necessity for creating performance support tools for military operations. Although most of our experience has been in military tasks, the situation seems similar in other human geospatial reasoning tasks [8].

This paper describes the techniques we have developed for qualitative spatial reasoning about sketch maps. We start by reviewing our approach to sketching and *nuSketch Battlespace*, our battlespace sketching software that has been used in several successful experiments. Next we provide an overview of the spatial representations of sketches and glyphs and the processing architecture that handles spatial computations. Then we describe the computation of spatial relationships, including qualitative topology and Voronoi diagrams. Path-finding and position-finding, two key tasks, are discussed next. We describe how these techniques are combined with analogical processing to provide a simple form of enemy intent hypothesis generation. Finally, we discuss plans for future work.

2. Overview of nuSketch Battlespace

Sketching is a form of multimodal interaction, where participants use a combination of interactive drawing and language to provide high-bandwidth communication. Sketching is especially effective in tasks that involve space, e.g., geospatial reasoning. While today's software is far being as fluent as sketching with a person, progress in multimodal interfaces has produced interfaces that are significantly more natural than standard mice/menu systems (cf. [2]).

The typical approach in multimodal interfaces is (a) to provide a more natural interface to a legacy software system and (b) to focus on recognition [1,2]. While this approach has led to useful systems, it has some serious limitations. First, today's statistical recognizers are not very good (indeed, much of the multimodal literature focuses on using multiple modalities to overcome the limitations in individual modalities). Our military users, based on their experience with previous multimodal interfaces, generally flatly refuse to use any system that requires speech recognition. Second, even if recognition improves to human-level or beyond, there is still the problem of providing software with a conceptual understanding of what is being sketched. Such knowledge is crucial for creating performance-support systems.

Our approach in the *nuSketch architecture* [13] is quite different and complements traditional multimodal research. We avoid recognition issues by using clever interface design. We focus instead on providing richer visual and conceptual understanding of what is sketched. We have created two systems based on this architecture: nuSketch Battlespace (nSB) [18], specialized for battlespace reasoning, and the sketching Knowledge Entry Associate (sKEA) [17], a general-purpose knowledge capture system. While sKEA also does geospatial reasoning when appropriate – the two systems share a common code base – we focus in this paper on nuSketch Battlespace for brevity.

nuSketch Battlespace is designed to help users develop courses of action (COAs) for land forces. It uses a large knowledge base concerning specialized military concepts as well as general common sense. We use a subset of Cycorp's Cyc knowledge base contents[1], with extensions developed by our group for qualitative and analogical reasoning and by the DARPA community for military concepts and reasoning. The interface uses special-purpose interface techniques to enable users to specify conceptual information (including the types of entities being sketched, timing information, and intent of actions), organized into layers to control complexity. Users can sketch terrain, specialized areas and paths (e.g., engagement areas, axes of advance), position units, and assign tasks and the reasons for doing them. Since planning in uncertain situations often involves exploring multiple hypotheses, and plans can involve complex sequential behavior and conditionals, nSB enables users to describe and link multiple states into a *comic graph*, a visualization based on action-augmented envisionments. The interface techniques that enable us to avoid recognition are described in [18]; here our focus is on the qualitative spatial reasoning the system performs.

nuSketch Battlespace has been successfully used in several experiments. First, a early version was combined with a natural-language input system (by AlphaTech and Teknowledge) and BBN's CADET system that generates synchronization matrices in an experiment to see if active-duty military personnel could successfully create COAs. As described in [24], commanders were able to generate COAs three to five times faster, without any degradation in plan quality. In DARPA's Rapid Knowledge Formation program, nSB was adopted by both teams to provide sketching and spatial reasoning services for their integrated knowledge capture systems. The KRAKEN system from the Cycorp team combined nSB with their natural language facilities, and the SHAKEN system from the SRI team combined nSB with their concept map facilities. In an evaluation run by an independent contractor this fall, both teams were able to demonstrate that military subject-matter experts were able to author COA critiquing knowledge using these systems. In DARPA's Command Post of the Future program, we have received long-term, valuable formative feedback from a variety of retired military officers. Their feedback has helped us improve the system to the point where we can have generals doing analogies between battlespace states within an hour of sitting down with the software for the first time.

3. Representing glyphs and sketches

This section describes the underlying ontology of sketches that we use. The basic unit in a sketch is a *glyph*. Every glyph has *ink* and its *content*. The ink consists of one or more polylines, representing what the

---

[1] We use our own KB and reasoning system instead of Cyc that is optimized for our needs.

user drew when specifying that glyph. (Each polyline includes width and color information in addition to its points.) The content is a conceptual entity, the kind of thing that the glyph is representing. For example, if a user drew a mountain range, there would be an entity created to represent the glyph itself and an entity to represent the mountain range. While each subsketch depicting the mountain range would have a distinct glyph, the contents of those glyphs would all be the same entity.

The type of a glyph's contents affects the interpretation of its spatial properties. For example, the spatial extent of glyphs representing mountains and lakes is taken to be the spatial extent of that terrain feature. On the other hand, the spatial extent of a military unit is ignored, since the size of such glyphs by convention has nothing to do with its footprint on the ground, so only its centroid is used in spatial reasoning. Path-like terrain features such as roads and rivers have a one-dimensional extent, but their width is not tied to the width of the line depicting them, since that would unduly burden our users' drawing abilities. In contrast, paths introduced in planning actions do have widths that are specified by special gestures during sketching, because they provide spatial constraints on the movements of units. (Regions just outside the path might be targets of artillery, and avoiding friendly fire is an important task constraint.)

While some basic spatial properties of glyphs are computed (described below), we do not perform any detailed shape reasoning on the ink comprising a glyph, nor do we attempt to visually decompose it. We call this *blob semantics* because it focuses on spatial relationships between glyphs rather than detailed reasoning about the visual structure of glyphs themselves. While inappropriate for recognition based on detailed visual similarity of specific features, it is an excellent approximation for most geospatial reasoning, where the focus is on configural relationships between glyphs. Given the crude nature of sketch maps, people are unlikely to be extremely accurate at reproducing shapes.

A sketch consists of one or more *subsketches*. Subsketches represent a coherent aspect of what is being sketched, such as a state of a plan, or a more detailed depiction or distinct perspective on something. Logically, subsketches are Cyc-style microtheories, local descriptions that must be internally consistent. In nSB, every subsketch represents a battlespace state. States can be partial, and are either hypothetical, observed, or planned. Visually, the user sees either a single subsketch at a time, or the *metalayer*, a special view where each subsketch is viewed as a glyph. The comic graph consists of these glyphs and relationships between them, expressed by drawing arrows between state glyphs.

Subsketches are composed of *layers*. In nSB, each layer represents a particular subset of information about a battlespace state. Examples include terrain features, friendly COA, and SITEMP (i.e., enemy COA). Every glyph exists on some layer. The layers of a subsketch are spatially registered, i.e., they share the same coordinate system. Distinct subsketches need not be spatially registered, although in nSB they tend to be. Logically, each layer in a subsketch is a microtheory. Visually, layers are depicted as overlays on a common workspace for that subsketch. The user can control whether or not a layer is visible, grayed out (which keeps layouts in focus without being distracting), or invisible, to control detail while sketching. nuSketch systems can also introduce new layers to display the results of their reasoning.

4. Spatial processing of glyphs

Spatial reasoning is carried out when a glyph is added or changed, and in response to queries from nSB reasoning facilities. nSB has two visual processors, which are threaded to enable computation while the user is thinking or sketching. We describe each in turn, as a prelude to the detailed discussion of the spatial operations.

The *ink processor* is responsible for computing basic spatial properties of glyphs and responding to queries concerning spatial relationships. Whenever a glyph is added or changed, basic spatial properties are computed for it, including a bounding box, area, overall orientation and roundness. Qualitative topological relationships are automatically computed between the new glyph and other glyphs on its layer.

The *vector processor* is responsible for maintaining a set of Voronoi diagrams describing spatial relationships between types of entities, and for the polygon operations used in position-finding and path-finding. Any time a glyph is added or changed, once the ink processor has updated its properties the Voronoi diagram(s) it is associated with are updated appropriately. When spatial constraints involving position-finding or path-finding need solving, the vector processor carries out the construction of obstacle and cost diagrams, the polygon operations needed to combine them, and the quad tree representation used in path-finding.

Conclusions reached by these processors are added to the LTMS-based working memory of the reasoner for that sketch. Timestamped assertions are used as assumptions in visual conclusions drawn by the system, so that when glyphs are moved, resized or deleted the appropriate conclusions are automatically retracted.

5. Spatial relationships between glyphs

Spatial relationships are the threads from which configural information is woven. Therefore computing them appropriately is a crucial problem for qualitative reasoning about sketches. We discuss four kinds of spatial relationships in turn: Qualitative topological relationships, Voronoi relationships, positional relationships, and relationships based on local frames of reference.

5.1 Qualitative topological relationships

We use the RCC8 algebra [3] to provide a basic set of qualitative relationships between glyphs. RCC8 is appropriate because it captures basic distinctions such as whether or not two glyphs are disjoint (DC), touching (EC), or inside one another (TPP, NTPP). These distinctions are used in several ways. First, they are used in controlling when to compute other relationships: computing whether or not one entity is east of another is moot unless they are DC, for example. Second, they suggest conceptual interpretations of relationships between the contents of the glyphs that they relate. For instance, an EC relationship between two glyphs which represent physical objects suggests that their contents might be touching. Finally, domain-specific inference rules can use these relationships when needed, e.g., containment.

Much of the work on RCC8 and other qualitative topological algebras has focused on using transitivity for efficient inference. For sketches the use of such tables is unnecessary, because we can simply calculate for each pair of glyphs what RCC8 relationship holds between them, based on the visual properties of their ink. By default, we compute RCC8 relationships between a glyph and everything else on its layer when it is first added or changed. RCC8 relationships with glyphs across layers in the same subsketch can be computed on demand during domain-specific reasoning.

5.2 Voronoi Relationships

Following [7], we use Voronoi diagrams to compute a variety of spatial relationships. Recall that, given a set of spatial entities (called *sites*, typically points), a Voronoi diagram consists of edges that are equidistant from a pair of points. The Delauney triangulation is the dual of the Voronoi, consisting of a set of arcs between sites that have an edge between them in the Voronoi diagram. As [7] describes, the Delauney triangulation provides a reasonable approximation to visual proximity, in that two sites are proximal exactly when there is an edge connecting them in the Delauney triangulation. Moreover, a number of approximations to spatial prepositions can be computed, including between and near. Again, these are approximations: It is known that, psychologically, spatial prepositions depend on functional and conceptual information as well as spatial information [4,10]. However, we have found them adequate for sketch maps.

Voronoi computations are defined in terms of sites being points, but glyphs have significant spatial extent. Consequently, adding a glyph to a Voronoi diagram involves adding sample points along the outer contour of the glyph's ink, each of which is treated as a site. These sites are marked with the glyph they derived from, so that while the Voronoi computations are done on the sampled sites, the results are expressed in terms of relationships between the glyphs. For example, two glyphs are `siteAdjacent` exactly when there exists a sample site on each glyph that is connected by an edge in the sample-level Delauney triangulation.

A key design feature in any system using Voronoi computations is what diagrams should be computed. We use several diagrams to capture different notions of proximity: A terrain-only diagram is useful for characterizing free space, and a units-only diagram is useful for grouping units, for example.

5.3 Positional relationships

Positional relationships provide qualitative position and orientation information with respect to a global coordinate frame. Positional relationships between contents are expressed in terms of compass directions. For example, a tank brigade can be south of a mountain and to the east of a bridge. Not all glyphs can participate in such relationships: The task of securing a bridge, while represented by a glyph in the sketch, is not itself something that participates in positional relations, although the location at which it occurs can.

A key design choice is what positional relationships should be computed. It might seem at first that, like RCC8 relationships, it could be worth computing positional relationships between every pair of RCC8-DC glyphs. This turns out to be a terrible strategy, both in terms of computational effort and in terms of the usefulness of the results. Computationally, positional relationships are used to provide concise summaries (if communicating a situation) and to provide a framework for describing the layout of a situation (for instance when computing spatial analogies). Consequently, we limit the automatic computation of them to pairs of geographic features and compute positional relations for other appropriate entities on demand.

5.4 Other frames of reference

Another type of positional relationship links two entities based on a local coordinate system. For example, if two entities are related to an oriented path, it is useful to talk about one entity being ahead, behind, or at the same location along that path. nuSketch computes such relationships on demand, using projection of the centroids of the entities to the closest point on the path to determine their relative position.

Some entities have a distinct orientation, even without having a path-like extent. Military units, for example, have fronts, flanks, and rears. Again, we compute such relationships on demand, based on orientation information associated with the entities.

6. Position-finding

Some of the most interesting implications of sketch maps involve constructing places: The good sites for a park, in an urban planning task, or a good site for an ambush, in a military setting. We use conceptual knowledge of the contents of glyphs, combined with spatial reasoning on their ink, to automatically construct regions that satisfy spatial and functional criteria.

Two important constraints in military spatial reasoning are fields of fire (i.e., what can someone's weapons hit?) and observation (i.e., what can someone see?). Some kinds of terrain features (e.g., mountains) block weapons, and thus provide cover. Other kinds of terrain features (e.g., forest) block visibility, and thus provide concealment. Cover and concealment are important concepts in military reasoning, since they provide protection from the enemy and deny them information[2]. Finding positions (i.e., regions of the sketch) that satisfy these properties is a critical spatial operation. For example, finding positions that provide concealment is an important sub-task in planning (or detecting) an ambush.

Our position-finding technique relies on polygon operations over relevant subsets of glyphs. Depending on the constraint(s) to be satisfied, some glyphs are treated as obstacles. New regions are constructed by projections from seed locations, subject to obstacle constraints. Regions that must satisfy multiple constraints are computed by combining the regions constructed for each constraint. The polygon operations of union, intersection and subtraction thus enable the conjunction, disjunction, and complement of constraints, respectively.

Terrain Type	Concealed?	Cover?
Mountains	Yes	Yes
Hills	Yes	Yes
Open/rolling hills	No	Yes
Forest	Yes	Partial
Scrub	Yes	Partial
Jungle	Yes	Partial
Swamp	No	No
Desert	No	No
Lake	No	No
River	No	No
Bridge	No	No
City	Yes	Yes
Road	No	No

Table 1: Concealment and cover provided by different terrain types

Let us consider concealment as an example. Suppose we are trying to find all regions where someone could hide from us. Domain knowledge indicates what kinds of terrain regions units can hide in (see Table 1), and thus what regions constitute obstacles. For each unit on our side, a new polygon is constructed by ray-casting to represent the region that is visible from that unit. (If there is numerical information as to limits of visibility, the polygon is also clipped using that information.) Let V be the union of these polygons, representing all of the areas that we can see. Let W be the polygons that results from subtracting out places where units cannot be (e.g., in lakes) from the entire sketch. (Notice that we allow polygons to have holes.) Then the set of polygons W – V constitutes the places where an enemy could hide. Fields of fire and cover, are computed similarly, using cover constraints and weapon ranges.

7. Path-finding

Planning and following routes is one of the major purposes of maps, and so path-finding is an important capability for sketch maps. As with position-finding, domain constraints are used to define what are obstacles, and hence by implication what is free space. What is an obstacle can depend on the type of unit moving: Forests are considered untrafficable for vehicles, for example, but trafficable by infantry. The costs of movement depend on the type of terrain. For example, it takes longer for infantry to move through a swamp than through a desert. In military planning, estimates of trafficability are often computed based on complex formulae involving specific details of vehicles and properties of soil and vegetation (e.g., rod cone index, stem spacing) [12]. This level of analysis can be automated using AI techniques, but it requires GIS data and a wealth of detail to do so [6]. For sketch maps, we have developed a simpler technique, for two reasons. First, sketch maps are often used in the early stages of planning, when many details have not yet been decided. Second, sketch maps are lower resolution than GIS systems, and hence are better suited for rough estimates than detailed calculations. Consequently, we use a simplified qualitative theory of trafficability, closer to the heuristic guidelines that we have seen used by commanders.

There is a standard qualitative representation for trafficability in military terrain analysis which divides space into regions that are unrestricted terrain (abbreviated UR or “go”), restricted terrain (abbreviated R or “slow go”), and severely restricted terrain (abbreviated SR or “no go”). Instead of demanding detailed descriptions of terrain, we assign trafficability categories based on the overall type of terrain. Since moving on foot is fundamentally more flexible than vehicles, our qualitative trafficability theory simplifies the vast array of units into two distinctions: armor versus infantry. Table 2 shows the trafficability implications of the terrain types in nSB.

Terrain Type	Armor	Infantry
Mountains	SR	R
Hills	R	UR
Open/rolling hills	UR	UR
Forest	SR	R
Scrub	UR	UR
Jungle	R	R
Swamp	R	R
Desert	UR	UR
Lake	SR	SR
River	SR	SR
Bridge	UR	UR
City	R	UR
Road	UR	UR

Table 2: Trafficability constraints

Terrain regions can intersect, which slightly complicates these assignments. For example, a road over a mountain range or through a swamp is still UR, while a lake in mountains remains SR. Given a sketch, we compute a single obstacle and cost diagram by finding the maximal partition under intersection of these regions, and assigning costs to regions with two terrain types based on rules like those above.

In path-finding, SR regions are treated as obstacles, and R regions are treated as higher-cost for travel than UR regions. Following [5] we use A* search over a quad tree representation for generating the lowest-cost obstacle-free path. Originally we had used a bitmap-

---

[2] Similar concepts, in terms of their computational structure, in urban planning include planning for drainage and for views.

based approach [15], but we were unable to make those techniques fast enough for interactive-time operation.

8. Example: Hypothesizing enemy intent by analogy

To illustrate the utility of these ideas, we demonstrate how they are used in the nSB subsystem that hypothesizes possible enemy actions. The inputs are a sketch of a precedent and a sketch representing the current situation. The output consists of a new layer which illustrates how, in the current situation, the enemy might attempt something similar to what they did in the precedent. We have simplified the general problem in several ways. First, we only generate hypotheses about a single enemy task, for constraint solving tractability. Second, we only consider precedents and situations consisting of single battlespace states. Third, we are providing the precedent as part of the input, rather than retrieving it automatically from a memory of experiences. However, even with these simplifications, this task represents a significant advance in the state of the art in combining analogical and spatial reasoning.

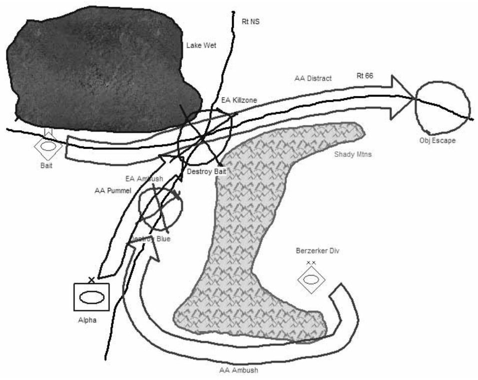

Figure 2: An ambush

Figure 2 shows an example precedent. In it, a small enemy unit (Bait) is trying to escape Alpha Battalion, which is planning to destroy it at EA Killzone. Unbeknownst to Alpha, this is a trap: Berserker Division, hiding behind the mountain range, attacks Alpha from the rear as Alpha goes after Bait, causing considerable damage. This precedent was created with nSB in the usual way, using a template-based interface to describe why the task was successful. In this case, the ambush is successful because the attacker was concealed and could travel to an engagement area on Alpha's path.

Figure 3 shows an example current situation, from another sketch. Your unit, Bravo, sees an enemy unit (Bait) trying to escape, and you are tempted to go after it. But, having heard about what happened to Alpha, you are worried. Using nSB, you can ask for hypothesized enemy tasks about the current situation based on the precedent sketched state. Its answer is shown in Figure 4: There are two places that an enemy unit might be hiding, to carry out an ambush similar to what happened before. The rest of this section describes how results like this one are computed.

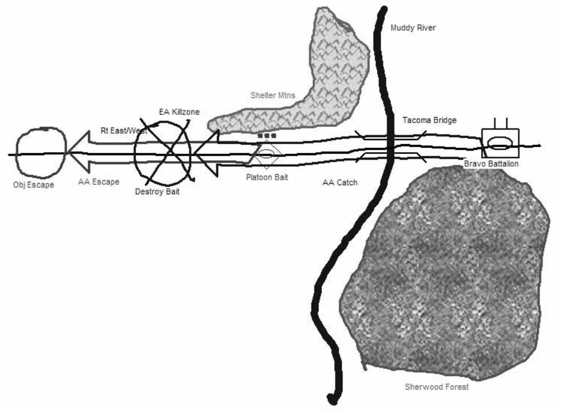

Figure 3: Current situation

A key aspect of our approach is the use of human-like analogical processing for comparisons. Our goal is to ensure that, within the limitations of our representations, things which look alike to human users will look alike to the software. This *shared similarity constraint* enables the software's conclusions to be more trusted by the user. We achieve a shared sense of similarity by using cognitive simulations of human analogical processing, over representations that approximate human visual representations. The cognitive simulation of analogical matching we use is the Structure-Mapping Engine (SME) [9], which is backed by considerable psychological evidence [19]. There is evidence that the structural alignment processes it models are operating in human visual processing [11], which makes using SME a reasonable choice. The shared similarity constraint has proven to be a valuable constraint on representation and reasoning choices, and has guided many of the representation and processing choices described in this paper.

When intent hypotheses are requested, nSB runs SME on the two descriptions, which are states from sketches. The descriptions include both visual and conceptual information. SME derives a set of candidate inferences about the current situation based on the comparison. So far, this is simply SME doing what it normally does. Next, the set of candidate inferences is searched to see if there is a hypothesized task which acts on a blue unit. Such a task represents something the enemy might be doing, if it can actually be made to work in the current situation. If such a task is found, a new entity is created to represent that task, and SME is re-invoked to mine the analogy further, importing additional information about the task. This additional information includes the other parts of the task (e.g., the other entities involved, such as the attacker and the location and the path) and their properties, including the explanation about why the task succeeded.

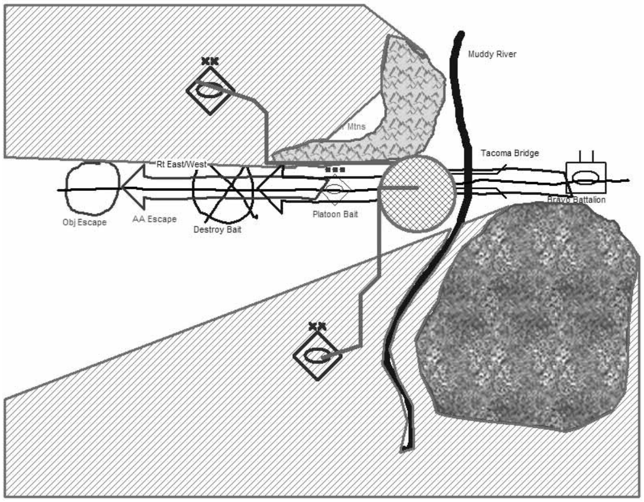

Figure 4: Two possible ambush hypotheses. The pink circle represents the engagement area, the regions represent possible starting locations for Red, and the purple lines indicate hypothetical paths

Once all of the information about the hypothetical task is mined from the analogy, the system must determine if this task is plausible. In the current system, we only take into account spatial constraints, ignoring factors such as relative combat power. Specifically, we solve for the locations and paths involved in the task, to see if we can find positions and a path that satisfy the task's constraints. Each combination of locations and path defines a way for that task to be executed in the current situation. For example, the engagement area for the hypothesized destroy task can be anywhere along the axis of advance for blue, the starting point for Red is a region that cannot be seen by blue, and the path must start at Red's location and end at the engagement area.

We use the spatial reasoning techniques described above to solve these constraints and construct the appropriate positions and path. All consistent solutions found are presented to the user, via a new layer depicting the solution, as shown in Figure 4.

Notice that path-finding is defined with respect to start and end points, whereas the start and end locations were only constrained by regions. Since sketch maps are by nature coarse, we simply use the centroid of a region when necessary, and display both the concrete location and the constraint region. In a performance support application this is a reasonable solution, since accurate optimization can depend on more information than the sketch map has, and once alerted to a general possibility, in our experience users are quick to see improvements. For creating game AIs it will be useful to optimize automatically, e.g., place the division at the northern edge of the mountain and attack from behind.

9. Other Related work

Qualitative spatial reasoning has often focused on mechanical systems (cf. [16,25]), but some have focused on navigation and locations (cf. [22]). None have focused on supporting the kind of complex reasoning that occurs in the military domain. Efforts in the synthetic forces literature start with GIS data rather than sketch maps. While terrain analysis is starting to be used in the computer game industry, the analyses are carried out by hand, typically by annotating maps during level design. Winston [26] was the first to model the use of precedents in supporting reasoning; our system uses a more sophisticated model of analogical reasoning and more complex reasoning to generate results, making it closer to case-based reasoning systems [23].

10. Discussion and Future work

We have argued that sketch maps provide an important arena for qualitative spatial reasoning, using battlespace reasoning as a source of examples. We have described the qualitative spatial representation and reasoning facilities in nuSketch Battlespace, a multimodal interface system that focuses on reasoning rather than recognition. We have shown that these facilities can be combined with analogical reasoning to do a sophisticated task, a subset of enemy intent hypothesis generation.

While these capabilities are a significant advance in the state of the art, much research remains before human-quality spatial reasoning facilities will be achieved. We see three key problems to address: (1) Optimization within constraint solutions, e.g., picking optimal combinations of starting and ending positions and paths. This will be very important for supporting wargaming, where one wants to see how a plan survives the best that an opponent might throw at it. (2) Sketch retrieval, i.e., automatically finding precedents (cf. [21]) to be used in generating enemy intent hypotheses and COAs. We plan to use our MAC/FAC model of similarity-based reminding [14] for this. (3) Moving beyond blob semantics, i.e., using more information about glyph shapes in matching and retrieval. Our shared similarity constraint suggests that shape descriptions need to be guided by results in visual psychology to the extent possible [11].

As these techniques advance, we intend to apply them in three ways. First, we plan on adding more performance support tools to nSB, such as trafficability calculators and COA critiquers, to help users generate better plans. Second, we plan on using it in intelligent tutoring systems for military training. Finally, we plan on providing interfaces to wargame engines, both as a way of providing wargaming for performance support, and as an interface to commercial computer games. Discussions are already underway with several computer game design studios concerning the use of our spatial reasoning techniques in their upcoming games.

Acknowledgements

This research was supported by the DARPA Command Post of the Future and Rapid Knowledge Formation programs, and by a National Science Foundation ROLE grant. We thank our military experts, both active-duty and retired, for their sustained and valuable feedback. Special thanks to

Major Robert Rasch, Major James Donlon, and Major William Turmel for helping design and implement the nuSketch Battlespace interface. John Everett provided useful suggestions about the paper.

References

1. Alvarado, Christine and Davis, Randall (2001). Resolving ambiguities to create a natural sketch based interface. *Proceedings of IJCAI-2001*, August 2001.
2. Cohen, P. R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. (1997). QuickSet: Multimodal interaction for distributed applications, Proceedings of the Fifth Annual International Multimodal Conference (Multimedia '97), (Seattle, WA, November 1997), ACM Press, pp 31-40.
3. Cohn, A. (1996) Calculi for Qualitative Spatial Reasoning. In Artificial Intelligence and Symbolic Mathematical Computation, LNCS 1138, eds: J Calmet, J A Campbell, J Pfalzgraf, Springer Verlag, 124-143, 1996.
4. Coventry, K. 1998. Spatial prepositions, functional relations, and lexical specification. In Olivier, P. and Gapp, K.P. (Eds) 1998. *Representation and Processing of Spatial Expressions*. LEA Press.
5. Davis, I. 2000. Warp Speed: Path Planning for Star Trek: Armada, *AI and Interactive Entertainment: Papers from the 2002 AAAI Spring Symp.*, AAAI Press, Menlo Park, Calif.
6. Donlon, J.J. and Forbus, K. D. Using a geographic information system for qualitative spatial reasoning about trafficability. 13th International workshop on qualitative reasoning (QR99), Loch Awe, Scotland. June 6-9, 1999.
7. Edwards, G. and Moulin, B. 1998. Toward the simulation of spatial mental images using the Voronoi model. In Olivier, P. and Gapp, K.P. (Eds) 1998. *Representation and Processing of Spatial Expressions*. LEA Press.
8. Egenhofer, M. (1997) Query Processing in Spatial-Query-by-Sketch in Journal of Visual Languages and Computing 8(4), 403-424 pp.
9. Falkenhainer, B., Forbus, K., Gentner, D. (1989) The Structure-Mapping Engine: Algorithm and examples. *Artificial Intelligence*, 41, pp 1-63.
10. Feist, M. and Gentner, D. 1998. On Plates, Bowls and Dishes: Factors in the Use of English IN and ON. *Proceedings of the 20th annual meeting of the Cognitive Science Society*
11. Ferguson, R. W. 2000. Modeling orientation effects in symmetry detection: The role of visual structure, Proceedings of the 22nd Conference of the Cognitive Science Society . Hillsdale, New Jersey: Erlbaum.
12. FM 5-33 *Terrain Analysis,* Headquarters, Department of the Army, July 1990.
13. Forbus, K., Ferguson, R. and Usher, J. 2001. Towards a computational model of sketching. *IUI'01*, January 14-17, 2001, Santa Fe, New Mexico
14. Forbus, K., Gentner, D. and Law, K. 1995. MAC/FAC: A model of Similarity-based Retrieval. *Cognitive Science*, 19(2), April-June, pp 141-205.
15. Forbus, K., Mahoney, J.V., and Dill, K. 2001. How qualitative spatial reasoning can improve strategy game AIs: A preliminary report. 15th International workshop on Qualitative Reasoning (QR01), San Antonio, Texas, May.
16. Forbus, K., Nielsen, P. and Faltings, B. "Qualitative Spatial Reasoning: The CLOCK Project", *Artificial Intelligence*, **51** (1-3), October, 1991.
17. Forbus, K. and Usher, J. 2002. Sketching for knowledge capture: A progress report. *IUI'02*, January 13-16, 2002, San Francisco, California.
18. Forbus, K., Usher, J., and Chapman, V. 2003. Sketching for Military Courses of Action Diagrams. *Proceedings of IUI'03*, January, 2003. Miami, Florida.
19. Gentner, D., & Markman, A. B. (1997). Structure mapping in analogy and similarity. *American Psychologist, 52,* 45-56.
20. Glasgow, J., Chandrasekaran, B., & Narayanan, N. (Eds.). (1995). *Diagrammatic Reasoning*. Cambridge Mass.: AAAI Press/The MIT Press.
21. Gross, M. and Do, E. (1995) Drawing Analogies - Supporting Creative Architectural Design with Visual References. in *3d International Conference on Computational Models of Creative Design*, M-L Maher and J. Gero (eds), Sydney: University of Sydney, 37-58.
22. Kuipers, B. 2000. The Spatial Semantic Hierarchy. *Artificial Intelligence* **119**: 191-233.
23. Leake, D. (Ed.) 1996. Case-Based Reasoning: Experiences, Lessons, and Future Directions, MIT Press.
24. Rasch, R., Kott, Al, and Forbus, K. 2002. AI on the Battlefield: An experimental exploration. *Proceedings of the 14th Innovative Applications of Artificial Intelligence Conference*, July, Edmonton, Canada.
25. Stahovich, T. F., Davis, R., and Shrobe, H., "Generating Multiple New Designs from a Sketch," in Proceedings Thirteenth National Conference on Artificial Intelligence, AAAI-96, pp. 1022-29, 1996.
26. Winston, P. 1982. Learning New Principles from Precedents and Exercises," *Artificial* Intelligence **19**, 321-350