Hansen, S.R., & Narayanan, N.H. (2000). On the Role of Animated Analogies in
Algorithm Visualizations. In B. Fishman & S. O'Connor-Divelbiss (Eds.),
Fourth International Conference of the Learning Sciences (pp. 205-211). Mahwah, NJ: Erlbaum.

On the Role of Animated Analogies in Algorithm Visualizations

Steven R. Hansen* and N. Hari Narayanan+
*Modeling & Simulations Dept., Air Command & Staff College
Maxwell AFB, AL 36112, USA
+Department of Computer Science & Software Engineering
Auburn University, Auburn, AL 36849, USA
Tel: (334) 844-4330, Fax: (334) 844-6329
Email: narayan@eng.auburn.edu

Abstract: If a "picture is worth a thousand words," then why have attempts over the past decade to use pictures and animations to replace or supplement traditional instructional methods for teaching algorithms produced such disappointing results? In an earlier paper (Hansen, Schrimpsher, & Narayanan, 1998) we described a research project based on the premise that a rethinking of algorithm animation design is required in order to harness its power to enhance learning. The key insight was that for algorithm animations to be effective, they had to be "chunked" and embedded within a context and knowledge providing hypermedia information environment. In this paper, we report on ablation studies which were designed to discover which aspects of the prototype hypermedia visualization system that was developed (called HalVis) contributed to student learning. These preliminary studies led to a surprising discovery that interactive and animated analogies appear to significantly prime learning about abstract and dynamic algorithm behaviors from subsequent visualizations. We first present the interactive features and learning modules of HalVis. Two ablation experiments conducted on HalVis are then described. This is followed by a discussion of the results, their implications and how these are shaping our future research.

Keywords: visualization, learning environments, quantitative assessment, and cognitive science

Introduction

Over the past decade, numerous studies and experiments have been conducted to test whether graphical animations of algorithm behavior would improve student learning of this dynamic, abstract, and difficult subject (e.g., Byrne, Catrambone, & Stasko, 1996; Stasko, Badre, & Lewis, 1993). While the pictures and animations are enthusiastically received by students, these studies have not proven conclusively that algorithm animations actually improve learning (see Hundhausen, 1997 for an excellent survey).

We believe that previous attempts at using algorithm animations as learning tools were unsatisfactory not due to any flaw with animation as a technique, but because of the approach used to convey information using the animations. Animations by themselves, even when accompanied by some textual feedback and interactive control, may not be enough. In an earlier paper (Hansen, Schrimpsher, & Narayanan, 1998), we presented a research project aimed at rethinking algorithm animation design. The key insight was that for algorithm animations to be effective, they had to be "chunked" and embedded within a context and knowledge providing hypermedia information environment. The prototype visualization system we designed and built is called HalVis (**H**ypermedia **Al**gorithm **Vis**ualizations). Two empirical studies (described in the earlier paper) demonstrated that students using HalVis significantly outperformed those learning from a traditional textbook. Later, a third experiment compared learning about an algorithm from HalVis to learning from a compilation of the best algorithm descriptions and illustrations (generated by examining 19 textbooks published between 1974 and 1997) *and* subsequently solving a set of typical problems on the algorithm. In this case no significant differences were found between the group of students who learned by interacting with HalVis and the group which perused the printed materials *and then* solved several exercise problems, though the HalVis group had a higher post-test mean score. Two subsequent experiments compared learning by interacting with HalVis against learning from a typical classroom lecture, and learning from an algorithm animation typical of prior research on this topic (Stasko, 1997). In both cases, statistically significant performance improvement was found for the groups of students working with HalVis. Thus, five empirical studies (summarized in Hansen, Schrimpsher, & Narayanan, 1999; 2000; more details available in Hansen, 1999; Hansen, Schrimpsher, Narayanan, & Hegarty, 1998) involving 133 undergraduates have thus far demonstrated that the

HalVis system significantly improves student learning.

Next, we turned our attention to investigating the source of the learning benefits of HalVis. It is not a mere animation presenter in the mold of prior algorithm animation research prototypes. In some ways, its architecture resembles that of the algorithm animation system described in (Recker et al. 1995) containing multiple representations in addition to the animation itself. Following the design principles elucidated in (Narayanan & Hegarty, 1998), HalVis incorporates four unique design features.

1. Providing three kinds of animations to illustrate different views of algorithm behavior - an animated analogy that illustrates the operational characteristics of an algorithm, a micro-level animation that focuses on details of the algorithm's behavior on small data sets, and a macro-level animation that shows the algorithm's aggregate behavior on large data sets.
2. Embedding animations within a hypermedia visualization that also employs textual descriptions, audio narratives and static diagrams to provide contextual information.
3. Presenting animations in discrete chunks accompanied by explanations of the specific actions being accomplished.
4. Encouraging student participation by allowing rich interactions with the animations and using probes or questions to stimulate critical thinking.

Therefore, the question arises as to which of these features can the learning benefits of HalVis be attributed to. We chose to employ ablation studies, in which a full version of a system is experimentally compared against ablated versions with one or a group of its components removed in order to understand the effect of the deleted component(s), as the experimental method to address this question. This paper reports on results from two such experiments we carried out to help answer this question. These studies led to a surprising discovery that interactive and animated analogies appear to significantly prime learning about abstract and dynamic algorithm behaviors from subsequent visualizations. We first present the interactive features and learning modules of HalVis. Two ablation experiments conducted on HalVis are then described. We conclude with a discussion of the implications of the results of these experiments.

Architecture of HalVis

HalVis is structured as three primary modules and two secondary modules. The primary modules are called Conceptual View, Detailed View and Populated View. The secondary modules are called Fundamentals and Questions. Fundamentals is a module that provides illustrated and (in some cases) animated explanations of basic algorithmic concepts such as complexity, recursion and iteration. This module is accessible from all of the primary modules via hyperlinks. For example, if the primary modules are presenting a recursive algorithm, all appearances of the word "recursion" in these modules will appear as underlined hyperlinks to the corresponding descriptions and depictions in the Fundamentals module. The Questions module provides a set of questions that ask the learner to reorder steps of the algorithm and to answer several multiple-choice questions. The system provides immediate feedback as to whether the student's answers are correct or not. The Fundamentals module is intended to provide basic knowledge and context for understanding the algorithm being presented by the primary modules. The Questions module provides the learner with an opportunity to assess his/her understanding of the algorithm. These two modules were not the focus of the studies described in this paper.

The Conceptual View (CV) introduces a specific algorithm in very general terms using a real world analogy. The analogy is interactive (students can manipulate it), animated, and accompanied by text and audio explanations. Each analogy highlights the central conceptual elements of the corresponding algorithm. For instance, the analogy for the MergeSort algorithm shows animated playing cards, which are divided and merged to create a sorted sequence of cards. Figure 1 shows frames from the animated analogy for the QuickSort algorithm. The analogy used in this case is that of a coach ordering a set of players in terms of their height, using a series of moves illustrated through smooth animation, that capture the essence of the QuickSort algorithm in terms of its fundamental operations. The "Show Me" buttons provide students with interactive control over this animated analogy.

The Detailed View (DV) is a module that takes the multiple representation approach to presenting detailed information about how the individual steps of an algorithm, when executed in sequence, manipulate data and accomplish a specific task (e.g., sorting a set of numbers). It presents four windows containing information that is synchronously updated as the simulated execution of the algorithm proceeds on an input data set. The Execution

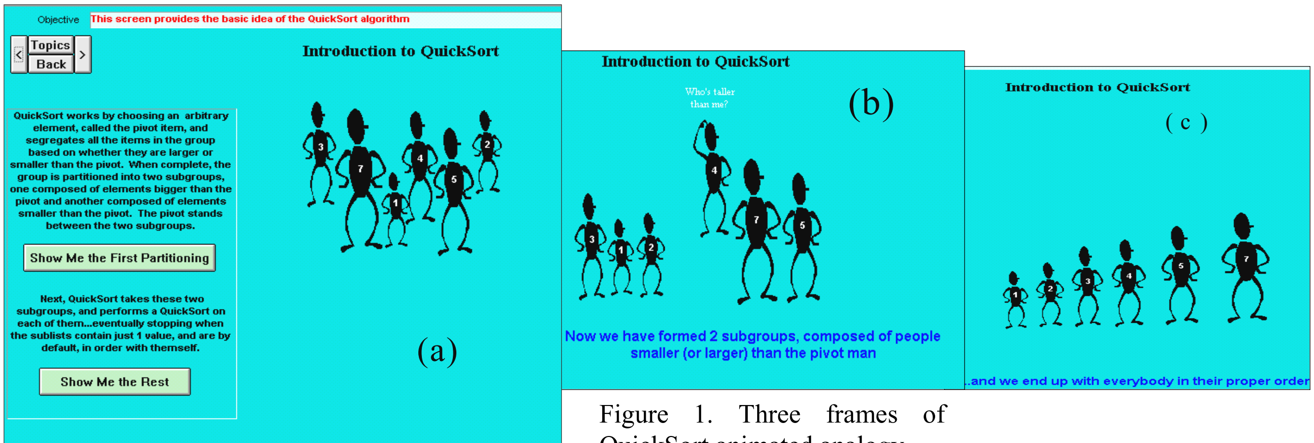

Figure 1. Three frames of QuickSort animated analogy.

Animation window shows how steps of the algorithm modify data structures using smooth animation. This window shows a micro-level animation of the algorithm's operations. The Execution Status Message window provides comments and textual feedback to the student about key events and actions occurring in the animation. This is also available as an audio commentary. The Pseudocode window shows the detailed steps or operations of the algorithm. The step that is being shown by the animation at any moment is highlighted in synchrony with the animation. Finally, the Execution Variables window contains a scoreboard-like panorama of the variables involved in the algorithm and their changing values. Figure 2 contains one frame of the micro-level animations in DV for the QuickSort algorithm, showing examples of these four windows. The Execution Animation Window appears on the

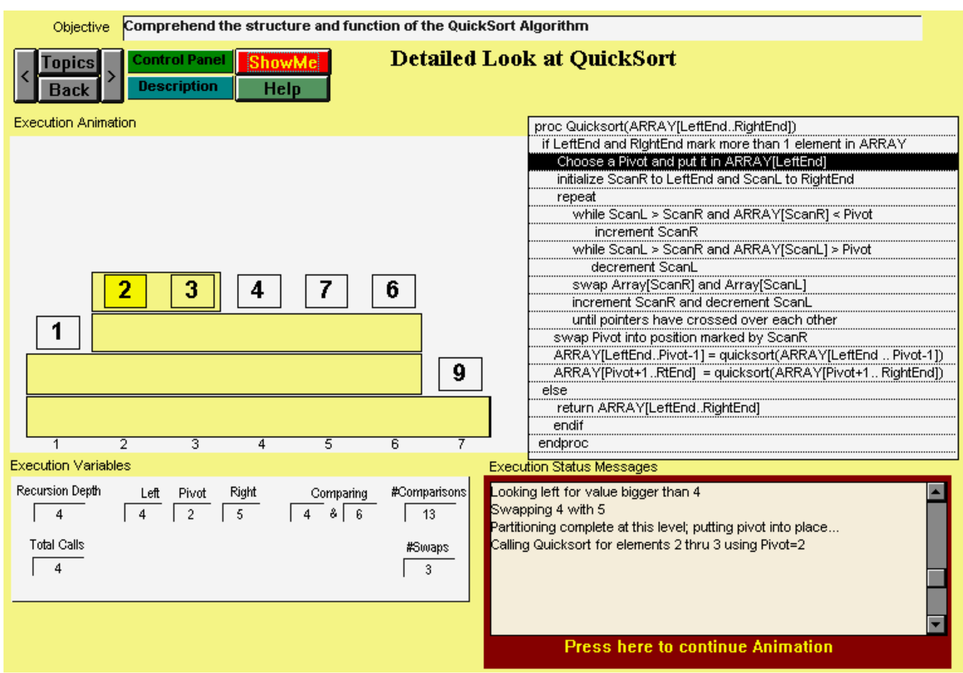

Figure 2. One frame of the QuickSort DV animation.

upper left (below the control buttons) of this figure, the Pseudocode window appears on the upper right, the Execution Variables window appears on the lower left, and the Execution Status Message window appears on the lower right.

The Populated View (PV) module presents a macro-level animated view of the algorithm's behavior on large data sets. A novel feature of this module is a facility for the student to make predictions about different parameters of algorithm performance, and then compare those against the actual performance when the animation is running. A sample screen of this module appears in Figure 3.

Ablation Experiments

Five empirical studies involving 133 students (Hansen, Schrimpsher, Narayanan, & Hegarty, 1998) demonstrated, with statistically significant results, the learning advantages of HalVis when compared to (1) learning from a typical textbook, (2) learning from lectures, and (3) learning from an algorithm animation typical of current research. This, coupled with the fact that the main point of departure of HalVis from algorithm animations designed by other researchers is that it is not a "pure" one-shot animation system, led us to ask the question: Which individual or combinations of features are producing the observed learning benefits?

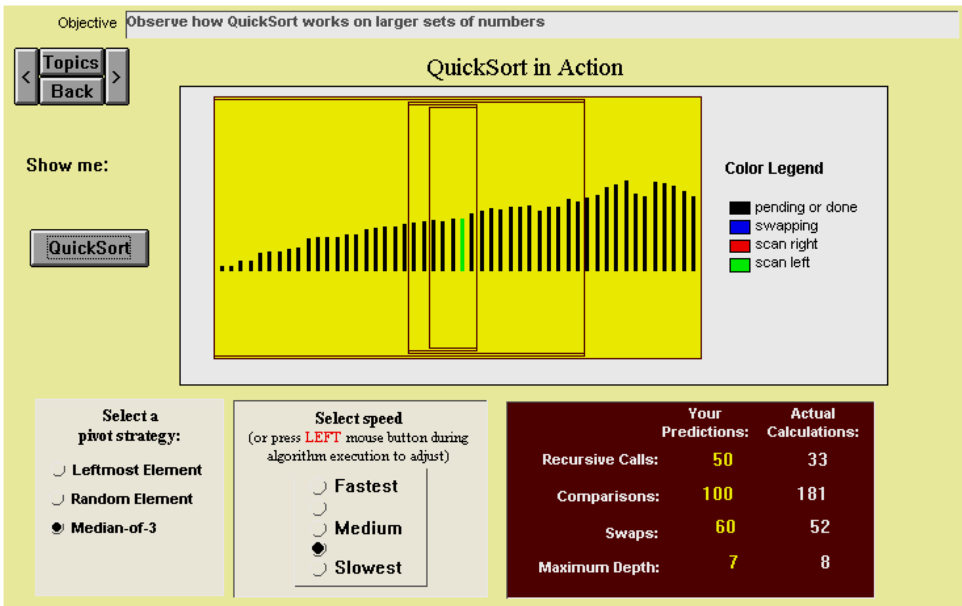

Figure 3. One frame of the QuickSort PV animation.

One approach to answering this is to build different versions of HalVis by selectively eliding individual or groups of related features, and to experimentally evaluate these versions against the original. We designed and carried out two such ablation studies.

Experiment I Procedure

The first experiment involved 32 undergraduate computer science students enrolled in a third year algorithm analysis course at Auburn University. Four matched groups of students interacted with one of four versions of HalVis illustrating the QuickSort algorithm – a complete version (CDP version: all three views) and three elided versions (CD version: Populated View removed; CP version: Detailed View removed; and DP version: Conceptual View removed). A pre-test/post-test combination measured individual learning and improvement with questions that probed conceptual and procedural knowledge about the algorithm. Students were tested on their ability to recognize and reorder pseudocode descriptions of algorithms, mentally simulate algorithmic operations, and predict resulting data structure changes. Each group received a brief, navigation-only orientation to the version of HalVis they were to use, then were assigned to a computer and instructed to interact with the visualization until they felt they understood the algorithm. No time limits were imposed for the tests or the visualization.

Experiment I Results

Our hypothesis was that the most important view was the most information rich view – the Detailed View. Therefore, the three groups that interacted with this view would outperform the group that was denied this view. We also expected that the Populated View would follow in significance and that the contribution of the Conceptual View would be the least. Figure 4 shows the average group improvement. As expected, the group that received all three HalVis views performed the best. However, these results also reveal a surprising impact of the Conceptual View. The groups that performed best were not the ones exposed to the Detailed View, but rather the groups that interacted with the Conceptual View. The improvements of the groups that received the Conceptual View with another view were more than twice the improvement of the DP group. Table 1 shows a statistical summary of this experiment, presenting data for pairwise comparisons between each of the groups with access to the Conceptual

View and the one group –DP – that did not have this access. Note that all comparisons yielded statistically significant results.

Perhaps the most noteworthy observation from the results of this ablation study was the effect of the Conceptual View in priming the learning of information presented in subsequent views. The groups that interacted with the Conceptual View in any combination with other views performed better than the group that lacked the Conceptual View. The impact of the Conceptual View was examined further in the next study that elided two views at a time.

Table 1. Statistical Summary for Experiment I.

CDP	55%	CP	45%	CD	46%
DP	21%	DP	21%	DP	21%
F(1,14)	16.71	F(1,16)	8.99	F(1,15)	7.16
p	$p < 0.001$	p	$p < 0.01$	p	$p < 0.018$

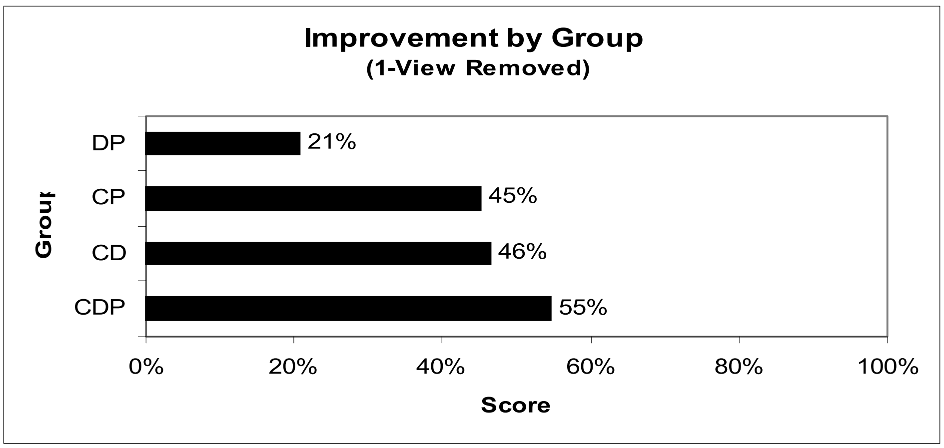

Figure 4. Group improvements for experiment I.

Experiment II Procedure

This experiment was designed to isolate each of the three views of our algorithm visualization framework to measure their respective impact and effectiveness. Its procedure was identical to that of experiment I. Twenty-seven undergraduate students enrolled in a third year algorithm analysis course at Auburn University participated in four matched groups. The CDP group worked with a full version of HalVis illustrating a graph algorithm to find shortest paths between nodes. The C group worked only with the Conceptual View of this algorithm. The D group worked only with the Detailed View of this algorithm. The P group worked only with the Populated View of this algorithm.

Experiment II Results

Our hypothesis was that the Detailed View would prove to be the most valuable because of the amount of information it provided. We were uncertain how impacts of the other views would get ranked, since neither the Populated View nor the Conceptual View contained the volume or depth of information available in the Detailed View. Figure 5 shows average improvements observed in each of the groups. As expected, the CDP group outperformed the others, followed closely by the D group. Interestingly, the C group outperformed the P group by 21%. It is illuminating to note how well the C group did with the limited amount of information that they received.

The importance of the Detailed View was confirmed, as was the value of the Conceptual View. We were surprised at the level of improvement observed in the group that only interacted with the Conceptual View, and this suggests that having a good analogy can produce surprisingly positive results. The performance of the Populated View group lagged behind the others. Yet the interaction logs we collected revealed that in each of the experiments, the animation in this view was executed about the same number of times as the animations in the other two views. Perhaps this serves as confirmation that some animations are merely ‘candy for the eyes’ in that they are entertaining to observe but not particularly informative. It should also be noted that the animation in the Populated View closely resembles algorithm animations developed by other researchers. Interestingly, students generally commented most favorably about the animation in the Populated View. It appears that they thought they were learning more from the Populated View than was actually the case.

Table 2 summarizes the various statistical comparisons between the full-version group and groups with individual views, and between groups with individual views. Except for the CDP:D comparison, the results are all statistically significant.

Table 2. Statistical Summary for Experiment II.

CDP	87%	CDP	87%	CDP	87%
C	57%	D	77%	P	36%
F(1,11)	14.51	F(1,11)	2.55	F(1,11)	51.82
p	p<0.003	p	p<0.14	p	p<0.00001
C	57%	D	77%	C	57%
D	77%	P	36%	P	36%
F(1,12)	5.82	F(1,12)	28.29	F(1,12)	5.35
p	p<0.033	p	p<0.0002	p	p<0.039

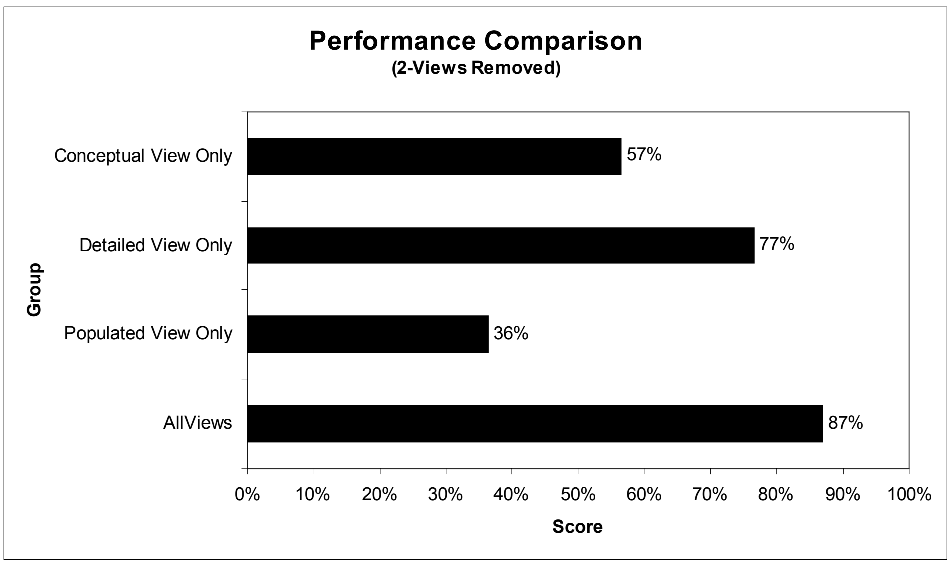

Figure 5. Group improvements for experiment II.

Conclusion

There exists a significant body of psychological literature on the role of analogies in communicating a target concept (e.g., Gentner 1989). Researchers have also investigated the educational value of paired interactive simulations as analogies which complement each other (Brophy & Schwartz, 1998). However, we believe this is the first time that the role of animated analogies in understanding computer algorithms from visualizations has been empirically studied. The power of animated analogies in this domain arises from the fact that these provide conceptual bridges between familiar scenarios and abstract components of algorithms. It has been noticed that students tend to employ analogies in describing how algorithms operate (Douglas, Hundhausen, & McKeown, 1995; Stasko, 1997) and analogies can serve to provide a form of scaffolding (Hmelo & Guzdial, 1996) for subsequent learning. Animations and visualizations on the desktop and in virtual reality are increasingly being proposed (e.g., Salzman et al. 1998) as tools of educational technology. In this context, results reported here lead us to speculate that concrete dynamic analogies can prime students to learn better from visualizations of abstract dynamic phenomena. If confirmed by additional experiments, this has important implications for learning technology. Therefore, our future research is aimed at seeking additional empirical evidence regarding this conclusion.

Analogies in HalVis have four important features: *dynamics* (analogies are animated), *fidelity* (to central characteristics of the algorithm), *interactivity*, and *familiarity* (analogies involve real-world objects and actions). Finer grained ablation studies are being planned to examine the differential effects these features have on the ability of analogies to prime subsequent learning. Additional ablation studies looking at the other features – animation chunking, animation embedding and rich interactivity – are also on the drawing board.

References

Byrne, M., Catrambone, R., & Stasko, J.T. (1996). *Do algorithm animations aid learning?* (Technical Report GIT-GVU-96-18). Graphics, Visualization & Usability Center, Georgia Institute of Technology, Atlanta, GA.

Brophy, S. P., & Schwartz, D. L. (1998). Interactive Analogies. *Proceedings of the International Conference of the Learning Sciences* (pp. 56-62), AACE.

Douglas, S., Hundhausen, C., & McKeown, D. (1995). Toward empirically-based software visualization languages. *Proceedings of the Annual Symposium on Visual Languages* (pp. 342-349), IEEE.

Gentner, D. (1989). The mechanisms of analogical reasoning. In S. Vosniadou & A. Ortony (Eds.), *Similarity and Analogical Reasoning*. Cambridge University Press.

Hansen, S. R. (1999). *A framework for animation-embedded hypermedia visualization of algorithms.* Unpublished doctoral dissertation, Department of Computer Science & Software Engineering, Auburn University, Auburn, AL.

Hansen, S. R., Schrimpsher, D., & Narayanan, N. H. (1998). Learning algorithms by visualization: A novel approach using animation-embedded hypermedia. *Proceedings of the International Conference of the Learning Sciences* (pp. 125-130), AACE.

Hansen, S. R., Schrimpsher, D., & Narayanan, N. H. (1999). From algorithm animations to animation-embedded hypermedia visualizations. *Proceedings of ED-MEDIA'99: World Conference on Educational Multimedia, Hypermedia & Telecommunications*, AACE.

Hansen, S. R., Schrimpsher, D., & Narayanan, N. H. (2000). Helping learners visualize and comprehend algorithms. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning, 1(3)*, To appear.

Hansen, S. R., Schrimpsher, D., Narayanan, N. H., & Hegarty, M. (1998). *Empirical studies of animation-embedded hypermedia visualizations* (Technical Report CSE98-07). Department of Computer Science & Software Engineering, Auburn University, Auburn, AL.

Hmelo, C. E., & Guzdial, M. (1996). Of black and glass boxes: Scaffolding for learning and doing. *Proceedings of the International Conference of the Learning Sciences* (pp. 128-134), AACE.

Hundhausen, C. D. (1997). *A meta-study of software visualization effectiveness.* Unpublished comprehensive examination paper, Department of Computer Science, University of Oregon, Eugene, OR. Available at http://www.lilt.ics.hawaii.edu/%7Ehundhaus/writings/MetaStudy.pdf.

Narayanan, N. H., & Hegarty, M. (1998). On designing comprehensible interactive hypermedia manuals. *International Journal of Human-Computer Studies*, *48*, 267-301.

Recker, M., Ram, A., Shikano, T., Li, G., & Stasko, J. (1995). Cognitive media types for multimedia information access. *Journal of Educational Multimedia & Hypermedia, 4(2/3)*, 183-210.

Salzman, M., Dede, C., Loftin, R. B., & Ash, K. (1998). Using VR's frames of reference in mastering abstract information. *Proceedings of the International Conference of the Learning Sciences* (pp. 249-255), AACE.

Stasko, J.T. (1997). Using student-built algorithm animations as learning aids. *Proceedings of the SIGCSE Technical Symposium on Computer Science Education* (pp. 25-29), ACM.

Stasko, J.T., Badre, A., & Lewis, C. (1993). Do algorithm animations assist learning? An empirical study and analysis. *Proceedings of INTERCHI'93 Conference* (pp. 61-66), ACM.

Acknowledgments

This research has been generously supported by the National Science Foundation under contracts CDA-9616513 and REC-9815016 to Auburn University. We gratefully acknowledge the advice of Mary Hegarty, Department of Psychology, University of California at Santa Barbara, who serves as an expert consultant on experiment design and data analysis to this project.