

How qualitative spatial reasoning can improve strategy game AIs

Kenneth D. Forbus

Qualitative Reasoning Group,
Northwestern University
1890 Maple Avenue
Evanston, IL, 60201
forbus@nwu.edu

James V. Mahoney

Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA, 94301
jvmahon@parc.xerox.com

Kevin Dill

Qualitative Reasoning Group
Northwestern University
1890 Maple Avenue
Evanston, IL, 60201
kdill@cs.nwu.edu

Abstract

Spatial reasoning is a major source of difficulties for strategy game AIs. We conjecture that *qualitative spatial reasoning* techniques can help overcome these difficulties. We briefly review the relevant qualitative reasoning ideas, and outline four potential advantages of our approach. We describe two explorations in progress: How visual routines can be used to quickly compute qualitative spatial descriptions for war games, and how qualitative descriptions can help in path-finding.

Introduction

Creating good strategy game AIs is hard. Spatial reasoning is a major source of difficulties: Terrain is of vital importance in war games, and geography is key in *Civilization*-style empire/trading games. Since today's strategy AI's are tightly bound to the underlying game world simulation, it is hard to start their development before the game world is up and running, and harder still to reuse the algorithms and representations in a new game, unless the underlying engine is extremely similar to the old engine.

We conjecture that using *qualitative spatial reasoning* techniques [2,3] to build strategy AIs can help overcome these problems. We begin by briefly reviewing the relevant qualitative reasoning ideas, then outline four advantages that we believe can be obtained using this approach. After that, we describe two explorations of this idea that are in progress: How visual routines can be used to quickly compute qualitative spatial descriptions for war games, and how qualitative descriptions can help in path-finding. We close with some thoughts about next steps and a request for collaborators.

What is qualitative spatial reasoning?

Qualitative representations carve up continuous properties into conceptually meaningful units [4,5]. This is very different from, say, chopping up numerical values into a set

of uniformly-sized bins or using a grid to divide up space. Let's start with the simpler one-dimensional case. Think about temperature of a fluid. To understand what state that fluid is in requires knowing what its temperature is relative to its freezing point and boiling point. The specific numerical value (say 72 instead of 73 degrees) is not as relevant as knowing that it is less than its boiling point and above its freezing point. Roughly speaking, temperature can in this circumstance be quantized into five discrete values, two of which are points and the other three of which are intervals. On the other hand, if you are going to immerse yourself in it, its temperature relative to your body temperature becomes relevant, as do your comfort ranges for bathing temperatures. The quantizations that are important for reasoning depend both on physical properties (i.e., freezing point, boiling point) and on task-specific properties (i.e., your desire to bathe in that water).

Qualitative spatial representations work in the same way. Qualitative spatial representations carve up space into regions¹, based on a combination of physical constraints and task-specific constraints. In reasoning about motion, for example, identifying flat versus sloped surfaces is useful because of the different ways that colliding objects can bounce on them. In reasoning about motion, identifying ranges of angles where parts may come in contact versus ranges of angles where they may move freely is one important criterion for carving up orientation. Qualitative spatial representations are almost always firmly rooted in a quantitative, diagrammatic representation, for a variety of technical reasons [2]. For our purposes, this means that we can use quantitative information (e.g., specific coordinates of a unit, or distances between units) in qualitative spatial reasoning. For instance, qualitative spatial descriptions for trafficability support both finding routes in a general way, and carrying out time-distance estimates about travel over those routes (e.g., can they get there in time?). In other words, by identifying and using

¹ We include edges and points, of course, as special cases.

these conceptually meaningful units, reasoning strategies can achieve results that are more human-like.

We believe that the spatial reasoning problems faced by strategy AIs can be better tackled using qualitative spatial representations. For example, one key spatial constraint for strategic reasoning is *trafficability*: The ability of a vehicle or unit to move across a specified piece of terrain [1]. In a *combined obstacle overlay*, US military planners divide an area of operations into *severely restricted*, *restricted*, and *unrestricted* terrain. Severely restricted means impassable, restricted means one can get through, although perhaps slowly or with damage, and unrestricted means that that unit can travel unhindered by that terrain. (We will see an example of this below.) Given a map and an echelon, the area of operations is carved up into regions of these three types. Further analysis of the unrestricted terrain identifies *mobility corridors* (i.e., paths over which sub-echelon units can move between interesting places) and *avenues of approach* (i.e., paths over which that echelon can move, based on the available mobility corridors). These spatial descriptions, and others commonly used by the US military, are good examples of qualitative spatial descriptions that have been evolved through human practice. Commanders and decision-makers use these tools constantly. If we want strategy AIs that can do as well as human players (without cheating), or that can communicate with human players in meaningful ways (e.g., as subordinate commanders or teammates or opponents), we need to exploit these and other qualitative spatial representations.

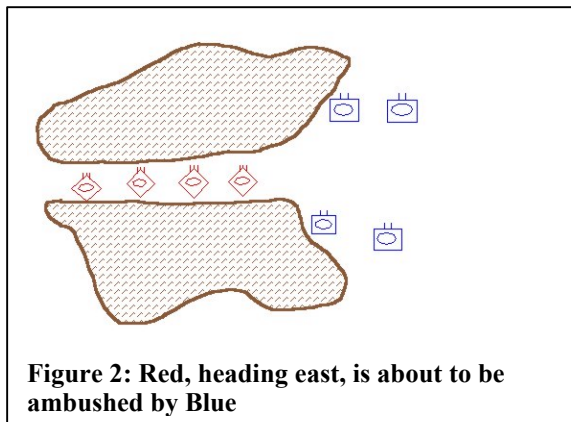


Figure 2: Red, heading east, is about to be ambushed by Blue

Potential advantages

We believe that four advantages could be obtained by using qualitative spatial reasoning in strategy game AI's: *more expressive spatial representations*, *better communication of intent*, *better pathfinding*, *more reusable strategy libraries*. We discuss each in turn.

More expressive spatial representations

Many strategies exploit particular properties of terrain. Being able to describe spatial configurations and properties

is thus essential to expressing strategies in a general way. For example, how can one characterize what are good locations for an ambush? One way of thinking about it is that an ambush is a place where you can focus far more combat power on your opponent than they can on you. One good type of ambush site, illustrated in Figure 1, is a "funnel", where your opponent is coming out of a tightly constrained space into a position where your forces are already arrayed, waiting for them. By providing more abstract descriptions of space, qualitative representations simplify recognizing and exploiting such configurations.

Better communication of intent

Commanders generally sketch when describing their plans.

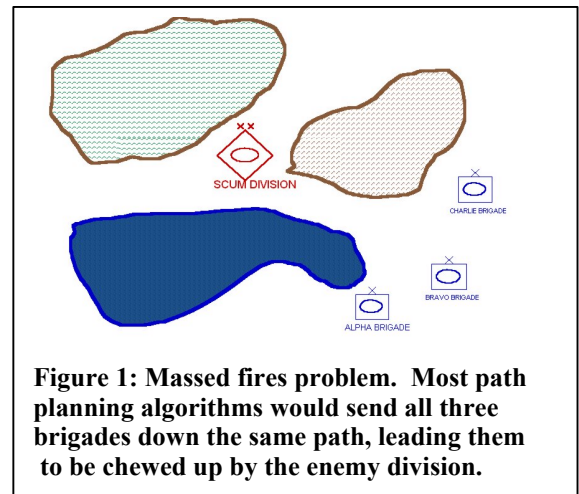
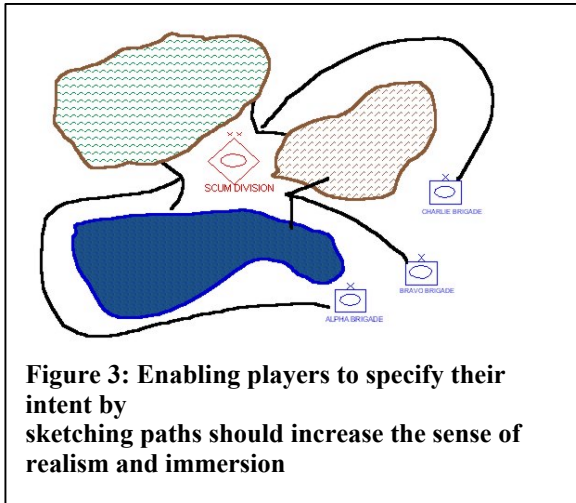


Figure 1: Massed fires problem. Most path planning algorithms would send all three brigades down the same path, leading them to be chewed up by the enemy division.

They specify their intent, so that units can choose their actions to be consistent with the commander's overall goals, rather than just blindly doing the concrete task assigned. They specify timing information and contingencies, so that their subordinates work as a coordinated force. Today's game interfaces, even with innovations such as waypoints and formations, are poor substitutes. Some might argue that forcing players to micromanage their forces is a good way to induce flow and thus enhance engagement in the game world. We believe instead that it leads to player frustration, especially when their units are being unrealistically stupid.

Consider a classic problem in strategy AI: Massed fires. If you assign three units to attack a specific enemy, they will chose paths to get to that enemy and then attack it. Many games are susceptible to the following problem: Suppose the quickest path to the enemy involves a tightly constrained tube, forcing your units to go down it single-file. The enemy, which would have been overwhelmed had your forces converged on it all at once, can then destroy each of them in turn as they enter the clearing (see Figure 2).

Good military planners solve this problem differently.



They can specify paths that the units will take, and can specify synchronization constraints (i.e., “using these axes of advance (see Figure 3), be in position to commence the attack at 0400”). Good communication is essential to good coordination of forces. The bandwidth between human commanders and their subordinate commanders is massively higher than the bandwidth between a human player and the units under his command. Providing higher bandwidth between a human player and the game’s strategy AI’s could, we believe, provide a more satisfying experience.

Better pathfinding

One complaint from game developers during the 2000 Symposium was that path finding was still a significant problem across game genres. We believe that qualitative spatial representations can help with these problems, by reformulating space in ways that are more amenable to search.

Problems with pathfinding seem to run rampant throughout the industry. Even modern, massively successful games can run in to trouble in this area. Diablo II is a good example of an extremely popular game that has difficulty determining where the NPCs should move, and how and when they should get there. These problems are at the very least annoying, and in some instances caused the NPC or even the player to be killed.

Diablo II’s pathfinding is most visible when the player has an NPC follower. It appears that the game attempts to keep the follower fairly close to the player, but not right next to him. The NPC tends to wander continuously within some acceptable area. It seems that the game does not make any attempt to plan the NPC’s path in advance or to consider how to get around obstacles. Instead, when player and follower are separated, the follower tries to move toward the player as directly as possible. In addition, there does

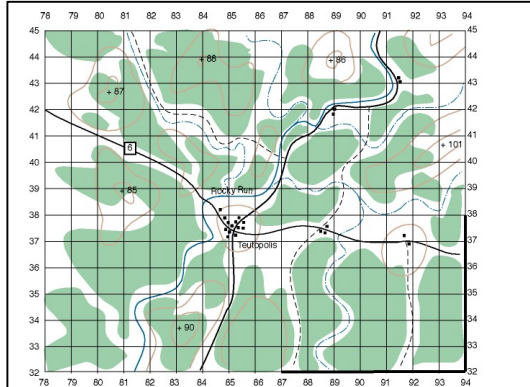
not seem to be any conception of “safe” vs. “dangerous” spaces. The NPC will frequently wander into areas that the player has not yet explored.

Because the pathfinding algorithm doesn’t have an understanding of obstacles and how to navigate around them, it is very easy for the follower to get “caught” in a local concavity (for instance the corner of a room you have exited). Many games have this problem for larger concavities, but it is more noticeable in Diablo II since almost any concavity can cause it. The solution applied by the folks at Blizzard is to magically transport the follower back to the player’s vicinity if they get too far apart. Thus, when the NPC becomes stuck the player has the option of continuing without it for a while, or to go back and get it. Neither option is particularly appealing to most players, in our experience. In addition, if this mechanism were applied to a game which creates a more “realistic” world (such as Baldur’s Gate) players might be much less willing to accept NPC’s that can teleport in this fashion.

The NPC’s inability to discriminate between safe and dangerous spaces can cause the player even more serious problems. When the player stands still (to heal, for example) the NPC tends to wander around him. As a result, it often becomes visible to the enemy, resulting in a fight for which the player may not be ready. There are even times when the player is fighting one group of monsters, and the NPC will wander off and draw more monsters into the fight, resulting in an attack which comes from several directions at once!

Reusable strategy libraries

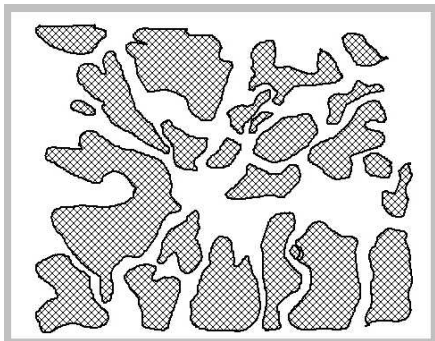
Reusable components are an important way to develop new software. Reusability typically enhances robustness and reduces development time. Unfortunately, today’s strategy AI’s are mired in the ontological choices of the specific game engine that they are developed for. This sharply limits reuse. Consider an alternative. Qualitative spatial representations provide a layer of description above the specifics of the game engine, optimized to express the distinctions needed to carry out its reasoning. Suppose we can automatically compute qualitative spatial representations from the game engine spatial descriptions. Then we can express the strategy AI’s spatial knowledge entirely in terms of the qualitative spatial representations. These representations would represent quantizations relevant for that genre of game (i.e., trafficability for war games with a land-based component), but would not rely on the specifics of the underlying game engine. As long as the “perceptual” component of the strategy AI were updated, the underlying game world could be completely reimplemented. Depending on how general the representations for other aspects of the game world are, the same strategy AI could be used with an entirely different game engine for that genre.



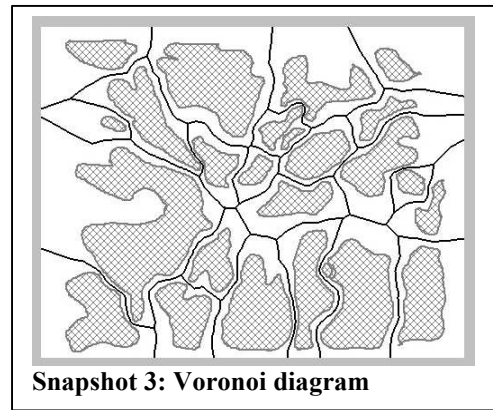
Snapshot 1: Map used in an on-line tactical decision game

Visual computation of qualitative spatial representations

Qualitative spatial descriptions are only going to be useful in games if they can be efficiently and automatically computed from the spatial descriptions in the underlying game world. Many aspects of these descriptions can be computed once for a given map, e.g., basic trafficability constraints due to terrain features. Other properties will need to be dynamically computed (i.e., changes in trafficability due to introduction of obstacles, visibility based on estimated enemy positions, fields of fire based on positions and available equipment). Figuring out the appropriate qualitative descriptions for battlespace



Snapshot 2: Severely restricted terrain, as drawn in the Tactical Decision Coach

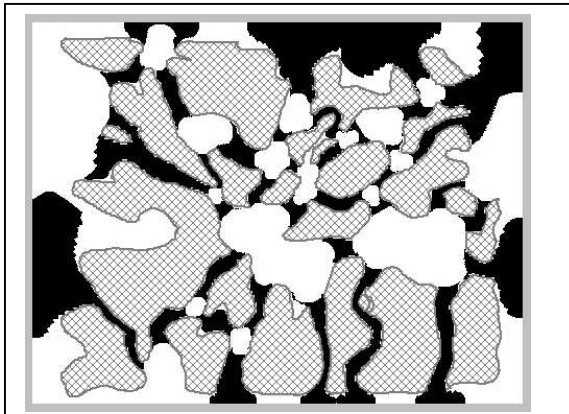


Snapshot 3: Voronoi diagram

reasoning is still very much research in progress. However, based on preliminary experiments on computing qualitative descriptions of free space, we suspect that these computations can be done practically in a gaming environment. This section summarizes those preliminary experiments.

Our work is based on Mahoney's MAPS approach to visual routines. MAPS provides a computational model of high-level visual processing that exploits operations on bitmaps, organized into a high-level functional language that supports rapid prototyping of visual operations. We are experimenting with visual operations for extracting qualitative descriptions of free space, expressed in terms of areas and paths connecting them. We will illustrate the process via an example. Snapshot 1 shows a map used in an electronic *tactical decision game*, a computer-facilitated on-line game where players take on the roles of unit commanders, responding to orders from a designated commander. Snapshot 2 shows the severely restricted regions (for armor), given that map. (These regions were sketched in via our multimodal interface system, since the map was given to us in the form of an uninterpreted bitmap. Such regions can easily be automatically extracted from either ink-based or bitmap based models in game engines [1].)

The next step in characterizing free space is to compute the Voronoi diagram, which identifies lines and points that are equidistant from a pair of obstacles (see Snapshot 3). The Voronoi diagram provides a starting point for characterizing free space. Regions of free space can be identified by growing out from the junctions of the Voronoi diagram, and paths are identified as the subsets of the edges of the diagram that are not subsumed by free regions. The results of these computations are show in Snapshot 4.



Snapshot 4: Free space regions and corridors

This description of free space and corridors still needs work – for example, there are edge effects where the distinctions between regions and paths seem visually unnatural. However, we do find it encouraging, given that we have only just started exploring this space of algorithms. From a practical standpoint, it is important to note that these computations only need to be done once per map. The entire sequence of computations described here took six seconds on a mid-range machine, using a Java-based general-purpose implementation. We believe that this is already fast enough for most purposes, and that substantial further performance optimizations could be done if necessary.

Using qualitative spatial representations in pathfinding

We are exploring two uses of qualitative reasoning in pathfinding. First, we believe that we can improve the performance of pathfinding algorithms by better describing the space in which pathfinding occurs. Second, we would like to be able to add soft constraints to pathfinding.

More efficient pathfinding

Effective pathfinding requires solving two problems: Describing the space and searching that description for a suitable path. Much effort has been put into developing search techniques such as A*. Much less effort has been put into creating better descriptions of space to search through.

Historically, there are two schools of thought about how to describe space for purposes of path planning: *cell decomposition* and *skeletonization* [6,7]. Cell decomposition, in its simplest form, simply involves overlaying the world with a grid. Each square in the grid is marked as trafficable or non-trafficable, and then path planning is done on the grid. There is, obviously, a tradeoff between grid resolution and search speed. More

advanced methods have been proposed that can deal with shaping the edges of cells to the edges of obstacles, or combining groups of small cells into a single large cell. Skeletonization attempts to reduce the world to a 1 dimensional skeleton along which units may move. Pathfinding is then reduced to finding a path to the skeleton, finding a way to move along the skeleton to a point near the goal, and then finding a way to move from the skeleton to the goal.

The method we are pursuing involves a hybrid of these two techniques. Using the qualitative spatial representations described above, we divide space into *tubes* (which can be used to get through constricted areas) and *open areas* (convex spaces in which we can move directly from any edge point to any other edge point). Concave areas can be handled either by subdividing them into smaller convex areas, or by moving along the wall as needed to get around the concavities. We believe that dividing space in this way will result in significantly more concise descriptions of space, allowing our search routines to more quickly produce a good path. Additionally, the resulting path should look fairly natural, eliminating the need to use post-processing techniques for smoothing.

Soft constraints in pathfinding

Military planners consider at least three distinct sources of constraints when deciding how to move between two points. These are *trafficability*, *visibility*, and *fields of fire*. Trafficability, as noted earlier, concerns the ease with which a unit can move along the path. Visibility describes how easily the unit can be seen by enemy units, and how well it can see the enemy (often, but not always, there is a tradeoff between the two aspects of visibility). Fields of fire are the areas in which the unit may be fired upon by enemy units. Most computer games appear to consider only trafficability. In the Diablo II example given above, for instance, the NPC's failure to consider its visibility is often significantly detrimental to the player.

If we can qualitatively divide space based on these constraints, we should be able to devise AIs which act far more cleverly. They will be able to exploit terrain to hide from the player, figure out weak spots in the player's defenses, and generally give a better game, without cheating, than today's game AI's. Furthermore, we can add an element of personality to our NPCs by setting their utility for these constraints differently (or we can allow the player to set this for NPCs they control).

Discussion

In this paper we have described work in progress on using qualitative spatial reasoning to improve AIs for strategy games. The central idea is that, by using more human-like representations of space, we can build strategy AIs that can

use terrain better in achieving their goals, take orders better (for subordinate commanders), and find their way around better. Moreover, by decoupling the representation of space in the strategy AI from the implementation of space in the game engine, we could take a large step towards making reusable strategy AIs, driving down development costs while improving their subtlety.

While we are working on applying these ideas in computer games, the main driver of our qualitative battlespace representation and reasoning work is the creation of intelligent software to support US military needs. For example, we are creating a *Tactical Decision Coach* that uses analogical reasoning to critique student battle plans, based on comparisons with expert solutions to the same tactics problems. This coach is a first step towards software assistants and advisors that could be part of future command post software. By working closely with military experts, we aim to develop systems that will be capable collaborators with human commanders. However, we also expect that what we learn about strategy and tactics will be applicable in turn to strategy games. It is no accident that military thinkers tend to be gamers (although often not computer gamers); simulation games provide an invaluable forum for experimentation and learning.

We are currently proceeding in this research in several ways. First, we are continuing to experiment with the MAPS system, to create qualitative spatial descriptions to support path-finding and deeper terrain analysis (e.g., spotting good places for ambushes). Second, working with Cycorp, we are expanding representations of human activities so that effects of plans can be more realistically considered (e.g., if you force troops to march harder they will get tired sooner). Third, we are fielding-testing alpha versions of our software to US military personnel and military experts, using their formative feedback to help drive our development process.

We are very interested in collaborating with strategy game designers to explore how these ideas can be used in games. As we improve our representations, we need to move beyond tactical decision problems and into interleaved planning and execution. One roadblock in doing this is the lack of good military simulations with APIs that would enable us to create software that could play on an equal footing with humans. (Existing military simulations, such as MODSAF, require human operators and substantial computing resources for anything but the simplest scenarios.) We would be very interested in collaborations with game developers who already have, or are willing to make available to us, such games with APIs.

Acknowledgements

This research was supported in part by DARPA under the Command Post of the Future program. We thank Major Robert Rasch (US Army), Captain Bill Turmel (US Army),

Brigadier General Keith Holcomb (USMC, retired) and Colonel Michael Heredia (US Army) for valuable discussions and insights about tactics and military planning. We thank Will Wright and Ian Davis for valuable discussions and insights about the state of the art in game design.

References

1. 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.
2. Forbus, K., Nielsen, P. and Faltings, B. "Qualitative Spatial Reasoning: The CLOCK Project", *Artificial Intelligence*, **51** (1-3), October, 1991.
3. Forbus, K. 1995. Qualitative Spatial Reasoning: Framework and Frontiers. In Glasgow, J., Narayanan, N., and Chandrasekaran, B. *Diagrammatic Reasoning: Cognitive and Computational Perspectives*. MIT Press, pp. 183-202.
4. Forbus, K. "Qualitative Reasoning". *CRC Handbook of Computer Science and Engineering*. CRC Press, 1996.
5. Kuipers, B. (1994). *Qualitative Reasoning: Modeling and simulation with incomplete knowledge*. Cambridge, Mass.: MIT Press.
6. Latombe, Jean-Claude, *Robot Motion Planning* (c) 1991, Kluwer Academic Publishers, Norwell MA
7. Russell, Stuart J and Norvig, Peter, *Artificial Intelligence: A Modern Approach*, (c) 1995, Prentice Hall, Upper Saddle River, NJ