Intelligent Visualization in a Planning Simulation

Robert St. Amant, Christopher G. Healey, Mark Riedl,
Sarat Kocherlakota, David A. Pegram, and Mika Torhola
Department of Computer Science
North Carolina State University
Box 7534
Raleigh, NC 27695-7534
stamant@csc.ncsu.edu

ABSTRACT

This paper describes a set of visualization techniques for interactive planning in a physical force simulation called AFS. We have developed a 3D environment in which textures are overlaid on a simulated landscape to convey information about environmental properties, agent actions, and possible strategies. Scenes are presented, via automated camera planning, such that some simple agent goals can be induced visually with little effort. These two areas of visualization functionality in AFS exploit properties of human low-level and intermediate-level vision, respectively. This paper presents AFS, its visualization environment, and studies we have run to explore the relationship between AFS visualizations and the high-level planning process.

1. INTRODUCTION

In mixed-initiative systems, users collaborate with an automated assistant to generate and carry out different courses of action. Effective collaboration between a human planner and an AI system requires that the participants work in areas where they perform best, use appropriate representations for communication, and effectively acquire and transfer authority for planning tasks [1, 5]. A number of mixed-initiative systems have been developed in planning and natural language processing research (e.g., TRAINS [11], TRIPS [12], COLLAGEN [23], AIDE [24, 25]), and significant progress has been made on abstract models of mixed initiative (e.g. [7]). Nevertheless, although the broad outlines of the area are gradually becoming better understood, basic questions about user interaction techniques for mixed-initiative assistance remain open.

The term *mixed-initiative assistance* covers a wide range of potential activities, including providing timely information for situation assessment, helping users focus on critical problem areas, making suggestions about appropriate actions, and handling plans and actions delegated by the user. Most systems developed to date have relied heavily on natural language in interacting with users, an appropriate choice for many situations. Our interest, in contrast, lies in a direct manipulation interaction style, that associated with conventional graphical user interfaces.

IUI'01, January 14-17, 2001, Sante Fe, New Mexico.

Direct manipulation techniques combined with graphical data presentation (which we will refer to here simply as GUI techniques) dominate modern interactive software. The purported benefits of GUIs, in comparison to other styles of interaction, include reduced error rates, faster learning and better retention, and facilitation of exploratory behavior. GUI interfaces gain these benefits by offering users a structured, predictable environment: like real-world objects, static software objects remain static over time; environmental response to a given action is the same if the action is repeated under the same conditions; actions are usually taken at the user's direction and pace, rather than those of the environment; the environment does not initiate activity, but rather only responds to user actions [25]. These properties reduce the space of user decisions to a more manageable level (e.g., time pressure, uncertainty, and environmental instability, including exogenous events, are abstracted away.) Unfortunately, the same properties that help users solve problems working alone also limit the role of an assistant. How can an assistant contribute effectively to the problem-solving process if it is not allowed, for example, to take visible actions that the user may not yet have thought of?

We believe that GUI techniques can contribute to the interaction between a user and an intelligent assistant, but that some concessions must be made in the design of the assistant. Our approach has been to emphasize the visual communication abilities of an assistant, so that it can use the visual GUI environment to guide and sometimes constrain the potential actions of the user. Our work on mixed-initiative assistance has focused on strategic, physical planning problems in AFS, an abstract force simulator [2, 22]. AFS is a general-purpose simulation system that supports experimentation with interactive planning techniques and their relationship to physical processes. AFS incorporates an assistant that works behind the scenes to generate plans potentially helpful to the user. A visualization interface presents these plans and their supporting information to the user by graphical means, relying as much as possible on visual techniques rather than language (i.e., text, symbols, or even iconic conventions) for communication. Our goal is not to build an intelligent, collaborative assistant that relies solely on direct manipulation and graphics (though intelligent rooms and ubiquitous computing research suggest that this is at least conceivable) but rather to gain a better grasp of the relationship between GUI techniques and mixed-initiative assistance. We believe that a better understanding of this relationship may lead to improved problem-solving performance and increased user acceptance of intelligent assistants.

Although our work arises from research on planning in the user interface, it can also be seen as a form of intelligent visualization, a staple of research in the intelligent user interfaces community.

Intelligent visualization researchers have built systems for automatic explanation generation, intelligent tutoring, and other tasks [10, 20, 8, 9, 14, 3, 4], relying on many of the same sources we use. The requirements for our work differ in some ways from these efforts, however: the assistant and the user observe an external process, each able to guide it but without complete control; camera manipulation is viewed as a means of explicit communication, rather than only a mechanism for visual orientation and focus of attention; problem-solving is reactive and opportunistic, with no extended narratives involved; neither party acts in a fixed role in the interaction (e.g., user as commander or student, assistant as tutor or information provider.) Similarities to existing systems will be obvious nevertheless.

The remainder of this paper is structured as follows. The AFS section describes the simulation, which provides a concrete setting for interaction issues relevant to mixed-initiative planning. We describe the visualizations AFS produces and explain their relationship to models of human visual processing. The experimentation section discusses a study of user consistency in associating visual features with specific physical interpretations; this consistency will eventually be exploited by AFS to convey the planning intentions of the assistant in an unobtrusive, natural-seeming fashion in the physical domain. In the conclusion, we tie this work to our ongoing efforts to formalize the concept of affordances [13] in the user interface.

2. AFS

AFS provides a physical domain in which abstract agents (which we alternatively call "agents," "force units," or simply "blobs") can interact, based generally on Newtonian physics [2]. Units and inanimate objects have mass, size, and shape; they may be solid or permeable; they move with variable friction over a domain-dependent surface; they apply force to one another, causing damage/mass reduction.

In AFS's Capture the Flag (CTF) domain, two teams of force units move over a terrain, their travel constrained by mountains, water, and forests. Each team is responsible for defending a set of stationary flags. A team successfully completes a scenario by destroying all the members of the opposing team or capturing all of its flags. Figure 1 shows a sample scenario, in a birds-eye or plan view. In this domain, as in all AFS domains, force units rely on a small set of primitive physical actions: they may *move* from one location to another and *apply-force* to other units and objects such as flags. These actions can be specialized and combined in various ways to form higher-level strategies, such as blocking a pass, encircling a flag, attacking an opponent in a group, and so forth. A hierarchical planner at the center of the system provides plan execution and monitoring.

Ordinary interaction with AFS is via direct manipulation. The user can direct agents by selecting them and assigning to them either low-level actions or higher-level plans. One role of the assistant is to interpret the strategic situation in the unfolding simulation, to inform the user of significant relationships or events, and to suggest ways of dealing with them, in order to help the user make informed decisions Visual communication toward these ends is carried out by two means: texture-based visualizations and scenario-based camera planning. We discuss each type in turn.

2.1 Scenario-based camera planning

The 3D version of the AFS environment, shown in Figure 2, allows the user to navigate via an "eye in hand" interface. The

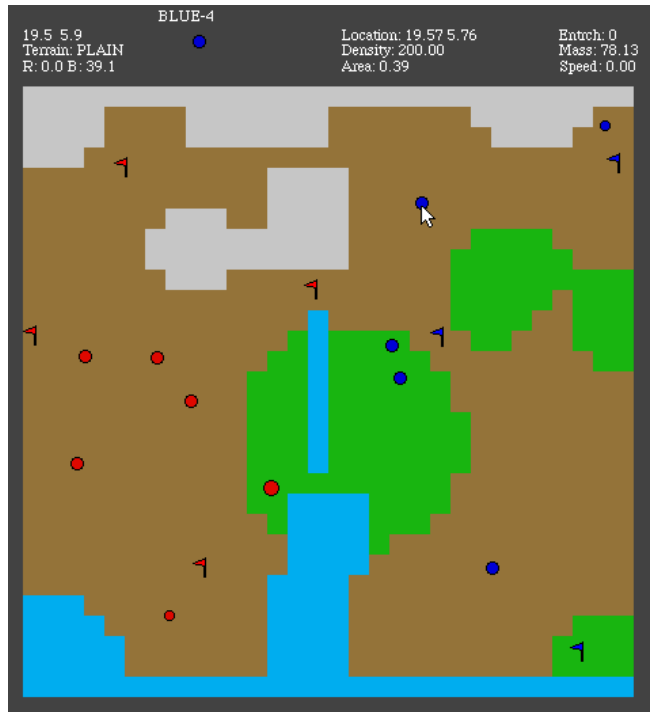

Figure 1. AFS plan view

user is free to view the scenario from any position and angle. The visualization interface can also manipulate the camera autonomously with a camera-planning module. In particular, the camera can be positioned by the system to present the scenario such that certain user actions are more readily evaluated and executed than others. The design of the camera planner takes a situated approach to problem solving and visualization. The system attempts to position the camera, and thereby situate the embodied user, to present a particular set of affordances to the user in the environment. In this way, the AFS system can lead the user into taking specific actions.

Our approach treats camera planning as a form of numerical constraint satisfaction. Constraints are placed on what the resulting visualization should look like, and the camera planner

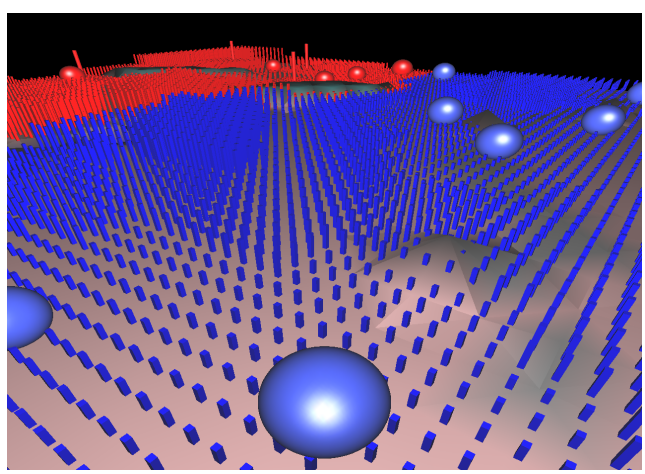

Figure 2. Camera planning result

must meet those constraints to place the camera in the optimal position. Inconsistencies cause conflicts between constraints, which are resolved by relaxing one of the constraints to some degree. Based on the high-level goals assigned to the force-units in the scenario, a set of visualization constraints are fed into the camera planner. The camera planner uses these constraints to evaluate possible camera positions as it performs a depth-limited hill-climbing search. Some of the constraints that can be passed into the camera planner are given below. Most are related to conveying a suggestion that one agent should attack a specific opponent or capture a flag.

- *In-scene*: Whether the agents and objects participating in the action are visible.
- *Centered-vertical*: Whether the agents and objects are centered vertically in the scene.
- *Background*: Whether "distractor" agents are visible.
- *Attack-angle*: How closely the "optimal" view is met: from an attacking agent to the center of a target object.
- *Agent-occlusions*: Whether any agents obscure participating agents or objects.
- *Terrain-occlusion*: Whether uneven terrain (e.g., a mountain range) obscures agents or objects.
- *Too-low*: Whether the angle of view is too low to show sufficient context.
- *Viewing-distance*: How closely a specific, constant viewing distance is matched.

These constraints are represented by heuristic functions that execute in sequence to evaluate every candidate camera position during the search. The constraints return numerical values and the camera planner satisfices with the smallest total score. To reduce the size of the search space, we approximate the solution algorithmically and select that as a starting point. The constraints are ordered by importance and weighted so that more important constraints are less likely to be relaxed before less important constraints. For example, *terrain-occlusion* is ranked high in importance because it is essential that specified objects are visible to the user. The weight values for each constraint were determined experimentally.

A sample visualization is shown in Figure 2. The assistant has smoothly moved the camera from its previous position to the current one, to convey the advice that the blue force unit in the foreground should attack the group of red flags immediately in front of it. (The suggestion is much more apparent on a normal-sized display than in the reduced figure.) AFS can also generate visualizations for comparing opponent agents, to help the user evaluate a potential suggestion. Further camera-planning visualizations have been designed and user-tested with paper diagrams, but these have not yet been implemented. The assistant currently operates under the restriction that the visible scene may not be modified to improve a visualization (e.g., simply removing irrelevant agents, flags, or landscape features). As we come to a better understanding of the capabilities and limitations of the system under this restriction, it may be relaxed.

Our design of the camera planning module is based in part on a visual routines model of intermediate vision [29]. This can be most easily seen when the system generates a visual comparison of two opposing forces: it moves to a point between the forces, at an appropriate distance, and arranges that they are seen with their lowest points touching a common horizontal line. It is straightforward to break this down into elemental operations such as setting markers and extending rays, which combine into routines, to allow for an efficient and accurate size comparison. The geometric computations that drive constraint satisfaction in the camera planner are not visual routines, but they are intended to share some of the same necessary functionality.

2.2 Texture-based visualizations

Our work on texture-based visualization examines another component of visual processing, the low-level human visual system. When we look at an image, certain visual features can be identified very quickly, without the need for search. These features are often called preattentive, because their detection precedes focused attention in the low-level human visual system [27, 30]. Preattentive features include visual properties like color, brightness, orientation, size, and motion. When applied properly, these features can be used to perform exploratory data analysis. Examples include searching for data elements with a unique feature, identifying the boundaries between groups of elements with common features, tracking groups of elements as they move in time and space, and estimating the number of elements with a specific visual feature. Preattentive tasks are performed very rapidly and accurately; they can often be completed in a "single glance" of 200ms or less. The time required for task completion is furthermore independent of display size; users can increase the number of data elements in a display with little or no increase in the time required to analyze the display.

Our research focuses on identifying such findings in the vision and psychophysical literature, then extending these results and integrating them into a visualization environment. To date, we have compiled an interlocking collection of results on the use of color (hue and luminance) [15] and texture (size, density, and regularity) [16, 17] for multidimensional visualization. These results have been used to visualize a number of real-world applications including medical scans [26], weather tracking [16, 17], and scientific simulations [19].

In our AFS work, we have applied these findings to the presentation of strategic, spatially distributed information to assist the user in making planning decisions. The design of our visualizations in AFS and the work cited above centers around the concept of a perceptual texture element, or *pexel*. Pexels are graphical icons that collectively convey color and texture information: hue, luminance, size, density, and regularity, among other possibilities. Pexels appear in the visualization in Figure 3 as small vertical strips of color standing on end over the landscape. In this visualization, for example, the height of a pexel represents the shortest time it will take any agent to reach a given location; the color of a pexel corresponds to the team of that agent; increased density is associated with target regions containing flags or opponents. As a simulation unfolds, the user sees the local colors and heights of the pexel field spread and change; it becomes immediately obvious when a red or blue flag is enveloped by a pexel field of the opposite color, indicating that it is in danger of capture. Patterns such as boundaries between regions of different color, density, and height can be determined at a glance, providing potentially useful strategic information.

This is not intelligent assistance in any significant sense; however, the ability to manipulate the association between textures and strategic information in the simulation does give an assistant an important communicative tool.

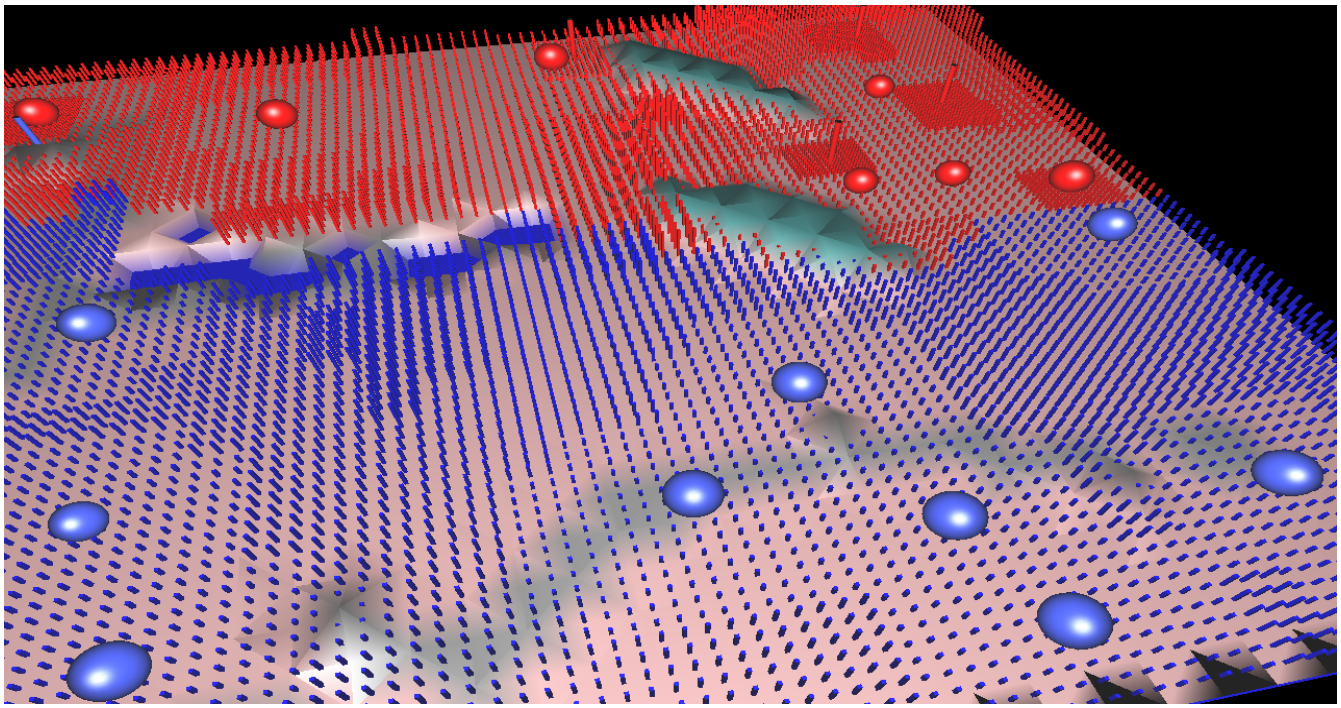

Figure 3. Texture-based visualization

3. EXPERIMENTATION

We gain some significant advantages in relying on relatively low-level perceptual mechanisms in our visualization techniques—speed, high volume, multi-dimensional data integration through texture manipulation, effective (though very limited) recognition of simple agent plans by assumptions about visual routines. One of the drawbacks of the approach, however, is that this interaction takes place below the cognitive level—that is, in AFS we have adopted the traditional view of planning as search through a problem space, with states represented in symbolic terms. Although users are able to extract properties of the visualizations efficiently, there is no necessary relationship between these properties and abstract concepts relevant to planning problems.

A straightforward solution is to rely on conventions for visual representation: the height of a pexel corresponds to some plan or situation assessment variable *x*, its hue, luminance, density, and so forth to other variables. These relationships must be learned by the user before the visualization can be interpreted. We have experimented with a different approach, however, one especially well suited to the physical planning domain. Texture fields such as the ones shown in Figures 2 and 3 can be viewed as abstractions for conveying information, but that can also be interpreted in physical terms. It is possible to see the pexel field as a field of grass, for example, or other, similar visual texture-producing ground cover. We naturally associate such textures with our physical interaction with it; that is, we are attuned to its affordances [13]. If users consistently relate specific visual texture properties to specific physical properties, such as ease of movement, direction, or speed, then AFS might exploit this relationship to convey physical planning suggestions in visual terms, without depending on the user's learned knowledge of display conventions.

To explore this issue, we conducted an experiment based on artificial visualization scenarios. Our 20 subjects were students and interns, both men and women, working at North Carolina State University. Their ages varied between 20 and 30. Each subject was presented with a sequence of snapshots consisting of a 3D field of red pexels surrounding a blue ball (which would represent an agent or blob in AFS), as shown in Figure 4. Values of blob radius, pexel density, and pexel height, which we will refer to as the variables Radius, Density, and Height, were varied

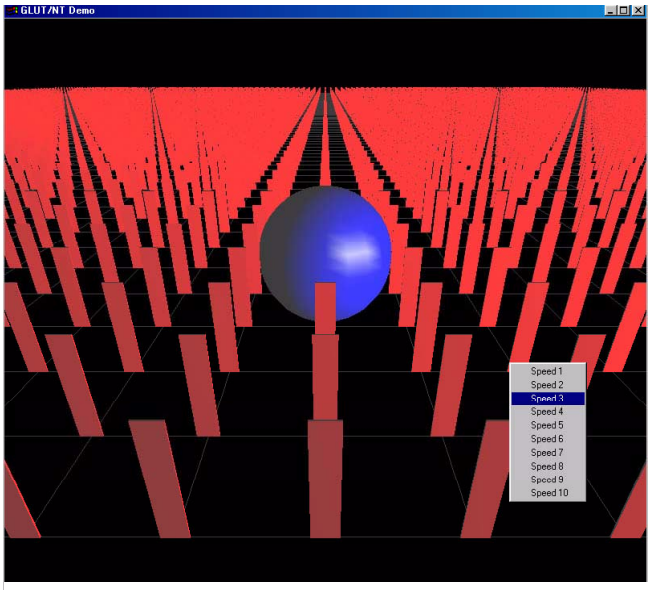

Figure 4. Sample experiment trial

across the snapshots. For each snapshot, the subject was asked, "If the ball were to be rolled across the field shown, how fast would it move?" A discrete set of choices was available from a pop up menu, ranging from Speed 1, the lowest, to Speed 10, the highest. Subjects were allowed to experiment with a few different snapshots before starting the experiment proper, in order to develop an internal calibration of speeds for the textures they would see.

Because of the small number of variables we chose to examine, a full factorial design was possible. Radius alternated between 0.60 and 0.75 units; Height ranged among four values of 0.25, 0.40, 0.55, and 0.75 units; Density ranged similarly among four values of 0.25, 0.40, 0.55, and 0.75 units. The units of measurement here are unimportant; the specific values were chosen after prototyping and testing by the experimenters. Thirty two combinations of these values are possible (2 x 4 x 4); each subject saw every combination three times, in randomized order. The data resulting from the experiment consisted of a Subject identifier, the specific values of Radius, Density, and Height for each snapshot, the Speed selected, and the Duration of the user's selection action.

An analysis of variance shows that all factors have a significant influence on the mean value of speed (for Radius $F=26.48$, $p<0.0001$; Density, $F=51.71$, $p<0.0001$, Height, $F=700.05$, $p<0.0001$.) Subjects most strongly associated Height with Speed, followed by Density and then by Radius.

Figure 5 shows projections through the experimental dataset. In the top graph, for example, we collected all the values for each combination of Subject and Height, merging the different values of Radius and Density, and computed the mean of each partition. For each subject we then see four values, shown as marks on a line, that represent the average speed of the blob for each of the four possible pexel height values: 0.25, 0.40, 0.55, 0.75. The other two graphs are constructed analogously for Radius and Density. We discarded the first subject's results because of procedural irregularities, but found significant patterns of similarity among the remaining subjects. In general, subjects judged that the higher the pexel field, the slower a blob will be able to move. (Subject 2's results are consistent with a reversal of the magnitude of speed choices; talking with subjects afterwards we found this to be a minor source of confusion for others as well.) Except for subjects 19 and 20, a comparable pattern holds for Radius: larger blobs are judged to be able to move faster than smaller ones. Finally, a more complex and slightly unexpected pattern holds for Density. For most subjects, higher pexel density is associated with higher speed, which corresponds to a physical interpretation in which the blob rolls over the field rather than through the individual pexels For a few subjects (e.g., 6, 12, 13), however, the lowest density affords faster movement as well, producing a U-shaped relationship between Density and Speed.

One last finding was a significant effect of Subject on Speed ($F=9.80$, $p<0.0001$). In combination with the observations above,

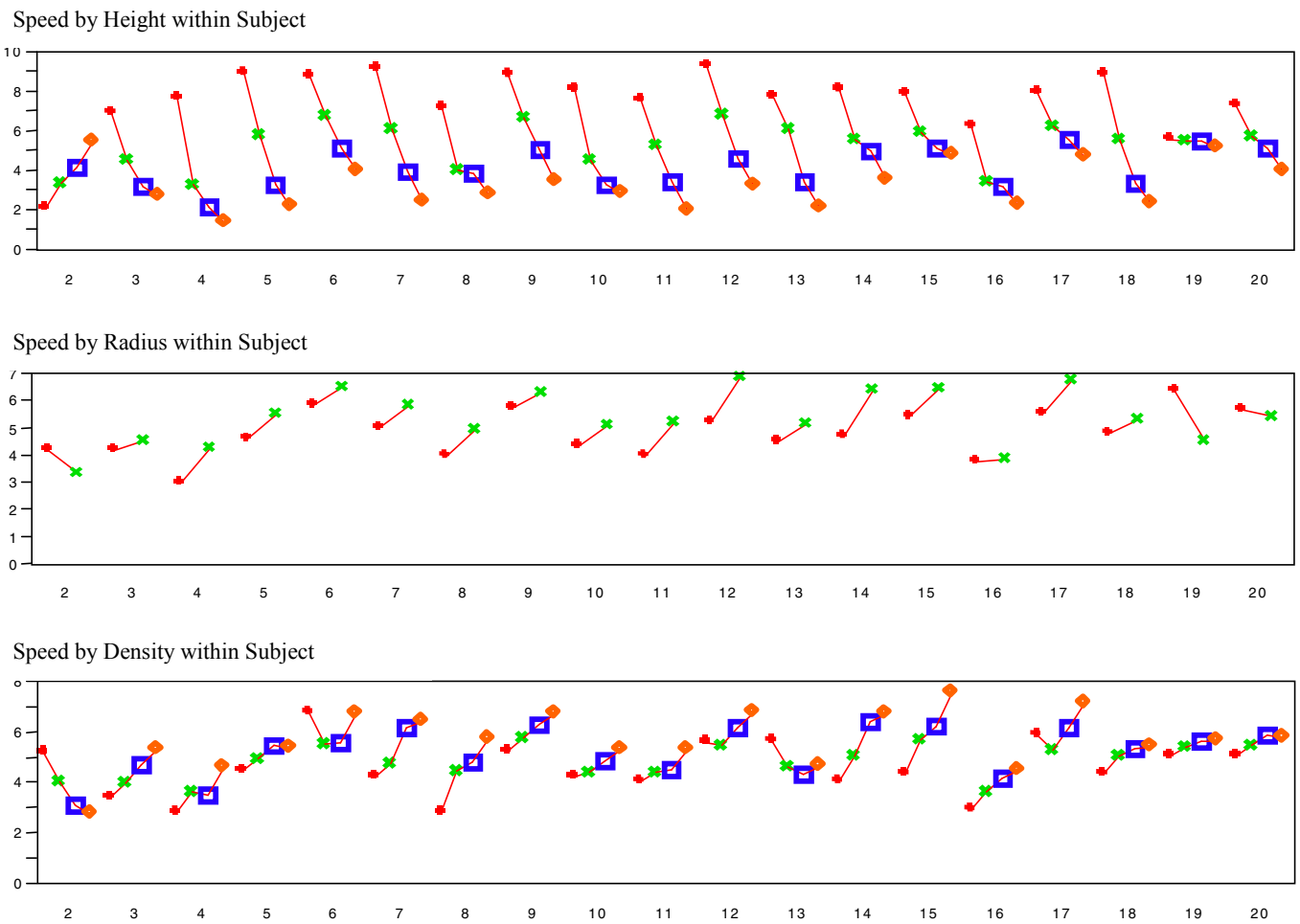

Figure 5. Subject consistency in visual interpretation

this suggests that subjects interpret the visualization textures in different ways (a small number of different ways, but still different.) This means that we cannot depend on different users having a single unified interpretation of AFS texture-based visualizations in physical terms. If this is desirable, some initial guidance is necessary (e.g., showing different scenarios with the preferred interpretations of combinations of size, radius, density, and speed) to ensure that users adopt appropriate interpretations.

This exploratory study is a small step in our research. With experimentally validated relationships of this kind, we are extending the current visualization assistant in AFS to convey suggestions about agent assignments and the strategic value of terrain locations by generating appropriate textures. For example, the assistant currently does path planning for force units, and can draw such paths in the plan view of the simulation to show its suggested courses of action. Textures offer a less obtrusive solution: the assistant can map lower height pexels over the regions over which force units can move more safely, or possibly to greater strategic effect, without forcing an obvious choice on the user. We are currently running a comparable study of user inference of direction of movement based on the orientation of pexels; we will continue by designing more formal experimental studies of the relative associations between texture features and physical behavior. We hope to identify consistent relationships between visual textures and physical properties such as speed and direction, but also accessibility, safety, vantage point utility, among others.

4. DISCUSSION

Our work in this area grows out of an interest in situated problem-solving and affordances in the user interface [25]. Situated problem solving differs from more conventional forms of problem-solving. When embodied within an environment, how one perceives the environment and what is perceptible within the environment is of paramount importance to what actions one chooses to make. The environment itself provides cues—affordances—about what appropriate actions there are, but these cues must first be perceived and understood within the context of one's relationship with elements within the environment. For example, the brink of a gulch may be perceived as affording falling off of, but if one is moving towards the brink at a high velocity, the brink may suddenly be perceived to afford leaping across [28]. Likewise, one cannot make use of tools within the environment if they are hidden from view or out of reach.

In our view, the process of situated problem solving is iterative in nature, involving three stages: *perception*, *intention*, and *action*. The first stage, *perception*, involves using the senses to determine possible actions that can be made in the environment. This process involves registering the affordances the environment provides for action. The second stage, *intention*, involves taking the results of perception and choosing the best action that will advance the organism towards a given goal. The final stage, *action*, occurs when intent is transformed into behavior by interacting directly with the environment [21]. Actuators in the environment, including the organisms own motor control, are activated in order to bring the organism closer to achieving its goal. Actuators affect the environment and the organism's relationship with the environment, resulting in a new situation. The cycle iterates, starting with perception of the environment and how it has been changed by the previous cycle. New intentions are formed and carried out.

Intentions are formed when there is a mismatch between the goals internal to the organism and the state of the external world [21]. Before intentions can be formed, the organism must sense its own situation within the surrounding environment: it must become aware of the environment's current state as well as its relationship with the environment. Needless to say, the perceptual senses play an important role in situated problem solving. The sensory apparatus, however, cannot be merely instruments for recording sensory stimuli; they must proactively interpret and transform the sensory stimuli into affordances.

The perception of affordances is primarily a cognitive interaction with perceptual stimuli. Affordances do not exist without an organism to perceive them. By presenting plans three-dimensionally, viewed from a camera that is allowed to move anywhere within the playing-space, we call on the metaphor of embodiment within a 3D environment. By embodiment we mean that the user interacts with the virtual 3D environment as if he were present in the environment at the location of the camera. With the user embodied in a 3D virtual environment, we are able to make use of the same problem-solving strategies one uses when interacting with the real-world environment.

5. REFERENCES

1. Allen, J. F. 1994. Mixed initiative planning: Position paper. [WWW document]. *ARPA/Rome Labs Planning Initiative Workshop.*
2. Atkin, M.; Westbrook, D. L.; Cohen, P. R.; and Jorstad, G. D. 1998. AFS and HAC: Domain-general agent simulation and control. *AAAI-98 Workshop on Software Tools for Developing Agents*, pp. 89-95.
3. Bares, W. H., and Lester, J. C. 1997. Realtime generation of customized 3D animated explanations for knowledge-based learning environments. *Proceedings AAAI*-97, 347-354.
4. Bares, W. H., and Lester, J. C. 1999. Intelligent multi-shot visualization interfaces for dynamic 3D worlds, *Proceedings IUI-99*, 119-126.
5. Burstein, M. H., and McDermott, D. V. 1996. Issues in the development of human-computer mixed-initiative planning. In *Cognitive Technology: In Search of a Humane Interface*, pp. 285-303. Elsevier.
6. Chapman, D. 1992. Intermediate Vision: Architecture, Implementation, and Use, *Cognitive Science* 16(4):491-537.
7. Cohen, R., Allaby, C., Cumbaa, C., Fitzgerald, M., Ho, K., Hui, B., Latulipe, C., Lu, F., Moussa, N., Pooley, D., Qian, A., and Siddiqi, S. 1998. What is Initiative? *User Modeling and User-Adapted Interaction* 8(3/4): 171-214.
8. Drucker, S., and Zelter, D. 1994. Intelligent camera con-trol in a virtual environment. In *Proceedings Graphics Interface '94,* 190-199.
9. Drucker, S., and Zelter, D. 1995. Camdroid: A system for implementing intelligent camera control. *Proceedings 1995 Symposium on Interactive 3D Graphics*, 139-144.

10. Feiner, S. K., and McKeown, K. 1991. Automating the generation of coordinated multimedia explanations. *IEEE Computer* 24(10):33-41.

11. Ferguson, G., and Allen, J. 1996. Trains-95: Towards a mixed-initiative planning assistant. *Proceedings Artificial Intelligence Planning Systems*, pp. 70-77.

12. Ferguson, G., and Allen, J. 1998. Trips: An intelligent integrated problem-solving assistant. *Proceedings AAAI-98.*

13. Gibson, J. 1979. *The Ecological Approach to Visual Perception.* Houghton Mifflin.

14. He, L. W., Cohen, M. F., and Salesin, D. H. 1996. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. *Proceedings SIGGRAPH '96*, 217-224. ACM Press.

15. Healey, C. G. 1996. Choosing effective colours for data visualization, *Proceedings Visualization '96*, pp. 263–270.

16. Healey, C. G., and Enns, J. T. 1998. Building perceptual textures to visualize multidimensional datasets, *Proceedings Visualization '98*, pp. 111–118.

17. Healey, C. G., and Enns, J. T. 1999. Large datasets at a glance: Combining textures and colors in scientific visualization, *IEEE Transactions on Visualization and Computer Graphics* 5(2):145–167.

18. Healey, C. G., Booth, K. S., and Enns, J. T. 1995. Real-time multivariate data visualization using preattentive processing, *ACM Transactions on Modeling and Computer Simulation* 5(3):190–221.

19. Healey, C. G. 1998. One the use of perceptual cues and data mining for effective visualization of scientific datasets, *Proceedings Graphics Interface '98*, pp. 177–184.

20. Karp, P., and Feiner, S. 1990. Issues in the automated generation of animated presentations. In *Proceedings Graphics Interface '90*, 39-48.

21. Kugler, P.N. and Shaw, R.E. 1990. Symmetry and symmetry-breaking in thermodynamic and epistemic engines: A coupling of first and second laws. In H. Haken & M. Stadler (eds.), *Synergetics of cognition,* pp. 296-331. Springer-Verlag.

22. Pegram, D. A., St. Amant, R., and Riedl, M. O. 1999. An approach to visual interaction in mixed-initiative planning. *AAAI-99 Workshop on Mixed-Initiative Intelligence*.

23. Rich, C., and Sidner, C. L. 1998. COLLAGEN: a collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction* 8(3/4).

24. St. Amant, R., and Cohen, P. R. 1998. Interaction with a mixed-initiative system for exploratory data analysis, *Knowledge-Based Systems* 10(5): 265–273.

25. St. Amant, R. 1999. User interface affordances in a planning representation. *Human Computer Interaction*.

26. Tam, R., Healey, C. G., and Flak, B. 1997. Volume visualization of abdominal aortic aneurysms, *Proceedings Visualization '97*, pp. 43–50.

27. Triesman, A., and Gelade, G. 1980. A feature-integration theory of attention, *Cognitive Psychology* 12:97–136.

28. Turvey, M. T., and Shaw, R. E., 1979. The primacy of perceiving: An ecological reformulation of perception for understanding memory. Perspectives on memory research, L. G. Nilsson (ed.), pp. 167-222. Lawrence Erlbaum.

29. Ullman, S. 1996. *High-level vision*. Cambridge, MA: The MIT Press.

30. Wolfe, J. M. 1994. Guided Search 2.0: A revised model of visual search, *Psychonomic Bulletin & Review* 1(2):202–238.