TOPICS
TOPICS IN COGNITIVE SCIENCE

Topics in Cognitive Science 3 (2011) 648–666

ISSN: 1756-8757 print / 1756-8765 online
DOI: 10.1111/j.1756-8765.2011.01149.x

CogSketch: Sketch Understanding for Cognitive Science Research and for Education

Kenneth Forbus, Jeffrey Usher, Andrew Lovett, Kate Lockwood, Jon Wetzel

Qualitative Reasoning Group, Northwestern University

Received 7 December 2009; received in revised form 23 April 2010; accepted 6 July 2010

Abstract

Sketching is a powerful means of working out and communicating ideas. Sketch understanding involves a combination of visual, spatial, and conceptual knowledge and reasoning, which makes it both challenging to model and potentially illuminating for cognitive science. This paper describes CogSketch, an ongoing effort of the NSF-funded Spatial Intelligence and Learning Center, which is being developed both as a research instrument for cognitive science and as a platform for sketch-based educational software. We describe the idea of *open-domain sketch understanding*, the scientific hypotheses underlying CogSketch, and provide an overview of the models it employs, illustrated by simulation studies and ongoing experiments in creating sketch-based educational software.

Keywords: Sketch understanding; Analogy; Qualitative reasoning; Visual reasoning; Spatial reasoning; Spatial cognition; Cognitive simulation

1. Introduction

Sketching enables people to externalize and communicate ideas. People draw maps, the structure of complex systems, and sequences of sketches illustrating how a process unfolds. The power of sketching is such that visual languages are invented to depict otherwise abstract ideas (e.g., electronic circuit schematics, software modeling diagrams, parse trees). Sketching is fascinating scientifically because it engages visual, spatial, and conceptual knowledge and skills. Consequently, understanding how people understand and communicate with sketches should provide important insights for understanding human cognition more generally. Moreover, if we can use models of sketch understanding to create software that can participate in sketching in human-like ways, there are potentially significant practical benefits. Consider, for example, intelligent tutoring systems (ITSs), an application of

Correspondence should be sent to Kenneth Forbus, Qualitative Reasoning Group, Northwestern University, 2133 Sheridan Road, Evanston, IL, 60208. E-mail: forbus@northwestern.edu

cognitive science that has had substantial beneficial impact on education and training already. Almost no ITSs exist for spatial topics. This means that the benefits of ITSs and intelligent learning environments are unavailable for highly spatial disciplines, which includes many aspects of science, mathematics, and engineering. Specific examples include understanding geological formations and understanding how a mechanical design is supposed to work. Consequently, in the Spatial Intelligence and Learning Center, we are creating a sketch understanding system, *CogSketch*, which is being used to explore spatial cognition and learning.

We have two interdependent goals in developing CogSketch. First, it is a new research instrument for cognitive scientists, a tool for running simulation experiments and for gathering and analyzing behavioral data. Second, it is a platform for sketch-based educational software. The first goal facilitates the second: Simulation experiments are used to improve the range of the system's spatial reasoning abilities and cognitive fidelity, and laboratory experiments provide experience in making simplified and robust interfaces, both of which improve it as a platform for educational software. The second goal also facilitates the first: The range of knowledge and skills involved in spatial understanding and learning is huge, so we pick our research topics in part by what problems arise in creating educational software. Our vision is that within 7 years, sketch-based intelligent educational software will be as widely available to learners as graphing calculators are today. This vision can only be achieved by artificial intelligence scientists, psychologists, and learning scientists working closely together.

We begin by outlining open-domain sketch understanding and the hypotheses that underlie CogSketch. Section 3 outlines how CogSketch works. Section 4 summarizes research using CogSketch in cognitive simulation experiments, and Section 5 describes education experiments. We close by discussing other related work and future plans.

2. Our hypotheses

Most sketch understanding systems treat understanding as a matter of recognizing ink, or ink plus speech, as a member of a limited number of predefined symbols [e.g., military symbols (Pittman, Smith, Cohen, Oviatt, & Yang, 1996), electronics/UML diagrams (Alvarado, Oltmans, & Davis, 2002), force diagrams (Lee et al., 2007)]. This limits them to expressing a small, fixed set of concepts. Recognition-based interfaces can be of great practical value—handwriting recognition is the most successful example—but our goal is fundamentally different. A key insight is that in human-to-human sketching, recognition is a catalyst, not a requirement (Forbus, Ferguson, & Usher, 2001; Landay et al., 2002). When people sketch with each other, we typically also talk, using language to label the intended meaning of pieces of ink, or of the spaces defined by the ink. There are several reasons for this: Most people are not artists, and most spatial concepts do not have standardized, easily recognizable symbols. Moreover, in many domains, the mapping from shapes to concepts is one to many: In a sketch describing the layers of the earth, for example, the core, mantle, and crust are all drawn as circles. Consequently, CogSketch provides several ways for people to conceptually label their ink as they draw, so that recognition is not required. This opens up

CogSketch to sketching anything for which it has conceptual knowledge. This is what we mean by *open-domain sketch understanding*.

We want CogSketch to model the perceptual, spatial, and conceptual understanding that people bring to sketching. Our key hypotheses are as follows:

> Hypothesis: *Perceptual processing produces qualitative spatial representations*. Qualitative representations quantize continuous properties, making meaningful units that can be manipulated symbolically (Forbus, 2007). Constructing representations of segments, regions, volumes, and relationships between and within them is, we argue, one of the key functions of perception. Qualitative spatial representations grounded in metric representations, such as the Metric Diagram/Place Vocabulary model (Forbus, 1983), have been used to create human-like performance in a variety of tasks, including reasoning about motion through space (Forbus, 1983), reasoning about mechanical systems (Forbus, Nielsen, & Faltings, 1991; Joscowicz & Sacks, 1991), recognizing patterns in weather data (Huang & Zhao, 2000), and reasoning about complex dynamical systems (Bradley, 1995; Yip, 1991). AI scientists are not alone in this view: In cognitive psychology, a roughly equivalent but independently developed distinction is the coordinate/categorical approach to spatial representation (Huttenlocher, Hedges, & Duncan, 1991; Kosslyn, Chabris, Marsolek, & Koenig, 1992). The psychological work provides the strongest direct evidence for a combination of qualitative and metric representations in human spatial reasoning, while the AI work provides the strongest direct evidence that these ideas scale to capture the range of human expertise in spatial thinking.
>
> Hypothesis: *Structure-mapping processes are used in visual reasoning*. Gentner's (1983) structure-mapping theory defines analogy and similarity in terms of comparison of structured representations. These structured representations are symbolic descriptions encoding entities, their attributes, relations between them, and relations between relationships (e.g., encoding causality and constraint). There is psychological evidence (Lovett, Gentner, Forbus, & Sagi, 2009a) that structure-mapping computations are used in visual processing. Qualitative representations provide visual and spatial structure[1] which is used in analogical operations of matching, retrieval, and generalization. These analogical operations are used to identify similarities and differences, and form components in models of larger-scale cognitive processing. For example, analogical matching and retrieval have been used to model causal reasoning in sketches (Klenk, Forbus, Tomai, Kim, & Kyckelhahn, 2005) and recognizing potential visual/conceptual relationships (Forbus, Usher, & Tomai, 2005).

For analogical processing, we use simulations of structure-mapping theory: SME (Falkenhainer, Forbus, & Gentner, 1989; Forbus & Oblinger, 1990) models matching, MAC/FAC (Forbus, Gentner, & Law, 1995) models retrieval, and SEQL (Halstead & Forbus, 2005; Kuehne, Forbus, Gentner, & Quinn, 2000) models generalization.[2] These simulations are described elsewhere and we omit further description of them for brevity. For a model of conceptual knowledge, we use the contents of the OpenCyc knowledge base,[3] plus our extensions. Our extensions include defining concepts and relationships for qualitative reasoning, visual processing, and analogical processing. The first two are summarized below; the analogy ontology is described in Forbus, Mostek, and Ferguson (2002). Everything else (e.g., concepts of everyday objects like wheelbarrows, windows, and raindrops) and relationships between them (e.g., types of physical connections, ownership) comes from the knowledge base (KB). The OpenCyc KB is quite broad, including over 58,000 concepts, 8,000 relations, and 1.3 million facts. Its developers used ideas from the cognitive science literature whenever possible, but of course there is no way to test the detailed fidelity of any large-scale knowledge base, and it is not clear what that would mean, given individual, cultural, and development differences between individuals. While far from perfect, we have found it sufficient for our purposes. Our models of visual representation and processing are outlined in Section 3.2.

3. How CogSketch works

CogSketch combines its visual, spatial, and conceptual knowledge about the elements in a sketch to create a qualitative, symbolic representation both of the sketch and of what it depicts. Analogical processing is used in the representation construction process, as well as in tasks using these representations. This section outlines how this is achieved.

3.1. Core concepts

3.1.1. Glyphs

In CogSketch, every user-drawn object in a sketch is a *glyph*. Each glyph has *ink* and *content*. The ink consists of one or more polylines, which are lists of points representing what was drawn. The content is a symbolic token that represents what the glyph denotes. Visual relationships are computed over glyphs, and depending on the semantics of the sketch (genre & pose, described below), they can lead to inferences about the spatial relationships between the content of those glyphs.

CogSketch relies on the user to segment their ink into glyphs and to label them with concepts from the knowledge base. There are several interface mechanics used to accomplish this, depending on circumstances, which are described elsewhere (Forbus, Usher, & Chapman, 2003). The most flexible allows the user to type in the name of a concept or relationship from the underlying knowledge base, which has the drawback of requiring technical sophistication and a detailed understanding of the OpenCyc ontology. The friendlier, but more limited, methods use either a short list of concepts (with natural language strings hiding the ontological details) or palettes of concepts depicted graphically when such conventions are known to a community (e.g., military task symbols). These interfaces have been used in a variety of experiments with students and military personnel (Barker et al., 2003; Rasch, Kott, & Forbus, 2002), suggesting that they are sufficiently usable. To support gathering data in certain types of experiments, CogSketch also includes a mode where unrestricted natural language strings can be used as conceptual labels.

Glyphs can be composed hierarchically by grouping. For example, the parts of a wheelbarrow can be individually drawn and labeled as glyphs, and then combined into a

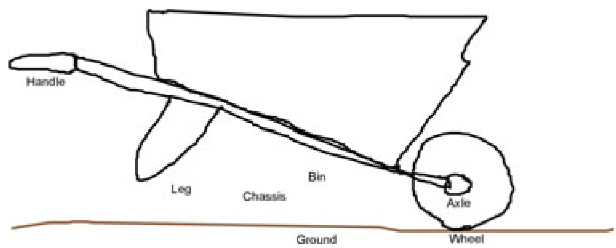

Fig. 1. A sketch of a wheelbarrow. Each part is drawn as a glyph.

higher-order glyph, with its own conceptual label, that includes the other glyphs as parts (see Fig. 1). There currently is no interface mechanism for decomposing complex glyphs into parts, although CogSketch's visual system has facilities for doing this automatically on demand, as described below.

Relationships between objects can be indicated by *relation glyphs*, which are drawn as arrows. The conceptual labels for relation glyphs are restricted to binary relations. *Annotation glyphs* provide modifiers to glyphs. For example, the radius of a circle or the height of a building can be indicated via a length annotation. Annotations are also used to express some physical relationships, for example, applied force or direction of motion.

3.1.2. Subsketches, metalayer, and layers

Sketches have structure. Complex sketches often consist of multiple subsketches. For example, in describing a building, one might have a subsketch that shows how it looks from the street, another subsketch representing its floor plan, and a third subsketch that is a schematic of part of its electrical system. A sketch in CogSketch consists of one or more subsketches. To express relationships between subsketches, a special interface called the *metalayer* is used. On the metalayer, each subsketch is treated as if it were a glyph. Arrows can be drawn between these glyphs to describe relationships between subsketches. This can be used to describe sequences of states in a complex behavior (e.g., Fig. 2), and to represent distinct possible outcomes via *comic graphs* (Forbus et al., 2003).

In CogSketch, a subsketch consists of one or more *layers*. Drawing programs commonly use layers to organize multiple parts of a drawing, a metaphor for using clear acetate sheets over paper when making complex sketches. Normal layers allow inking, and a special bitmap layer allows users to specify a bitmap that can be drawn over by other layers. Layers share the same coordinate system, but many default CogSketch operations are only done between glyphs on the same layer. Each layer has a *genre* and *pose*, which help CogSketch construct appropriate spatial relationships for the contents from visual relationships between the glyphs. Genre concerns the intended spatial interpretation of the visual entities in the sketch, whose geometry is further modified by pose. For example, in the Abstract-View genre, the visual relationships between the glyphs (left/right, above/below) provides no information about spatial relationships between their contents (e.g., electronic components in a schematic). For the Physical-View and Geospatial-View genres, the relationship between visual and spatial relationships also depends on the pose. For example, if glyph A

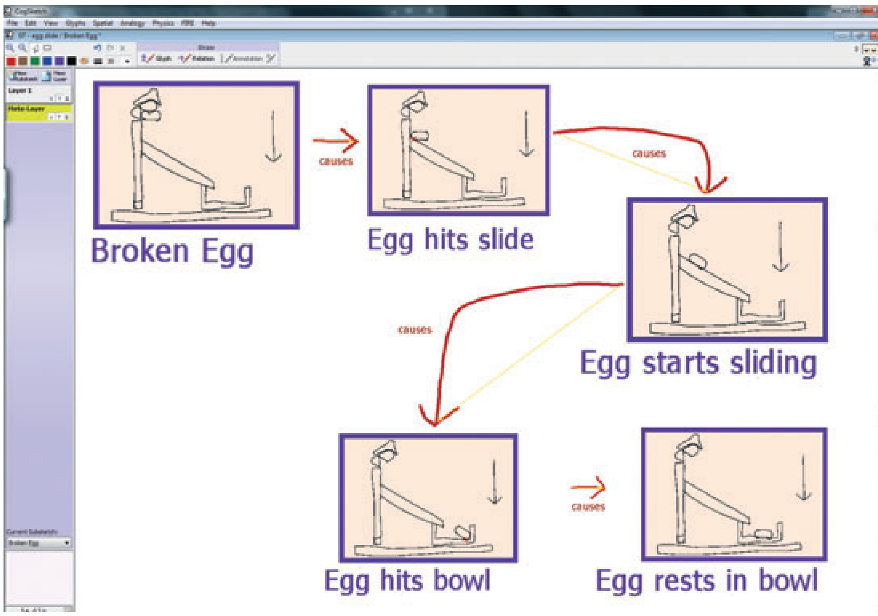

Fig. 2. The metalayer can be used to describe complex processes, such as this sequence of states in a student's design for a machine that lets a one-handed person easily crack an egg. Each step of this sequence is a sub-sketch.

is above glyph B and the genre is physical and the pose is side-view, then A is *above* B. But if the genre is Geospatial-View and the pose is Looking-from-top, then A is *north of* B.

3.2. Visual processing

Wherever possible, we have used psychological evidence to constrain the visual representations and computations CogSketch computes. Improving the fidelity of its visual processing is an ongoing project.

3.2.1. Ink processing

CogSketch automatically computes a number of qualitative visual relations and attributes for glyphs in a layer. These represent general visual features of the sketch, and so they do not make use of any task- or domain-specific knowledge about the objects being sketched. For example, a glyph's size is based on the area of its bounding box. A symbolic description of size, ranging from tiny to huge, is computed by comparing this area to the overall size of the sketch.

CogSketch computes the RCC-8 qualitative relations (Cohn, 1996) that describe all possible topological relations between two 2D shapes (e.g., *disconnected*, *edge-connected*, *partially overlapping*).[4] RCC-8 relations are used to guide the generation of other relations. These include positional relations (e.g., above/below, left/right) and containment.

Positional relations are only computed by default between adjacent glyphs, the intuition being that the network of visual relationships we compute should respect the neighborhood structure of the sketch. CogSketch calculates adjacency via Voronoi diagrams (Edwards & Moulin, 1998). There are conditions for which Voronoi adjacency may not accurately reflect psychological judgments of locality, but empirically it has been sufficient for our purposes. There are four *visual* positional relations that are computed between adjacent glyphs: rightOf, above, enclosesHorizontally, and enclosesVertically. These visual relations are used along with a layer's genre and pose to generate spatial positional relations between the content of the glyphs, as noted above.

CogSketch also uses RCC-8 relations to identify two types of glyph groups in a sketch: *connected glyph groups* and *contained glyph groups*. A connected glyph group consists of a set of glyphs whose ink strokes intersect. A contained glyph group consists of a single container glyph and the set of glyphs fully contained within it. Fig. 3 shows an example of a connected glyph group and a contained glyph group.

3.2.2. Representing shapes of glyphs

For some tasks, decomposing glyphs into their component edges in order to represent their shape is crucial. CogSketch uses two levels of representation. The *scene* level treats glyphs as entities and focuses on relationships between them. The *shape* level of representation treats the edges of a glyph as entities and focuses on relationships between them. CogSketch computes scene-level representations by default, and shape-level representations on demand. These representations are a key component for analogical processing used in visual tasks.

CogSketch includes algorithms for decomposing and merging the polylines that make up a glyph into visually meaningful edges. (For brevity we omit the details; see Lovett, Tomai, Forbus, & Usher, 2009b.) Qualitative representations are then computed for the edges of a glyph and the relationships between them. The qualitative representations include three kinds of information:

1. Attributes describing basic shape properties. For edges, these include shape (Straight/Curved/Ellipse), length relative to the longest edge in the glyph (Tiny/Short/

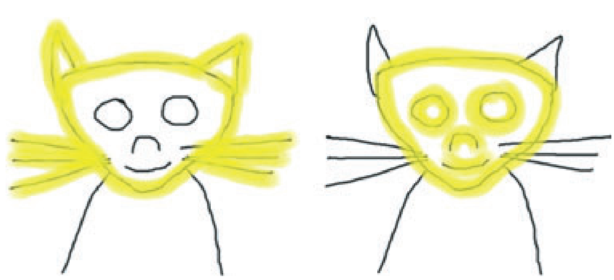

Fig. 3. Examples of glyph groups. The glyphs in a connected glyph group (left) all touch a common glyph. Here, the whiskers and ears all touch the glyph representing the cat's head. In a contained glyph group (right), a set of glyphs are inside another glyph. Here, the eyes, nose, and mouth of the cat are inside the glyph for the head.

Medium/Long), and whether they are aligned with the horizontal or vertical axis. Corners are classified as concave or convex.
2. Relations that describe whether two edges are parallel, perpendicular, connected, intersecting (i.e., connected at an X-junction), or edge-intersecting (i.e., connected at a T-junction).
3. Higher-order relations that describe relationships along contours. For example, the cycleAdjacentAngles relation is used to relate every pair of adjacent corners moving clockwise around the contour of a shape. Similarly, relationships are added to corners that are determined to be right angles or whose edges are of equal length.

These qualitative relationships help SME find mappings between two shape representations. The higher-order relationships provide local orientation-independent cues about subsets of the shape which help guide matching. The particular representation choices are motivated by both the psychological literature and our own simulation experiments. For example, we encode corners as concave/convex, as opposed to acute/right/obtuse, because this distinction is more fundamental and is known to be visually salient (Ferguson, Aminoff, & Gentner, 1996).

An important role of SME in visual processing is determining whether two shapes are the same, different, or rotated/reflected versions of each other. This process involves first identifying the bounding edges of a glyph, that is, those which touch the area outside the glyph. (If a glyph is not closed, then all edges are bounding edges.) Identifying outlines as a starting point was suggested by Hoffman and Richards (1984). SME is used to compare the representations of the bounding edges: If the mapping it produces does not contain correspondences for every edge, the shapes are marked as different. To determine possible rotation/reflection relationships, quantitative comparisons are performed for each pair of corresponding edges. Rotation requires that every pair of corresponding edges should have the same difference in their orientations. Reflection requires that every pair of corresponding edges should be reflected about the same axis. Since SME can produce multiple mappings, it is possible to find multiple possible rotations or reflections between two shapes. Our default is to prefer the smallest possible rotation, based on evidence from mental rotation tasks (Shepard & Cooper, 1982). When the rotation is zero degrees, the two shapes are taken to be identical.

3.2.3. Additional on-demand encoding

CogSketch's default processing computes a variety of relationships at the scene level, but shape representations are currently only computed on demand. When they are computed, additional scene-level attributes and relationships become possible to compute. For example, CogSketch computes equivalence classes of shapes, within which all glyphs are related via rotation or reflection, and assigns them an arbitrary attribute name. Thus, while it does not try to identify shapes as exemplifying particular known types (e.g., squares), it does recognize that shapes are the same. Within each equivalence class, one glyph is picked as a reference for size and orientation, and CogSketch assigns attributes based on whether other glyphs are reflected or rotated versions of the reference.

Also computed on demand are attributes for border and fill colors, based on the bounding edge representation. When one glyph is inside another, the quadrant of the larger object in which the smaller object is located is encoded via a relationship, based on psychological evidence of qualitative and metric encoding in simple figures (Huttenlocher et al., 1991).

3.2.4. Interactions with conceptual knowledge

As described above, conceptual labelling of glyphs, relation glyphs, and annotation glyphs represent links between visual and conceptual knowledge. In addition, the visual relationships between glyphs often suggest conceptual relationships between the objects they depict. For example, two glyphs that visually contact one another might suggest that the depicted objects are in physical contact (i.e., connected, hinged, etc.). CogSketch can, on demand, create a list of potential conceptual relationships for pairs of entities that are suggested by the visual relationships between their glyphs and the concepts that they represent. Users can choose which relationship holds in order to further inform CogSketch of their intended meaning (Forbus et al., 2005).

3.3. Bringing it all together

The symbolic information computed by CogSketch's visual processing is stored in a working memory, organized by subsketch and layer. This working memory is also connected to the knowledge base, which can store sketches both propositionally and with ink as required. The reasoning engine used provides basic propositional inference and forward-chaining rules used in bookkeeping (Forbus & de Kleer, 1993). Access to visual operations is provided through the reasoning engine via procedural attachment, enabling queries and results to fluidly combine visual and conceptual terms. Analogical mapping, retrieval, and generalization are built into CogSketch, operating over the working memory and knowledge base representations. For example, retrieving sketches using MAC/FAC to then reason analogically with them using SME has been the core operation in AI systems that solve everyday physical reasoning problems from a human-normed test (Klenk et al., 2005) and generate plausible conceptual interpretations of visual relationships in sketches (Forbus et al., 2005).

4. CogSketch as cognitive simulation

We have used CogSketch to simulate a variety of visual and spatial reasoning tasks. These simulations provide evidence that our hypotheses in Section 2 are correct. This section summarizes some of our simulation experiments, pointing to relevant publications for further details. Note that the representations and processing in CogSketch are still evolving, as we learn how to improve its cognitive fidelity with each new experiment. This means that some of the simulation studies below use slightly different versions of the representations described above. We are now using this body of simulation results to perform a kind of tomography, using the multitude of constraints they provide to develop a single reference set of representations and processes.

4.1. Geometric analogies

In his seminal work, Evans (1968) wrote a program to solve geometric analogy problems (see Fig. 4), the first program to do any kind of analogy. Surprisingly, although some researchers have discussed other potential models for this task (e.g., Bohan & O'Donoghue, 2000; Schwering, Krumnack, Kuehneberger, & Gust, 2007), we are unaware of any simulation since then that can perform this task over the range of problems that Evans used, with automatically encoded stimuli. Consequently, we have used CogSketch to develop a new model of this task (Lovett et al., 2009b). Its key insight is that these problems can be solved through a process of *two-stage structure mapping*. In the first stage, the model compares image A to image B to compute Δ(A,B), a representation of the differences between images A and B. Δs are based on candidate inferences, part of the mapping produced by SME when it compares two cases.[5] Similarly, the model compares image C to each of the five possible answers to produce a Δ(C,x) for each answer. In the second stage, the Δs produced by SME in the first stage are fed back through SME for a second comparison. In this stage, Δ(A,B) is compared to the Δ's for each of the five possible answers, and the description with the most similar Δ is chosen as the answer. By contrast, Evans' ANALOGY system casts the problem as finding transformations between figures. There are many transformations that could account for each Δ, making transformation-based approaches more difficult. Our model uses CogSketch to encode the stimuli. Because there are several different strategies that can be used for encoding (e.g., preferring rotation vs. reflection) and mapping (e.g., whether or not to examine alternate interpretations), our model predicts that some problems will involve more comparisons, and hence longer reaction times, in people. These predictions were subsequently borne out in an experiment (Lovett et al., 2009b).

4.2. Raven's progressive matrices

This test is a widely used test of general intelligence (Raven, Raven, & Court, 1998). A slight variation of the two-stage structure mapping model used for geometric analogies can be used directly with CogSketch's current visual representations to achieve the same

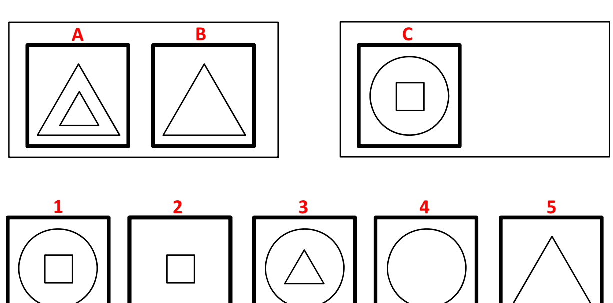

Fig. 4. A sample geometric analogy problem.

level of performance as adult Americans on two sections out of five on the test (Lovett, Forbus, & Usher, 2007). We are currently extending the model so that it can handle the entire test. The best prior model (Carpenter, Just, & Shell, 1990) used hand-coded representations of the stimuli.

4.3. Oddity task

In oddity tasks, participants are shown an array of images and asked to pick the one that does not belong. For example, in Dehaene, Izard, Pica, and Spelke (2006) an oddity task was used to look at geometric reasoning in Americans and Mundurukú, an indigenous South American group. Our model of this task uses SEQL to produce generalizations from subsets of the array and to look for elements that are substantially lower in similarity to these generalizations (Lovett, Lockwood, & Forbus, 2008). The inputs for the simulation were copy-and-pasted versions of the Power Point stimuli used in the original experiment. The model performs as well as most human subjects and shows an error pattern similar to humans. In addition, an ablation experiment (on the model) suggests why certain kinds of problems are hard for people.

4.4. Spatial language learning

While spatial prepositions make up a relatively small set of the words in any given language, the process of assigning them to a given visual scene is actually quite complex. Psychological studies have shown that a variety of factors, including geometry, functional roles of the objects, whether those roles are being fulfilled, and naive physics, can all play a role in how people assign prepositions to scenes (Coventry, Prat-Sala, & Richards, 2001; Feist & Gentner, 1998). The combination of visual and conceptual relationships available in CogSketch has enabled us to model how spatial prepositions involving containment and support are learned in both English and Dutch (Lockwood, Lovett, & Forbus, 2008). Using the stimuli from Gentner and Bowerman (2009), equivalent sketches were drawn with CogSketch, containing both geometric information and conceptual information (e.g., that the small round things in the left of Fig. 5 are raindrops and the entity enclosing them is a

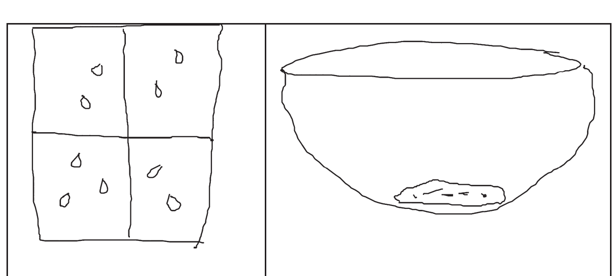

Fig. 5. Sample stimuli for the spatial language category learning experiments.

window, while the right stimulus involves a cookie and a bowl). Each training example was labelled with the appropriate spatial preposition: *in* and *on* when learning English, and *in*, *op*, *aan,* and *om* when learning Dutch. There were 32 examples in all, with 8 for *in* and 24 for *on* in English, and 8 each for *in*, *op*, *aan*, and *om* in Dutch. For each preposition, SEQL was given the exemplars for it and produced a set of probabilistic generalizations and unassimilated exemplars representing its understanding of that preposition. SEQL's ability to produce multiple generalizations enables it to handle disjunctive concepts. Unassimilated exemplars allow it to represent outliers, which may remain outliers, or become assimilated into some future generalization as the distribution of exemplars it receives changes. To test the models it learned, we used leave-one-out cross-validation, taking the model containing the most similar generalization or unassimilated exemplar to the new exemplar (as computed via SME) to be the system's labeling. The model was able to successfully learn these prepositions with only 8–24 sketches per preposition, which is several orders of magnitude fewer training examples than prior cognitive models (e.g., Regier, 1996).

5. CogSketch as educational software platform

A platform, in computing terms, is capable of supporting multiple types of systems. Consequently, we are exploring two distinct models of educational software using CogSketch: *Sketch Worksheets* and the *Design Buddy*. Each is discussed in turn.

5.1. Sketch worksheets

Paper-based worksheets are a staple in many classrooms. For example, in a geology class, students might be asked to highlight a fault on a photograph or draw the layers of the Earth. Sketching is a valuable way of learning spatial relationships. With pencil-and-paper sketching, feedback is delayed, and assessment is time-consuming and difficult. Experience with intelligent tutoring systems indicates that immediate feedback leads to better student learning (Corbett & Anderson, 2001). With CogSketch, we can provide rapid feedback to students, and hopefully make assessment simpler and more efficient, thereby improving learning.

Fig. 6 shows a sketch worksheet for a physical geology class, where the ink illustrates a typical student response. Students outline the geological features by creating glyphs, labelled with the appropriate concept, over the photograph. Coaching is provided by using SME to compare the student's sketch with the instructor's sketch. (Internally, the student's sketch is a subsketch, and the instructor's sketch is another subsketch which is kept hidden from the student.) Potential problems with the student sketch are found by analyzing the correspondences and candidate inferences of the mapping that SME produces for this comparison. For example, when a worksheet is developed, the instructor marks which facts are important and what advice to provide if they are not in the student's sketch (e.g., ''Is this really the location of the hanging wall?''), as indicated by a candidate inference from the

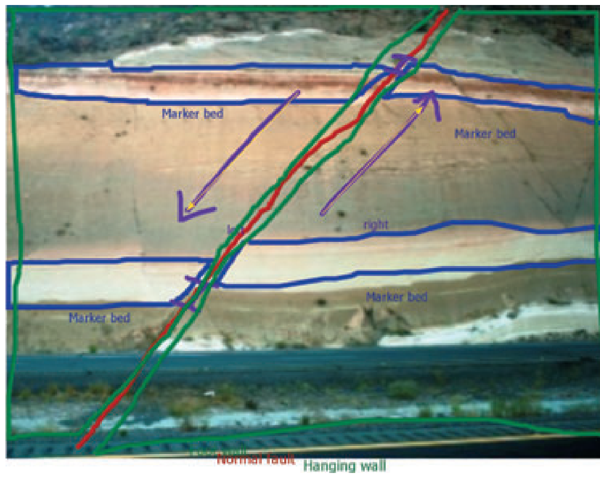

Fig. 6. A CogSketch worksheet with student response. Students were asked to draw the fault, the marker beds and which way they moved, and identify the hanging wall and foot wall.

teacher's sketch to the student sketch involving that fact. The student can then move and/or redraw their glyphs to improve their sketch, and ask for more help. Worksheets are developed through an authoring environment provided by CogSketch. Authors choose what concepts to include as possible conceptual labels, whether numerical values should be entered for some properties (e.g., the radius of the Earth's core, in a worksheet on the layers of the Earth) via annotation glyphs, and names and commentary on each concept and relationship, if the defaults are not suitable.

The sketch worksheet model is designed to be simple and general-purpose: If the appropriate KB concepts (or close stand-ins) can be found, a worksheet for that problem can be created. The current version of the authoring environment requires some understanding of the OpenCyc KB conventions and contents, which is daunting for most instructors without help. Most ITS authoring environments do not use independently developed off-the-shelf knowledge bases (e.g., Aleven, Sewall, McLaren, & Koedinger, 2006), requiring them to start from scratch (or from previously constructed systems with the same environment), which is more work but does ensure that whatever linguistic information the environment needs is entered along with new knowledge. We plan to use and expand natural language resources in the KB to make authoring easier. The first classroom use of worksheets was Fall 2009, in a physical geology course at Northwestern University taught by Prof. Brad Sageman. The first assignment involved four sketch worksheets and was quite successful, based on instructor and student feedback. Consequently, the instructor added a second sketch worksheet assignment, where students drew the constituents of the carbon cycle, was added (Yin, Forbus, Usher, Sageman, & Jee, 2010).

There is already some evidence from a laboratory experiment that CogSketch could be useful in automated assessments. Jee, Gentner, Forbus, Sageman, and Uttal (2009) found that when experts versus novices drew geological processes, or marked up images with geological features, there were distinct and easily recognizable differences between the two groups. In process diagrams, experts tended to include more arrows, which in such diagrams indicate the processes that are occurring and relate different aspects of the cycle to one another, and they tend to begin their diagram with such information. In marking up photographs of geological formations, the experts tend to draw more geologically relevant features, often in an idealized manner. This cannot be attributed to differences in drawing skill, since drawings of control photographs (e.g., fruit, lasagna) were indistinguishable. Importantly, the same pattern of results hold for sketching from memory, for copying, and for tracing. This suggests that comparisons of student sketches in a very simple copying task could be diagnostic of their mental models of the domain, analogous to the use of a sorting task to ascertain expertise (Chi, Feltovich, & Glaser, 1981). As other researchers have noted (Cheng & Rojas-Anaya, 2008), timing information can provide another implicit measure of expertise. CogSketch records timestamps for each ink point drawn and other interface events, which we are using to investigate other possible assessment measures.

5.2. Design Buddy

Engineers must communicate with their teammates and with clients in developing and refining designs. Using sketches to communicate is an essential part of the process. CAD software is only used in later stages of design, called *detailed design.* The early stages (*conceptual design*) are where the key ideas are worked out, to see if a design might be suitable before doing detailed designs. At Northwestern, first- and second-year students learn design and communications in an integrated manner, creating designs and prototypes that address real-world problems for external clients. Examples include patients at the Rehabilitation Institute of Chicago, whose physical handicaps require new tools to help them accomplish everyday tasks, such as chopping vegetables or trimming their nails. The instructors find that one of their hardest pedagogical problems is teaching students to use sketches to communicate their designs. Consequently, we are creating a CogSketch application, the *Design Buddy*, to tackle this problem.

The Design Buddy is a form of teachable agent (Blair, Schwartz, Biswas, & Leelawong, 2006) that gives students practice in explaining designs. Students explain their design by drawing a set of subsketches indicating the distinct intended behaviors of their design (see Fig. 2). Transitions on the metalayer indicate how one behavior leads to another. This is an example of a comic graph, which can be viewed as a form of comic strip, although there can be branches (representing different possible outcomes) and cycles (representing repetitive behaviors). In addition, they can make certain kinds of simple English statements about particular states, transitions between them, and purposes using a form-based interface. The Design Buddy critiques their description of intended behaviors, providing feedback to the student. It does this by qualitatively reasoning about the behaviors it believes are possible in the system as sketched and comparing them to the behaviors described by the student to look for mismatches. It also looks at each transition in the student's sketch, analyzing it to see if it is physically possible, given what it knows. Discrepancies between the student's explanation of the intended behavior and the Design Buddy's understanding of the possible behaviors are used to provide feedback (Wetzel & Forbus, 2009).

This application is significantly more difficult than worksheets for four reasons. First, it involves more substantial domain reasoning, rather than just matching. Design Buddy uses a qualitative model of mechanics (Kim, 1993; Nielsen, 1988) for causal reasoning about surfaces, forces, and motion. Qualitative mechanics is a natural fit for conceptual design: Most of the parameters needed for numerical simulation simply do not exist at this stage of design, and qualitative models capture the kinds of causal explanations that designers produce when talking about their designs. Second, the interface must be sufficiently natural to communicate complex behaviors without distracting the student too much. Third, we must develop coaching strategies that help students learn to explain designs in terms that practicing engineers would use. Finally, the assigned projects change every quarter, and a wide variety of design problems arise. We have started to tackle the last problem, of ensuring broad coverage, by analyzing a corpus of student designs from previous years. Out of 39 projects, 19 did not involve mechanics or motion of solid objects (e.g., electrical circuit problems, fluid flow problems), and hence CogSketch could not handle them. Of the remaining 20, four required 3D reasoning beyond what CogSketch can currently do, four required reasoning about gears, but the final 12 designs could be handled by CogSketch in its current form. Currently we are extending CogSketch to handle all of the motion-oriented designs, and doing pull-out studies with Northwestern students to refine the interface and coaching strategies.

6. Other related work

We are inspired in part by Saund and Mahoney's perceptual organization approach to sketching (Saund, Mahoney, Fleet, Larner, & Lank, 2002), which shows how human-like understanding of ink can lead to more natural editing interactions. We differ from them in our addition of conceptual understanding to the software, and in our collaboration with psychologists to calibrate our visual processing with human data as much as possible. The SketchIt system of Stahovich, Davis, and Shrobe (2000) shares our concern with carrying out qualitative mechanics analyses of sketched devices, but it requires users to hand-segment surfaces. That may be reasonable for a professional design tool, but for an educational setting we must do this automatically. The Electronic Cocktail Napkin (Gross & Do, 1996) was an earlier sketch understanding system meant to facilitate design. Like our system, it was able to decompose glyphs into their component edges. However, it was focused more on learning to recognize the objects represented by glyphs and less on determining how different glyphs relate to each other. Adler and Davis (2004) describes their ASSIST system, which allows users to sketch a physical system while verbally describing it. A speech recognition system parsed the description and used the information to refine the sketch (e.g., positioning objects such that they are equally spaced in a row). This type of system requires the designer to specify the meaning of the words that are of interest in sketching, thus limiting the system's breadth. In contrast, our system ties conceptual labels to a large, preexisting knowledge base.

7. Discussion and future work

CogSketch is an ambitious project, and we are far from achieving our vision. Nevertheless, we are encouraged by our results so far. As Section 4 indicates, we have already successfully simulated a number of psychological findings, which lays some of the groundwork for our education experiments. The first version of CogSketch is already publicly available.[6] We are eager for feedback that helps us make CogSketch more usable by the research and education communities. To support AI researchers, for example, CogSketch has an interface that gives other programs access to all of its visual processing and reasoning capabilities. Included in the distribution are sample sketch worksheets, the sketch worksheet authoring environment, and a sketch with all of Evans' geometric analogy problems and the ability to run our model on them, to support experimentation. Community feedback is helping guide us in CogSketch's future development, so we can realize our vision and help make sketch-based intelligent systems commonplace in education.

Notes

1. For this paper, we take visual computations to be processing that occurs without regard for the relationship between the sketch and the external environment, and spatial computations to be processing that takes that relationship into account. When someone is looking at a U.S. map as a visual display, California is to the left of Illinois, but spatially it is to the west.
2. SME stands for ''Structure-Mapping Engine,'' MAC/FAC stands for ''Many are called, few are chosen,'' and SEQL stands for ''Sequential Learning.''
3. http://www.opencyc.org
4. RCC-8 stands for ''Region Connection Calculus.'' Different calculi have different numbers of mutually exclusive and collectively exhaustive relationships; this version has eight.
5. SME can compute candidate inferences in both directions, that is, from base to target and from target to base.
6. The phrase ''CogSketch'' to any reasonable search engine will yield the current URL.

Acknowledgments

This work was supported by NSF SLC grant SBE-0541957, the Spatial Intelligence and Learning Center (SILC), and by a grant from the Intelligent and Autonomous Systems Program of the Office of Naval Research.

References

Adler, A., & Davis, R. (2004). Speech and sketching for multimodal design. *Proceedings of the 9th International Conference on Intelligent User Interfaces*, Island of Madeira, Portugal.

Aleven, V., Sewall, J., McLaren, B. M., & Koedinger, K. R. (2006). Rapid authoring of intelligent tutors for real-world and experimental use. In Kinshuk, R. Koper, P. Kommers, P. Kirschner, D. G. Sampson, & W. Didderen (Eds.), *Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006)* (pp. 847–851). Los Alamitos, CA: IEEE Computer Society.

Alvarado, C., Oltmans, M., & Davis, R. (2002). A framework for multi-domain sketch recognition. *Proceedings of AAAI Spring Symposium on Sketch Understanding*, Palo Alto, CA.

Barker, K., Blythe, J., Borchardt, G., Chaudhri, V., Clark, P., Cohen, P., Fitzgerald, J., Forbus, K., Gil, Y., Katz, B., Kim, J., King, G., Mishra, S., Morrison, C., Murray, K., Otstott, C., Porter, B., Schrag, R., Uribe, T., Usher, J., & Yeh, P. (2003). A knowledge acquisition tool for course of action analysis. *Proceedings of the Innovative Applications of Artificial Intelligence Conference*, Acapulco, Mexico.

Blair, K., Schwartz, D., Biswas, G., & Leelawong, K. (2007). Pedagogical agents for learning by teaching: Teachable agents. *Education Technology*, *47*, 56–61.

Bohan, A., & O'Donoghue, D. (2000). LUDI: A model for geometric analogies using attribute matching. *Proceedings of the 11th Artificial Intelligence and Cognitive Science Conference (AICS-2000)*, Galway, Ireland.

Bradley, E. (1995). Autonomous exploration and control of chaotic systems. *Cybernetics and Systems*, *26*, 299–319.

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.

Cheng, P. C.-H., & Rojas-Anaya, H. (2008). Measuring mathematic formula writing competence: An application of graphical protocol analysis. In B. C. Love, K. McRae, & V. M. Sloutsky (Eds.), *Proceedings of the Thirtieth Annual Conference of the Cognitive Science Society* (pp. 869–874). Hillsdale, NJ: Lawrence Erlbaum Associates.

Chi, M., Feltovich, P., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, *5*, 121–152.

Cohn, A. (1996) Calculi for qualitative spatial reasoning. In J. Calmet, J. A. Campbell, & J. Pfalzgraph (Eds.), *Artificial Intelligence and Symbolic Mathematical Computation, LNCS 1138* (pp. 124–143). New York: Springer Verlag.

Corbett, A. T., & Anderson, J. R. (2001). Locus of feedback control in computer-based tutoring: Impact on learning rate, achievement and attitudes. *Proceedings of ACM CHI'2001 Conference on Human Factors in Computing Systems*, Seattle, WA, 245–252.

Coventry, K. R., Prat-Sala, M., & Richards, L. V. (2001). The interplay between geometry and function in the comprehension of 'over', 'under', 'above', and 'below'. *Journal of Memory and Language*, *44*, 376–398.

Dehaene, S., Izard, V., Pica, P., & Spelke, E. (2006). Core knowledge of geometry in an Amazonian Indigene Group. *Science*, *311*, 381–384.

Edwards, G., & Moulin, B. (1998). Toward the simulation of spatial mental images using the Voronoi model. In P. Olivier & K. P. Gapp (Eds.), *Representation and processing of spatial expressions* (pp. 163–184). Hillsdale, NJ: LEA Press.

Evans, T. (1968). A program for the solution of geometric-analogy intelligence test questions. In M. Minsky (Ed.), *Semantic information processing* (pp. 271–353). Cambridge, MA: MIT Press.

Falkenhainer, B., Forbus, K., & Gentner, D. (1989). The Structure-Mapping Engine: Algorithm and examples. *Artificial Intelligence*, *41*, 1–63.

Feist, M. I., & Gentner, D. (1998). On plates, bowls, and dishes: Factors in the use of English 'in' and 'on'. *Proceedings of the 20th Annual Conference of the Cognitive Science Society*, Madison, WI.

Ferguson, R., Aminoff, A., & Gentner, D. (1996). Modeling qualitative differences in symmetry judgments. *Proceedings of CogSci96*, La Jolla, CA.

Forbus, K. (1983). Qualitative reasoning about space and motion. In D. Gentner & A. Stevens (Eds.), *Mental models* (pp. 53–74). Hillsdale, NJ: LEA associates, Inc.

Forbus, K. (2007). Qualitative modeling. In F. Harmelen, V. Lifschitz, & B. Porter (Eds.), *Handbook of knowledge representation* (pp. 361–394). New York: Elsevier.

Forbus, K., & de Kleer, J. (1993). *Building problem solvers*. Cambridge, MA: MIT Press.

Forbus, K., Ferguson, R., & Usher, J. (2001) Towards a computational model of sketching. *Proceedings of Intelligent User Interfaces*, Santa Fe, NM, pp. 77–83.

Forbus, K., Gentner, D., & Law, K. (1995). MAC/FAC: A model of Similarity-based Retrieval. *Cognitive Science*, *19*(2), 141–205.

Forbus, K., Mostek, T., & Ferguson, R. (2002) An analogy ontology for integrating analogical processing and first-principles reasoning. *Proceedings of IAAI-02*, Edmonton, AB.

Forbus, K., Nielsen, P., & Faltings, B. (1991). Qualitative spatial reasoning: The CLOCK Project. *Artificial Intelligence 51*(1–3), 417–471.

Forbus, K., & Oblinger, D. (1990). Making SME greedy and pragmatic. *Proceedings of the 12th Annual Meeting of the Cognitive Science Society*, Cambridge, MA.

Forbus, K., Usher, J., & Chapman, V. (2003). Sketching for military courses of action diagrams. *Proceedings of IUI'03*, Miami, FL.

Forbus, K., Usher, J., & Tomai, E. (2005). Analogical learning of visual/conceptual relationships in sketches. *Proceedings of AAAI-05*, Pittsburgh, PA.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*, 155–170.

Gentner, D., & Bowerman, M. (2009). Why some spatial semantic categories are harder to learn than others: The Typological Prevalence hypothesis. In J. Guo, E. Lieven, S. Ervin-Tripp, N. Budwig, S. Özçaliskan, & K. Nakamura (Eds.), *Crosslinguistic approaches to the psychology of language: Research in the tradition of Dan Isaac Slobin* (pp. 465–480). New York: Lawrence Erlbaum Associates.

Gross, M. D., & Do, E. (1996). Demonstrating the electronic cocktail napkin: A paper-like interface for early design. *ACM Conference on Human Factors (CHI '96)*, 5–6.

Halstead, D., & Forbus, K. (2005). Transforming between propositions and features: Bridging the gap. *Proceedings of AAA-05*, Pittsburgh, PA.

Hoffman, D. D., & Richards, W. (1984). Parts of recognition. *Cognition*, *18*, 65–96.

Huang, X., & Zhao, F. (2000). Relation based aggregation: Finding objects in largespatial datasets. *Intelligent Data Analysis*, *4*, 129–147.

Huttenlocher, J., Hedges, L. V., & Duncan, S. (1991). Categories and particulars: Prototype effects in estimating location. *Psychological Review*, *98*(3), 352–376.

Jee, B., Gentner, D., Forbus, K., Sageman, B., & Uttal, D. (2009). Drawing on experience: Use of sketching to evaluate knowledge of spatial scientific concepts. *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, Amsterdam, The Netherlands.

Joscowicz, L., & Sacks, E. (1991). Computational kinematics. *Artificial Intelligence*, *51*, 381–416.

Kim, H. (1993). *Qualitative reasoning about fluids and mechanics*, Ph.D. dissertation and ILS Technical Report, Northwestern University.

Klenk, M., Forbus, K., Tomai, E., Kim, H., & Kyckelhahn, B. (2005). Solving everyday physical reasoning problems by analogy using sketches. *Proceedings of 20th National Conference on Artificial Intelligence (AAAI-05)*, Pittsburgh, PA.

Kosslyn, S. M., Chabris, C. F., Marsolek, C. M., & Koenig, O. (1992). Categorical versus coordinate spatial representations: Computational analyses and computer simulations. *Journal of Experimental Psychology: Human Perception and Performance*, *18*, 562–577.

Kuehne, S., Forbus, K., Gentner, D., & Quinn, B. (2000). SEQL: Category learning as progressive abstraction using structure mapping. *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society*, Philadelphia, PA.

Landay, J. A., Hong, J., Klemmer, S., Lin, J., & Newman, M. (2002). Informal PUIs: No recognition required. *Proceedings of AAAI Spring Symposium on Sketch Understanding*, Palo Alto, CA.

Lee, W., De Silva, R., Peterson, E. J., Calfee, R. C., & Stahovich, T. F. (2007). Newton's Pen – A pen-based tutoring system for statics. *Proceedings of the Eurographics Workshop on Sketch-Based Interfaces and Modeling*, Riverside, CA.

Lockwood, K., Lovett, A., & Forbus, K. (2008). Automatic classification of containment and support spatial relations in English and Dutch. *Proceedings of Spatial Cognition*, Frieburg, Germany.

Lovett, A., Forbus, K., & Usher, J. (2007). Analogy with qualitative spatial representations can simulate solving Raven's Progressive Matrices. *Proceedings of the 29th Annual Meeting of the Cognitive Science Society*, Nashville, TN.

Lovett, A., Gentner, D., Forbus, K., & Sagi, E. (2009a). Using analogical mapping to simulate time-course phenomena in perceptual similarity. *Cognitive Systems Research*, *10*, 216–228.

Lovett, A., Lockwood, K., & Forbus, K. (2008). Modeling cross-cultural performance on the visual oddity task. *Proceedings of Spatial Cognition 2008*, Freiburg, Germany.

Lovett, A., Tomai, E., Forbus, K., & Usher, J. (2009b). Solving geometric analogy problems through two-stage analogical mapping. *Cognitive Science*, *33*(7), 1192–1231.

Nielsen, P. E. (1988). *A qualitative approach to rigid body mechanics*. (Tech. Rep. No. UIUCDCS-R-88-1469; UILU-ENG-88-1775). Urbana, IL: University of Illinois at Urbana-Champaign, Department of Computer Science.

Pittman, J., Smith, I., Cohen, P., Oviatt, S., & Yang, T. (1996). Quickset: A multimodal interface for military simulations. *Proceedings of the Sixth Conference on Computer-Generated Forces and Behavioral Representation*, Orlando, FL, 217–224.

Rasch, R., Kott, A., & Forbus, K. (2002). AI on the battlefield: An experimental exploration. *Proceedings of the 14th Innovative Applications of Artificial Intelligence Conference*, Edmonton, AB.

Raven, J., Raven, J. C., & Court, J. H. (1998). *Manual for Raven's Progressive Matrices and Vocabulary Scales: Section 1. General Overview*. Oxford, England: Oxford Psychologists Press.

Regier, T. (1996). *Thehuman semantic potential: Spatial language and constrained connectionism*. Cambridge, MA: MIT Press.

Saund, E., Mahoney, J., Fleet, D., Larner, D., & Lank, E. (2002). Perceptualoganization as a foundation for intelligent sketch editing. *Proceedings of AAAI Spring Symposium on Sketch Understanding*, Palo Alto, CA, 118–125.

Schwering, A., Krumnack, U., Kuehneberger, K., & Gust, H. (2007). Using gestalt principles to compute analogies of geometric figures. *Proceedings of the 29th Annual Conference of the Cognitive Science Society*, Nashville, TN.

Shepard, R. N., & Cooper, L. A. (1982). *Mental images and their transformations*. Cambridge: MIT Press.

Stahovich, T., Davis, R., & Shrobe, H. (2000). Qualitative rigid-body mechanics. *Artificial Intelligence 119* (1–2), 19–60.

Wetzel, J., & Forbus, K. (2009). Automated critique of sketched mechanisms. *Proceedings of the 21st Innovative Applications of Artificial Intelligence Conference*, Pasadena, CA.

Yin, J., Forbus, K. D., Usher, J., Sageman, B., & Jee, B. (2010) Sketch Worksheets: A Sketch-based Educational Software System. *Proceedings of the 22nd Innovative Applications of Artificial Intelligence Conference*, Atlanta.

Yip, K. (1991). *KAM: A System for Intelligently Guiding Numerical Experimentation by Computer*. Cambridge, MA: MIT Press.