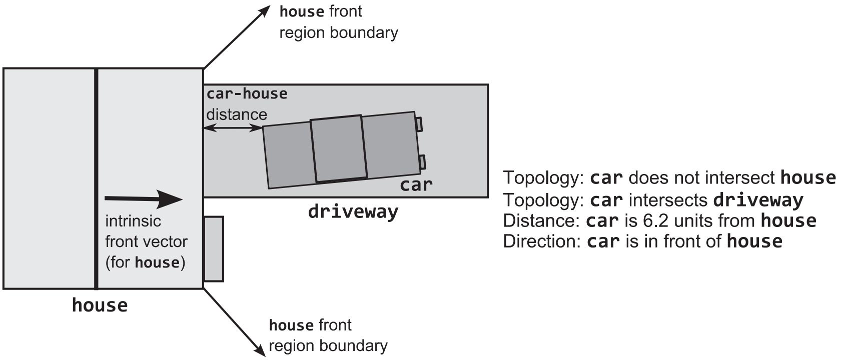

house front region boundary
car-house distance
car
driveway
intrinsic front vector (for house)
house
house front region boundary
Topology: car does not intersect house
Topology: car intersects driveway
Distance: car is 6.2 units from house
Direction: car is in front of house

Fig. 7. Information derivable through spatial predicate extraction.

In the Scout domain (Fig. 4), visual predicate extraction is used, for example, to determine the distance of the orange path overlapping the agent's blue view. Similar to spatial predicate extraction, the symbolic query structure is created in working memory to direct SVS with perceptual pointers referring to depictions and a corresponding color of interest rather than objects in the spatial scene. Visual processing searches the pair of depictions specified in the query for the overlapping orange (path) and blue (view) pixels. Working memory is augmented with the resulting distance, which can then be used for subsequent reasoning.

For both spatial and visual predicate extraction, our theory allows for a wider variety of predicates than what is present in the current implementation. In particular, previous implementations have allowed for spatial predicates encoding size, orientation, and a larger variety of topological relations, and visual predicates encoding topological relationships, size, and distance. The architectural commitment is that predicate extraction is a distinct, fixed process in the system, not that the implemented set of predicates is complete. As examples of the kinds of predicates an architecture might support, Table 1 lists some of the predicate extraction types supported by SVS or its predecessor systems.

4.4. Image creation

The information provided to Soar through perceptual pointers and predicate extraction is often not enough to allow general-purpose problem solving. In both example domains the process of choosing an action involves determining the consequences. Those consequences cannot be inferred solely with predicate extraction over perceived information. Instead, imagery processes must be employed. While imagery has often been proposed in the past as a means for problem solving (Helstrup, 1988), the exact means by which images are created in a problem-independent manner have been rarely specified. One of the contributions of this work is exploring this problem.

4.4.1. Predicate projection

Creating a new spatial image often involves transforming a qualitative description to a quantitative representation in the scene. SVS incorporates a predicate projection process to

Table 1
Predicate extraction types

Type	Example Properties
Direction	Left-of, Right-of, Front-of, Behind, Above, Below, Between
Distance	Scalar value
Topology	Disconnected, Externally connected, Partially overlaps, Tangential proper part, Nontangential proper part, Equals[5]
Size	Smaller, Larger, Equal
Symmetry	Horizontal-symmetry, Vertical-symmetry
Shape	Point, line, curve, triangle, enclosed-space
Color	Red, Green, Blue, White, Black

allow this (Wintermute & Laird, 2007). Predicates supported by SVS include geometric concepts like *hull* and *intersection*. A *hull* image is the convex hull of two or more objects in the scene, and an *intersection* image is the region of intersection of two or more objects. In addition, SVS supports predicates like *on* that specify qualitative information about spatial relationships between objects, but not what its shape is. As with predicate extraction, predicate projection is a fixed process, but there is no strong commitment that the current library of available operations (more fully described in Wintermute, 2009b) is complete.

In Pegged Blocks World, predicate projection can be used to create images of blocks in new positions. Fig. 6 shows this, as block C is imagined in the spatial scene. The figure omits the symbolic predicate projection command: This is the location structure on the create-image command in working memory. To arrive at the given image, this structure should encode that the block (C in this case) is on the surface of the table, centered with respect to peg2.

4.4.2. Memory retrieval

In contrast to predicate projection, which *describes* the qualitative properties of the spatial image, memory retrievals refer to specific objects and quantitative spatial relationships in perceptual LTM via perceptual pointers. These relationships and objects are then instantiated in a spatial image. This involves, for example, Soar requesting SVS to instantiate an object of a known type, or at a known location in the spatial scene. Images can be created by a combination of both memory retrieval and predicate projection processes, for example, by imagining a specific object in long-term memory at a qualitatively described location. The image of block C in Fig. 6 is an example of this, as the shape of the block is retrieved from LTM. Memory retrieval is also used in the Scout domain. The agent imagines enemy icons at specific locations on the perceived map in the spatial scene (Fig. 3D) based on perceived information or symbolic knowledge of enemy tactics (e.g., vehicle formations).

4.4.3. Motion simulation (motor imagery)

While the previous approaches to image creation are powerful, they are insufficient to solve problems involving nontrivial motion. For example, consider predicting if a turning car will hit an obstacle (Fig. 8). In the figure, the agent must determine whether or not the car can drive to the goal, via the waypoint, without colliding with an obstacle. The path of a car steering toward the waypoint, then toward the goal is shown. An agent able to derive this path can check if it intersects obstacles, solving the problem. For a general agent that is capable of the same range of behavior as humans, there are many types of motion that the system may need to predict: the outcome of its own actions, the motions of others, and of objects not under the control of any agent, such as the path of a bouncing ball.

In SVS, information of this type is encoded in *motion models* (Wintermute & Laird, 2008). By transforming one continuous spatial state to another, motion models provide fine-grained quantitative resolution needed for accurately representing motion. Motion models are in perceptual LTM and can be applied to any object in the spatial scene, resulting in a motion simulation. This simulation is a sequence of steps, controlled by Soar. The agent can use predicate extraction between each time step, gaining information from the simulation, such as whether the car intersects an obstacle in Fig. 8.

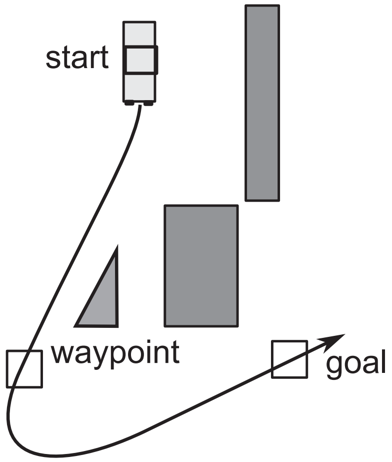


Fig. 8. An example motion problem.

Motion models can be used to simulate many kinds of motion, including the motion of the agent's own effectors. In a system incorporating real effectors, motion models should be intimately tied to their control (Wintermute, 2009a). For this reason, the motion processing module in Fig. 5 is connected to the output of the system.

4.4.4. Visual generation

If SVS were a more comprehensive model of human processing, perception would directly create structures in the visual buffer, and internal processing would derive the spatial scene from those structures. In imagery contexts, SVS supports modification of the visual buffer derived from contents in the spatial scene and under the control of symbolic structures. This process is called visual generation (Lathrop & Laird, 2007) and is useful in the support of further visual processing, such as predicate extraction or recognition. Through this process, the spatial imagery processes discussed previously are used indirectly to create visual images.

To support visual generation, symbolic structures in working memory first specify the object(s) in the spatial scene to generate and corresponding texture(s) from perceptual LTM to render via their perceptual pointers. Since visual generation converts 3D spatial information to 2D visual information, symbolic structures must also specify a particular perspective in the spatial scene. In many cases, the perspective is simply the egocentric view of the agent. However, from a computational standpoint, the viewpoint of the agent in the scene is not privileged; it is just as easy to generate images from other viewpoints, such as from the perspective of another agent or from a top-down view as in the Scout agent's imagined scenes (Figs. 3D and 4). In a precise model of human imagery, this aspect of the architecture would likely need to be reconsidered, as it is not clear if a human can generate images from any perspective.

Once specified, the symbolic object(s), texture(s), and perspective perceptual pointers are transmitted to SVS's visual generation process for generating the 2D scene in the visual buffer using standard graphics rendering techniques. The result of this process is a new depiction in the visual buffer. For example, in the scout domain, visual generation creates the depictions of the imagined enemy vehicles and the agent's imagined view illustrated in Figs. 3D and 4B.[6] More than one depiction may be created, which visual recognition processes (discussed next) may use for predicate extraction.

4.5. Visual recognition

Although our theory specifies that implicit visual recognition processing occurs continuously during perception, in implementation the focus has been on visual recognition during imagery. Supporting visual recognition in general is a major challenge, but allowing simple recognition processes in certain domains can be useful. After visual generation, the agent explicitly ''recognizes'' the perceived or imagined objects in the depiction via perceptual pointers (e.g., in Fig. 3D the agent knows the identity of the two imagined enemy vehicles, the two imagined friendly icons, the perceived enemy vehicle, and the perceived map because those are the objects it generated or is currently perceiving). Further processing is required to recognize implicitly any resulting composite object(s).

The visual system in SVS enables visual recognition either with mathematical manipulations (e.g., edge detectors, Hough transforms) or with manipulations that leverage properties of a depiction such as its topological structure and explicit representation of space. The implementation technique (not a theoretical commitment) we use for depictive manipulations are pixel-level rewrite rules that encode pixel transformations required to create new depictions (Furnas, Qu, Shrivastava, & Peters, 2000; Lathrop & Laird, 2009). As a simple example, Fig. 9A illustrates two pixel-level rewrite rules. The top rule states, ''if there is a black pixel adjacent to a gray pixel then change the gray pixel to white.'' Similarly, the bottom rule states, ''if there is a black pixel diagonal to a gray pixel then change the gray pixel to white.'' When a rule matches a region of the depiction, the right-hand side action rewrites

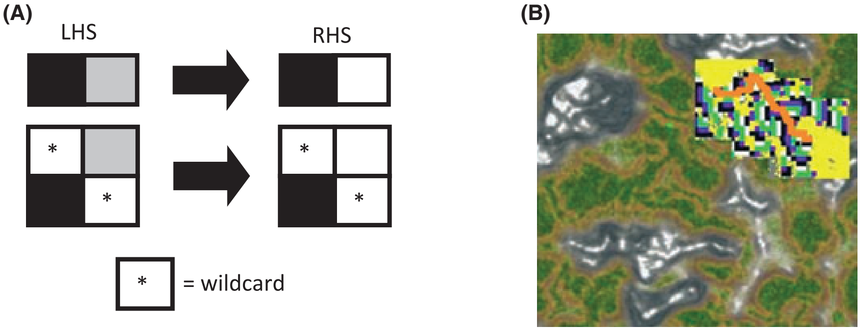


Fig. 9. Pixel rewrites. (A) An example of two-pixel rewrite rules for manipulating a bitmap. (B) The output of pixel-level rewrite rules to detect a likely path through terrain. There are two sets of rules used to produce the path: (1) a set to produce a ''distance field flood'' in the region of interest, and (2) a set that uses the resulting distance field flood to create a path from a source (enemy location) to a destination (enemy hypothesized goal location).

the appropriate pixel(s). The asterisk represents wildcard values and a pixel rewrite rule may specify rotations (90, 180, 270°) to search for a match.

To control visual recognition, symbolic structures in Soar specify the pixel-rewrite rules, depiction(s), and the order in which a set of pixel-rewrite rules or mathematical manipulations are processed. For pixel-rewrite rules, each rule has a priority associated with it. SVS processing iterates over the depiction while there are rules that still match the depiction(s).[7] With appropriate rules, the result of this process can be meaningful depictions, such as an outline of the enclosed space in an object (Lathrop & Laird, 2007) or a likely path on a map (Fig. 9B, Lathrop & Laird, 2009). As these processes derive meaningful visual information from undifferentiated pixels, we consider them *recognition* processes. Visual recognition processing is typically followed by predicate extraction. For example, in the Scout domain, predicate extraction determines the portion of the path that overlaps the agent's view (Fig. 4B).

5. Agents in spatial and visual domains

In our experiments in the Scout domain, we found that an agent using imagery provides more information concerning the enemy than a comparable agent without mental imagery (Lathrop, 2008; Lathrop & Laird, 2009). The SVS agent in this domain uses the comprehensive capabilities of Soar/SVS, especially the capabilities of the visual system to manipulate the depictive representations, recognize topology (e.g., what portion of a path is covered by the agent's view), and measure distance (what is the length of the observed portion of the path). The agent with mental imagery provides more information to reason from as its spatial and visual reasoning is more accurate than reasoning strictly with abstract symbols. The computational advantage frees resources to perform other cognitive functions (e.g., observe, send, and receive reports), and using Soar for control allows the agent to maintain reactivity (e.g., interrupting mental imagery when new perceptual information arrives).

In the pegged blocks world domain, our experiments have focused on providing evidence that the capabilities provided by imagery can compensate for problems that arise when an agent's perception system is unable to capture all relevant properties of the problem state necessary to choose an action. Given such a perception system, an agent is able to achieve better performance with imagery than without it (Wintermute & Laird, 2009).

These agents use the spatial system of SVS. Soar requests primitive information about the scene via predicate extraction, which it symbolically composes to form an abstract state. Actions are imagined by using predicate projection, and the next state is inferred using the same abstraction technique. For example, in Fig. 6, the agent has inferred that a collision would occur. Given this information, the agent can decide on an action. While this domain is simple, approaching it from a functional standpoint in a comprehensive architecture allows us to investigate a previously underexplored aspect of imagery. In contrast to prior ''imagery debate'' arguments, which often pit perceptual and abstract representations against each other, benefits here are gained through the simultaneous representation of information in both systems.

6. Discussion

With respect to other work in modeling visuospatial cognition, our work is distinguished by three related foci: A focus on functionality rather than modeling fidelity, a focus on comprehensive architecture rather than isolated processes, and a focus on implementation rather than theory. Of course, we do not strictly abide by these foci: We do aim for model fidelity at some level, we address some areas of visuospatial cognition more than others, and there are unimplemented aspects of our theory. However, due to the degree that we follow these foci, our research provides an interesting perspective from which to examine several aspects of the field. Here, we will briefly discuss these perspectives, roughly organized by research areas.

6.1. Functional analyses of imagery-like processing

In recent years, mental imagery research has largely been empirical with focus shifting from behavioral evidence to neurologic evidence of the brain structures activated during mental imagery (Kosslyn et al., 2006; Pylyshyn, 2002). In the meantime, the core theoretical arguments for the functionality of imagery have remained largely unchanged. These arguments have tended to consider the benefits of particular representational formats independently of a fully defined cognitive architecture. These benefits are present in our system. Depictive productions manipulating the visual buffer are able to leverage the explicit encoding of space for increased efficiency. Similarly, location, size, and orientation are explicit in the spatial scene, enabling efficient extraction of related properties. Efficiency benefits of this sort have been examined in many previous systems (e.g., Funt, 1976; Gelernter, 1959; Glasgow & Papadias, 1992; Larkin & Simon, 1987; Tabachneck-Schijf, Leonardo, & Simon, 1997), but not within the context of a cognitive architecture.

Larkin and Simon's (1987) work has been particularly influential in this area, and, as mentioned above, our system demonstrates the same efficiency benefits. However, our architecture differs in several ways. Whereas the evaluation of their arguments were realized in an entirely symbolic medium, we take their arguments one step further by taking the representations literally and introducing the spatial and depictive representations constrained by cognitive architecture. In addition, where Larkin and Simon correctly point out that diagrammatic and sentential representations can, in theory, represent the same information, once a comprehensive architecture is fleshed out, this equivalence may not be present.

In our system, the visual buffer, spatial scene, and symbolic working memory are individually able to represent information that the constraints of the architecture do not allow to be transformed into other representational formats. This inequivalency provides a different functional argument for the use of imagery. In the pegged blocks world, the primary reason that it is beneficial to imagine a block in a particular position (Fig. 6) is not efficiency, but rather that the constraints of the architecture admit no other way of determining whether a collision would result. Similarly, in the scout domain, acquiring the overlap of a scout's view of an irregular path is constrained to the visual buffer. While these results are partly due to the particularities of our architecture, we are working toward related general principles.

Regardless of implementation details, any architecture employing abstract symbolic representations might demonstrate a similar need for imagery (Wintermute & Laird, 2009).

6.2. Theories of grounded cognition

Our architecture also provides perspective on ''grounded'' or ''embodied'' cognitive theories (Barsalou, 2008). The system retains and uses perceptual-level information during cognition, with the perception and action systems internally used in this process. In that way, the system bears a resemblance to theories such as Barsalou's (1999) proposal for a perceptual symbol system and Grush's (2004) emulation theory of representation. From this point of view, the contents of Soar's working memory can be considered as the symbolic *aspects* of the underlying representation in the memories of SVS. While grounded theories have often been pitched as alternatives to existing symbolic theories, our implementation has shown that the existing symbolic Soar architecture is compatible with nonsymbolic processing in SVS. This is because Soar is largely a theory of high-level decision making, complementary to the processing in SVS, which identifies properties and makes inferences but does not choose actions or control high-level processing.

6.3. Mental models

A large proportion of prior research in spatial and visual cognition has been pursued under the broader theory of mental models (Johnson-Laird, 1989; Ragni & Steffenhagen, 2007; Zwaan & Radvansky, 1998). In mental model theory, the model is a concrete, situated representation where the reasoning is not based on formal rules of inference but rather proceeds by comparing the imagined situation with predicates (''Does block A intersect block B?,'' ''Does the path overlap the view?''). Our architecture's imagery system can be viewed as a realized form of mental models. The spatial and depictive representations in SVS are the instantiation of a particular situation where the representations may have a combination of perceived and imagined objects. SVS helps explain the details of one theory of how generation of these situated representations occur and how the resulting predicates are extracted. The symbolic representations in Soar control the high-level processing, directing SVS as to what objects should be placed in the imagined model through perceptual pointers. The desired predicates are queried based on task knowledge so that the system focuses on the task and its corresponding goals. These details provide clarification as to how mental models are formed and inspected.

6.4. Egocentric/allocentric distinctions

A common issue discussed in the psychological literature is the distinction between ego- and allocentric spatial representations (Klatzky, 1998). Strictly speaking, the spatial scene is an allocentric representation, as it encodes absolute positions in space, not relative to the agent's location. As we are not aiming for precise model fidelity, however, this distinction has not turned out to be important in our system, as egocentric information can be easily calculated from the representation. The behavior of the agent is unaffected by the details of the underlying spatial encoding, since decisions are made based on qualitative properties of objects relative to one another, properties that are preserved across translation, scaling, and rotation. However, if the architecture were extended to precisely model humans in tasks such as navigation through an environment, this distinction might play a more prominent role.

A related issue is encountered in the Scout domain where the agent may be observing an approaching enemy vehicle (Fig. 3A) and then decide to look at its map (Fig. 3D) to further analyze the situation. Since the agent perceives the map, the map's depiction is already in the visual buffer. The agent must determine where on the map it is located, along with any enemies it can see. This presents a problem where conversion from ego- to allocentric information is needed. The Scout agent uses task knowledge for this, but it is an open question what (if any) additional architectural mechanisms are necessary to support this form of transformation in general.

6.5. Computational approaches

Previous unified computational theories of spatial and visual cognition exist, but many focus on reasoning with abstract symbolic representations (Baylor, 1971; Carpenter, Just, & Shell, 1990; Lyon, Gunzelmann, & Gluck, 2008; Moran, 1973). Such theories normally assume perception is a transducer, where its primary purpose is to transform sensory data into abstract symbolic representations that capture the relevant properties of salient objects. This basic scheme is shared in many AI designs, which reason solely with symbolic representations while disregarding perceptual-level representations. In contrast, we have found that by maintaining perceptual representations and manipulating them through imagery to be extremely beneficial from a functional standpoint, making them essential parts of the theory.

A few architectures have moved toward perceptual-level representations. Two such extensions to the ACT-R architecture have been proposed (Gunzelmann & Lyon, 2007; Harrison & Schunn, 2002). Reflecting differing research goals, however, the designs have focused more on high-fidelity modeling, and less on broad functionality. While these systems are more precise at modeling the tasks they cover, neither addresses aspects that we find are important in our system, such as general-purpose predicate extraction and projection, motion simulation, visual generation, and visual recognition. Unifying these approaches into a comprehensive, implemented architecture capable of precise modeling is certainly a fertile direction for future research.

Other systems integrate symbolic reasoning with spatial and visual representations (Barkowsky, 2007; Glasgow & Papadias, 1992; Tabachneck-Schijf et al., 1997). These architectures have been examined chiefly in terms of particular problem domains, such as geographic reasoning (Barkowsky, 2007), molecular scene analysis (Glasgow & Papadias, 1992), or modeling supply and demand (Tabachneck-Schijf et al., 1997), so it is difficult to compare these architecture's generality with SVS, which has been used across multiple domains. Other comprehensive theories for this integration have also been put forth (Barsalou, 2008; Grush, 2004) but not in the form of implemented architectures.

The closest parallel to our work is that of Kurup and Chandrasekaran (2006), which combines the Soar cognitive architecture with a diagrammatic reasoning system. They, like us, are motivated by how a task-independent architecture gains functionality by combining symbolic and perceptual representations during reasoning. Although their work focuses on diagrammatic reasoning tasks, overall they strive for general implementation and functionality. There are a few key theoretical differences between our approaches. First, their theory constrains the type of imaginable objects to points, curves, and regions while we leave the type of imaginable object open-ended to any object experienced through perception, imagined by composing known objects, or emerging from imagery processing. Second, we are committed to the three representations (symbolic, spatial, and visual depictive), while Kurup and Chandrasekaran are noncommittal as to the form of diagrammatic representations. Their previous implementations use sentential, mathematical representations and manipulations. We make a clear distinction between the spatial and visual representations, as there are different types of reasoning that can be performed with each.

In summary, our core theoretical commitments that, taken together, distinguish this work from previous research are the following:

- The use of three representational formats: abstract symbolic, quantitative spatial, and visual depictive
- Simultaneous representation of abstract and concrete information across the three short-term memories mapped via perceptual pointers
- A perceptual long-term memory for the encoding of spatial objects and transformations, visual textures, and motion patterns
- Use of abstract symbols for control, decision making, focus of processing, and modification of spatial and visual representations
- A fixed set of predicate extraction processes for topology, direction, and distance driven by top-down, symbolic queries
- Imagery processing through predicate projection, memory retrieval, and motion simulation, driven by top-down, symbolic commands
- Use of visual recognition processes that take advantage of the explicit representation of space
- Integration with the constraints of a comprehensive cognitive architecture

7. Conclusion

We have presented a comprehensive implementation of a general-purpose cognitive architecture for spatial and visual reasoning. In designing this system, we are motivated by psychological findings, specifically in mental imagery. We focused on the functionality gains that such a system provides, rather than precise human modeling. Our approach is grounded within an established symbolic cognitive architecture, Soar. The resulting architecture has been used to create comprehensive agents to solve complex tasks, such as in the

Scout domain (Lathrop & Laird, 2009), as well as to explore basic principles of visuospatial cognition, such as the use of imagery to recognize visual features (Lathrop & Laird, 2007), the use of imagery-based predictions as a means to improve problem state representation (Wintermute, 2010; Wintermute & Laird, 2009) and the connection between cognition and action (Wintermute, 2009a). Our approach allows the system to achieve the accuracy and efficiency inherent with perceptual reasoning, while allowing central cognitive processes to retain control. Processing emerges from the fluid combination of different representations, leveraging the particular efficiencies and capabilities of each.

Notes

1. SVS is a component of Soar; however, for simplicity in this article, we will use ‘‘Soar’’ to refer to the previously existing symbolic components of the architecture.
2. We are using the term ‘‘orientation’’ to refer to precise information about which way an object is facing (e.g., the front is toward <1.45,0.2,0>), and the term ‘‘direction’’ to refer to qualitative information about how objects relate: ‘‘the tree is to the left of the house,’’ ‘‘the house is North of the city,’’ etc.
3. The Scout agent is implemented in a predecessor to the version of the architecture presented here. However, the general architecture remains the same, with only minor differences in the capabilities used.
4. Further details of SVS, focusing on spatial processes, can be found in Wintermute (2009b).
5. These are a subset of the RCC-8 Topological Relationships (Cohn, Bennett, Gooday, & Gotts, 1997). Some of the RCC-8 relationships were not implemented in SVS or its predecessors.
6. In Fig. 4B, the agent’s imagined view (blue) is a visually generated depiction. The path (orange) is another depiction created through visual recognition processing (Fig. 4A).
7. Rules may match more than one depiction if there are multiple depictions in the visual buffer.

References

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

Barkowsky, T. (2007). Modeling mental spatial knowledge processing: An AI perspective. In F. Mast & L. Jäncke (Eds.), *Spatial processing in navigation, imagery, and perception* (pp. 67–84). Berlin: Springer.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences*, *22*, 577–660.

Barsalou, L. W. (2008). Grounded cognition. *Annual Review of Psychology*, *59*, 617–645.

Baylor, G. W. (1971). *A treatise on the mind’s eye: An empirical investigation of visual mental imagery*. PhD thesis, Carnegie-Mellon University, Pittsburgh, PA.

Carpenter, P. A., Just, M. A., & Shell, P. (1990). What one intelligence test measures: A theoretical account of the processing in the Raven Progressive Matrices Test. *Psychological Review*, *97*(3), 404–431.

Cohn, A. G., Bennett, B., Gooday, J., & Gotts, N. M. (1997). Qualitative spatial representation and reasoning with the region connection calculus. *GeoInformatica*, *1*(3), 275–316.

Funt, B. V. (1976).*WHISPER: A computer implementation using analogues in reasoning*, PhD Thesis, University of British Columbia, Vancouver, BC. Available from ProQuest Dissertations and Theses database (UMI No. 28685)

Furnas, G., Qu, Y., Shrivastava, S., & Peters, G. (2000). The use of intermediate graphical constructions in problem solving with dynamic, pixel-level diagrams. In M. Anderson, P. Cheng, & V. Haarslev (Eds.), *First international conference on the Theory and Application of Diagrams: Diagrams 2000* (vol. 1889, pp. 314–329). Berlin: Springer.

Gelernter, H. (1959). *Realization of a geometry theorem-proving machine*. Paper presented at the International Conference on Information Processing, Paris: Unesco.

Glasgow, J., & Papadias, D. (1992). Compuational imagery. *Cognitive Science*, *16*, 355–394.

Grush, R. (2004). The emulation theory of representation: Motor control, imagery, and perception. *Behavioral and Brain Sciences*, *27*(03), 377–396.

Gunzelmann, G., & Lyon, D. R. (2007). Cognitive architectures: Valid control mechanisms for spatial information processing. In H. Schultheis, T. Barkowsky, B. Kuipers, & B. Hommel (Eds.), *Technical report #SS07-01: AAAI spring symposium series: Control mechanisms for spatial knowledge processing in Cognitive/Intelligent systems* (pp. 23–28). Menlo Park, CA: AAAI Press.

Harrison, A. M., & Schunn, C. D. (2002). ACT-R/S: A computational and neurologically inspired model of spatial reasoning. In D. Freudenthal, J. M. Pine, F. Gobet, W. D. Gray, & C. D. Schunn (Eds.), *Proceedings of the twenty-fourth annual meeting of the Cognitive Science Society* (pp. 1008). Mahwah, NJ: Erlbaum.

Helstrup, T. (1988). Imagery as a cognitive strategy. In M. Denis, J. Engelkamp, & J. T. E. Richardson (Eds.), *Cognitive and neuropsychological approaches to mental imagery* (pp. 241–250). Dordecht/Boston/Lanchester: Martinus Nijhorff.

Hernández, D. (1994). *Qualitative representation of spatial knowledge*. Lecture Notes in Artificial Intelligence (Vol. 804). Berlin: Springer-Verlag.

Johnson-Laird, P. N. (1989). *Mental models: Towards a cognitive science of language, inference and consciousness*. Cambridge, MA: Harvard University Press.

Klatzky, R. L. (1998). Allocentric and egocentric spatial representations: Definitions, distinctions, and interconnections. In C. Freska, C. Habel, & K. Wender (Eds.), *Spatial cognition*, Lecture Notes in Computer Science (Vol. 1404, pp. 1–18). Heidelburg: Springer-Verlag.

Kosslyn, S. M. (1994). *Image and brain – the resolution of the imagery debate*. Cambridge, MA: MIT Press.

Kosslyn, S. M., Thompson, W. L., & Ganis, G. (2006). *The case for mental imagery*. New York: Oxford University Press.

Kurup, U., & Chandrasekaran, B. (2006). Multi-modal cognitive architectures: A partial solution to the frame problem. In R. Sun & N. Miyake (Eds.), *28th annual conference of the Cognitive Science Society* (pp. 1646–1651). Vancouver, BC: Cognitive Science Society, Inc.

Laird, J. E. (2008). Extending the Soar cognitive architecture. In P. Wang, B. Goertzel, & S. Franklin (Eds.), *Proceedings of the first conference on Artificial General Intelligence* (pp. 224–235). Amsterdam: IOS Press.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, *33*(3), 1–64.

Langley, P., Laird, J., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, *10*, 141–160.

Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, *11*, 65–99.

Lathrop, S. D. (2008). *Extending cognitive architectures with spatial and visual imagery mechanisms*. PhD thesis, University of Michigan.

Lathrop, S. D., & Laird, J. E. (2007). Towards incorporating visual imagery into a cognitive architecture. In R. L. Lewis, T. A. Polk, & J. E. Laird (Eds.), *Proceedings of the eighth international conference on Cognitive Modeling* (pp. 25–30). London: Taylor & Francis/Psychology Press.

Lathrop, S. D., & Laird, J. E. (2009). Extending cognitive architectures with mental imagery. In B. Goetzel, P. Hitzler, & M. Hutter (Eds.), *Proceedings of the second conference on Artificial General Intelligence (AGI-09)* (pp. 97–102). Amsterdam: Atlantis Press.

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

Moran, T. P. (1973). *The symbolic imagery hypothesis: An empirical investigation via a production system simulation of human behavior in a visualization task*. PhD thesis, Carnegie-Mellon University, Pittsburgh, PA.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Norman, J. (2000). Differentiating diagrams: A new approach. In M. Anderson, P. Cheng & V. Haarslev (Eds.), *Theory and application of diagrams* (Vol. 1889, pp. 105–116). Berlin: Heidelberg Springer-Verlag.

Pylyshyn, Z. (1973). What the mind's eye tells the mind's brain: A critique of mental imagery. *Psychological Bulletin*, *80*, 1–24.

Pylyshyn, Z. (1981). The imagery debate: Analogue media versus tacit knowledge. *Psychological Review*, *88*, 16–45.

Pylyshyn, Z. W. (2001). Visual indexes, preconceptual objects, and situated vision. *Cognition*, *80*(1–2), 127–158.

Pylyshyn, Z. (2002). Mental imagery: In search of a theory. *Behavioral and Brain Sciences*, *25*, 157–238.

Ragni, M., & Steffenhagen, F. (2007). Qualitative spatial reasoning: A cognitive and computational approach. In D. S. McNamara & J. G. Trafton (Eds.), *Proceedings of the 29th annual conference of the Cognitive Science Society* (pp. 1415–1420). Austin, TX: Cognitive Science Society.

Tabachneck-Schijf, H. J. M., Leonardo, A. M., & Simon, H. A. (1997). CaMeRa: A computational model of multiple representations. *Cognitive Science*, *21*(3), 305–350.

Wintermute, S. (2009a). *Integrating action and reasoning through simulation*. In B. Goetzel, P. Hitzler, & M. Hutter (Eds.), *Proceedings of the second conference on Artificial General Intelligence (AGI-09)* (pp. 192–197). Amsterdam: Atlantis Press.

Wintermute, S. (2009b). *An overview of spatial processing in Soar/SVS* (Technical Report No. CCA-TR-2009-01). Ann Arbor: University of Michigan Center for Cognitive Architecture.

Wintermute, S. (2010). Using imagery to simplify perceptual abstraction in reinforcement learning agents. In M. Fox & D. Poole (Eds.), *Proceedings of the twenty-fourth AAAI conference on Artificial Intelligence (AAAI-10* (pp. 1567–1573). Menlo Park, CA: AAAI Press.

Wintermute, S., & Laird, J. E. (2007). Predicate projection in a bimodal spatial reasoning system. In R. C. Holte & A. Howe (Eds.), *Proceedings of the twenty-second AAAI conference on Artificial Intelligence (AAAI-07)* (pp. 1572–1577). Menlo Park, CA: AAAI Press.

Wintermute, S., & Laird, J. E. (2008). Bimodal spatial reasoning with continuous motion. In D. Fox & C. P. Gomes (Eds.), *Proceedings of the twenty-third AAAI conference on Artificial Intelligence (AAAI-08)* (pp. 1331–1337). Menlo Park, CA: AAAI Press.

Wintermute, S., & Laird, J. E. (2009). Imagery as compensation for an imperfect abstract problem representation. In N. Taatgen & H. van Rijn (Eds.), *Proceedings of the 31st annual conference of the Cognitive Science Society (CogSci-09)* (pp. 631–636). Austin, TX: Cognitive Science Society.

Zwaan, R., & Radvansky, G. (1998). Situation models in language comprehension and memory. *Psychological Bulletin*, *123*, 162–185.