Empirically Evaluating the Use of Animations to Teach Algorithms

Technical Report GIT-GVU-94-07

Andrea W. Lawrence

Computer Science Department
Spelman College
Atlanta, GA 30314

E-mail: alawrenc@etta.auc.edu

Albert N. Badre
John T. Stasko

Graphics, Visualization, and Usability Center
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280

E–mail: {badre, stasko}@cc.gatech.edu

Abstract

As algorithm animation systems become more widely available and easy to use, instructors must decide whether to utilize these systems to assist their teaching. Although these systems have generated excitement and interest in both teachers and students, little empirical evidence exists to promote their use. This article describes a study involving the use of algorithm animations in classroom and laboratory settings. Results indicated that allowing students to create their own examples in a laboratory session led to higher accuracy on the post-test examination of understanding of the algorithm as compared to students who viewed prepared examples or no laboratory examples.

EMPIRICALLY EVALUATING THE USE OF ANIMATIONS TO TEACH ALGORITHMS

Andrea W. Lawrence
Computer Science Department
Spelman College
Atlanta, GA 30314
alawrenc@etta.auc.edu

Albert Badre
John Stasko
Graphics, Visualization and Usability Center
Georgia Institute of Technology
Atlanta, GA 30332

Abstract

As algorithm animation systems become more widely available and easy to use, instructors must decide whether to utilize these systems to assist their teaching. Although these systems have generated excitement and interest in both teachers and students, little empirical evidence exists to promote their use. This article describes a study involving the use of algorithm animations in classroom and laboratory settings. Results indicated that allowing students to create their own examples in a laboratory session led to higher accuracy on the post-test examination of understanding of the algorithm as compared to students who viewed prepared examples or no laboratory examples.

Keywords: algorithm animation, software visualization, computing education, passive/active learning, empirical studies, TANGO

1 Introduction

Currently, several systems exist which allow the creation of animated versions of computer algorithms [Bro88, Sta90, Glo92, Bro91, SK93]. These systems have varying approaches and varying levels of user control. Some even allow the use of three dimensional effects [SW93, BN93] or the use of sound [BH92]. The purpose of these systems is to allow the dynamic visual representation of programs or algorithms. Such representations may be used to appeal to the power of the human visual system. Since computer programs may be unclear in textual format, it is hoped that the graphical animated format will aid understanding. Consequently, many of these systems have been used as instructional aids.

The use of these animations in instruction raises many questions, as does the incorporation of any new instructional aid. One question which has arisen frequently is whether these animations would be superior to transparencies in a lecture presentation. Another question is whether students will excel when given an additional laboratory sessions which allows them to observe several examples of the algorithm. Still another question is the best format for the laboratory session. Possible formats include allowing subjects to design their own data sets or animations or giving subjects predesigned materials.

When systems for algorithm animation have been used in educational settings, the systems often receive rave informal reviews from users. Nevertheless, there has been little formal research to determine the efficacy of using animations to teach computer algorithms and the questions listed above remain unanswered. Previous studies in related areas have provided mixed results. For example, in the domain of algebra problems, Reed indicates that an external lesson strategy must be combined with the animation in order to focus student attention on pertinent features of the animated display [Ree85]. In the area of teaching computer-based tasks, Palmiter and Elkerton present an experiment comparing animation teaching of a computer based-task to text-only presentation of the same task. In their study, the animation group was faster and enjoyed the lesson more [PE91]. However, in a delayed test, the text-only group was faster.

In the actual domain of animated algorithms, Stasko, Badre and Lewis present an empirical study of learning an algorithm using animations [SBL93]. Their study involved an advanced algorithm and examined students learning about the algorithm using (1) a textual description or (2) a textual description accompanied by an animation. The animation subjects scored slightly higher, but the difference was not statistically significant. The authors concluded that the animation alone is not sufficient to improve understanding, but must be included in a more active learning environment for complete understanding. Their study afforded tight controls on the two conditions, but it did not investigate the use of animated algorithms in a classroom setting. Our study was designed to mimic more closely a traditional classroom and laboratory use of animated algorithms.

In particular, our study was conducted in order to measure the efficacy of using animated algorithms in varied teaching approaches. The approaches varied in the level of control and active involvement of the student. The experiment was conducted in a classroom setting. Algorithm animations utilized were created with the XTango algorithm animation package [Sta92] and the Polka algorithm animation system [SK93]. Several variables were studied, including style of presentation of an animated algorithm as a lecture accompaniment (animation or transparencies), use of a laboratory session to clarify algorithm concepts, and student interaction with the animation during the laboratory session, where one group received prepared data sets and the second group created their own data sets.

This experiment was an in-depth look at how animated algorithms may be used in the teaching of computer algorithms. As in actual teaching of such algorithms, group sessions were used. Varied conditions allowed some groups to participate in a lecture with an extra laboratory session while other groups participated in lecture sessions only. All lecture sessions were accompanied by an example of the algorithm which was either a series of transparencies prepared in advance or the same data set illustrated by an animation. Laboratory sessions were of two types, either using prepared data sets or allowing the student to create a series of personalized data sets.

The question of how best to present material is an age-old question to the pedagogue. How, indeed, may one best transfer to others the concepts which are so clear to the teacher, yet such unknown territory to the learner? Felder and Silverman [FS88] among others, stress that students have many different ways of learning and that the learning and teaching styles of both student and teacher affect the results of the teaching process. One aspect of this is

LABORATORY CONDITION		Polka Animation	Prepared Slides
Lecture Only		15	15
Lecture plus Lab	PassiveLab	7	9
	Active Lab	7	9

Table 1: Design of Experiment

the active/passive dimension which reflects whether students learn best by experimentation or by developing theories. This issue is addressed by the two laboratory formats used in the experiment.

The use of animation is also becoming increasingly common in the area of Human Computer Interaction. It becomes crucial to formally investigate questions of how animation may best be incorporated into such areas as animated help [Suk88], animated interfaces [BS90], and animated presentation of concepts [BSM91]. Issues of design and use of animations hinge on the careful study of how animations may most effectively be used.

2 Empirical Study

This study was designed to investigate several presentation issues concerning the use of animations to present algorithms. The subjects of our study were students at the Georgia Institute of Technology enrolled in CS1410, the first programming course for Computer Science majors at the Institute. The students were volunteers who received class credit for their participation. Sixty-two students participated in the experiment.

2.1 Design

The experiment was a 2 x 2 (nested 2) design as represented in Table 1. One variable was presentation of the lecture example, using Animation or Prepared Slides. The second variable was Lecture Only or Lecture plus Laboratory Session. This design also encompassed a nested 2 level factor under laboratory session where the variable of concern was Laboratory Type, either Active or Passive.

The algorithm used was Kruskal's Minimum Spanning Tree Algorithm. Kruskal's MST Algorithm finds a set of edges of a graph that form a path to all vertices of the graph and that are also of minimum cost or weight. The first step in the algorithm is to sort the graph edges by their weight. The next step is to iteratively add the edge of least weight that does not form a cycle. The problem of finding a minimum spanning tree is commonly presented in early computer science courses and is often solved by using either Kruskal's or Prim's algorithm.

2.2 Materials

A lecture describing the algorithm was presented to all groups. The lecture was written in advance to ensure that each group would receive the same information. Students in the Lecture/Animation groups watched on individual workstations an animated example of the Kruskal's Minimum Spanning Tree Algorithm created by the Polka Algorithm Animation software. This software, which is similar to XTango, allows step-by-step control of the animation. Students in the Lecture/Slides group were shown the same example graph by means of a series of prepared transparencies. These transparencies were created from window-dumps of the Polka example. The examples differed in the dynamic nature of the animation.

For those students in the active and passive laboratory groups, a prepared sheet of instructions explained how to access the XTango animation of Kruskal's Minimum Spanning Tree Algorithm. The only difference in the two handouts was that the Active group was instructed how to create a graph to use as input to the animation, while the Passive group was given the names of prepared data files to use as input. A sample window from the animation appears in Figure 1.

The version of the animation used in this experiment was based on previous experiments which indicated that a monochrome version of the algorithm with algorithmic steps appearing as text was best, as measured by performance on a post-test [Law93]. This type of animation was used for both the lecture example and for the animation laboratory.

All groups completed a multiple-choice/true-false on-line test requiring application or understanding of the algorithm. Groups also completed a free response test on paper that was designed to require the students to articulate concepts relating to understanding the algorithm.

Sample questions from the on-line test appear below:

1. In Kruskal Algorithm, the first step in finding the Minimum Spanning Tree is:
   ______Sort the edges by weight
   ______Select the two shortest edges
   ______Select the shortest edge from node 1
   ______None of the above

2. In the given graph, if edges HG, IC, GF, CF, and AB are already in the path, which edge will be added next?
   ______IG
   ______CD
   ______HI
   ______None of the above

The questions below are selected from the free-response test.

1. Under what conditions would the next shortest edge not be added to the Minimum Spanning Tree?

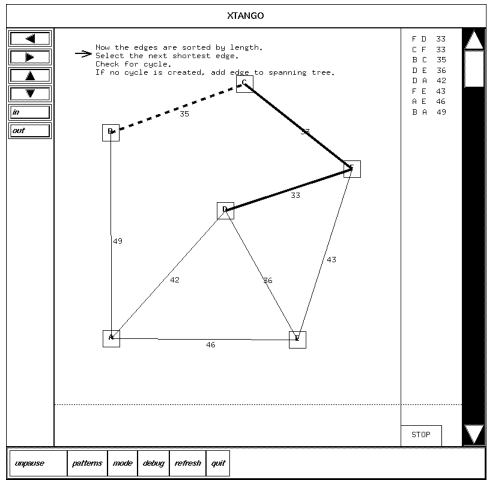

Figure 1: Kruskal's MST

2. What, in your opinion, is the key part of the algorithm which guarantees the Spanning Tree obtained will be minimal?

2.3 Procedure

The subjects were divided into 4 groups of approximately sixteen students each. See Figure 1 for the exact numbers. The groups were Lecture/Animation, Lecture/Slides, Lecture and Lab/Animation, Lecture and Lab/Slides. Each laboratory section was further divided into Active and Passive subsections, four groups. Students were randomly assigned to a particular sub-group. All students listened to the lecture presentation of the algorithm, accompanied by either the Polka animation or by the prepared slides. For those in the laboratory condition, interaction with the XTango animation followed. Students in the Active group created graphs and observed the workings of the algorithm on those graphs. Students in the Passive group were given a list of prepared file names and asked to observe the workings of the Kruskal MST Algorithm in the XTango environment on those files. All students were allowed twelve minutes for the laboratory session. The twelve minute time limit was derived from a previous experiment where it was determined that the average time a student spent experimenting with the graphs was twelve minutes [Law93]. Following this, the students completed the two post-tests.

3 Hypotheses

We hypothesized that the subjects who received the lecture accompanied by the animated example would perform better than those subjects who received the slides example. We also hypothesized that the addition of a laboratory session would lead to improved performance. Accuracy, the dependent variable, was measured on two instruments, the on-line fixed choice test and the paper and pencil short answer test that was designed to measure behavioral objectives based on concepts and applications of the algorithm. The tests differed

Fixed Response Test (Total = 19)			
LABORATORY CONDITION		Polka Animation	Prepared Slides
Lecture Only		11.87	11.80
Lecture plus Lab	Passive Lab	13.71	13.22
	Active Lab	13.83	13.89
Free Response Test (Total = 21)			
LABORATORY CONDITION		Polka Animation	Prepared Slides
Lecture Only		14.47	16.13
Lecture plus Lab	Passive Lab	16.43	16.67
	Active Lab	18.14	17.89

Table 2: Cell Means, Post-Tests

in approach. The fixed choice test was designed to concentrate on questions of procedure and the small steps of the algorithm. Questions were either multiple choice or true/false. The paper test was designed to concentrate on conceptual issues including motivation as well as the overall algorithm. Questions on this test were free response and required an explanation, an example, or a conclusion about a concept.

4 Results

Cell means for the experiment appear in Table 2. Inspection of cell means for both the on-line and the free response test, indicated that the active and passive laboratory groups scored higher than the no laboratory group. The active laboratory group also had the highest scores on the free-response test. These results appear in graph form in Figure 2. Statistical analysis was then undertaken in order to determine which of these differences were significant.

4.1 On-line Test

The following results are based on a maximum possible of nineteen points, one point for each correct question. The questions on the on-line test were either true/false or multiple-choice in format. This format allows two techniques, recognition and guessing, which are not easily applied in free-answer style questions.

The two factors in the first analysis of variance on test scores were Lecture Only versus Lecture Plus Lab and Polka Animation versus Prepared Slides.

In a comparison of groups that received the laboratory session, results indicated that students completing a laboratory session performed marginally better than those who had no laboratory session ($F=2.80$, d.f.1,59, $p< 0.1$) as measured by the on-line test. Cell means were 13.5 (of 19) for the laboratory session, compared to 11.83 for the no-laboratory

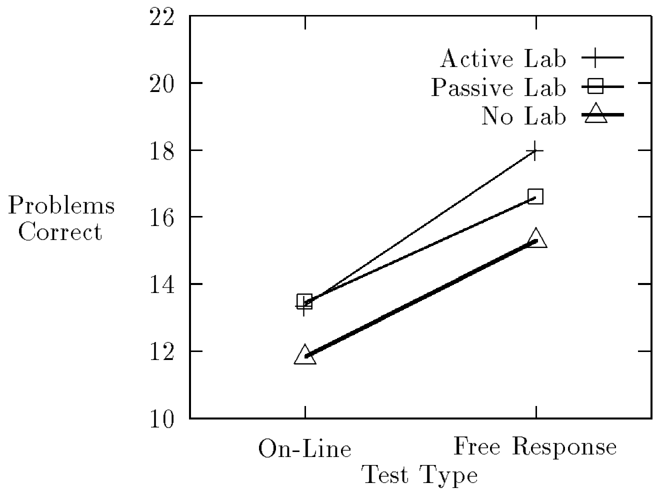

Figure 2: Cell Means, Three Lab Conditions

condition. There was no significant difference between the two groups: lecture accompanied by slides or lecture accompanied by an animation example. This may be explained by the fact that both groups were able to use visual techniques to supplement the algorithm and that either of these methods was adequate for the purpose.

4.2 Free Response Test

The following results are for the paper post-test requiring statement or application of concepts. This test had seven questions, each counted as three points for a maximum high score possible of twenty-one points. The three point scale allowed answers to be broken down into sections. The questions were designed to address the basic concepts necessary for understanding of the algorithm, in addition to a complete demonstration of the working of the algorithm on a provided graph.

Parallel analysis of variance was performed on the free-response test results. Results of this analysis indicated that students who completed a laboratory session performed significantly better on the free-response post-test ($F=4.36$, d.f. 1,58, $p < 0.05$) than those who did not. The amount of difference in this result indicates that student laboratory participation is more effective for questions which require more conceptual knowledge than questions which require recognition of the individual steps of the algorithm.

4.3 Laboratory Style, Active, Passive

These results led to further study of the differences among the conditions based upon the type of laboratory session – active, passive, or none. The factor used in the second

Cell Means	
No Laboratory	15.3
Passive	16.6
Active	18.0

Table 3: Cell Means for Three Lab Conditions, Number Correct of Twenty-One, Free-Response Test

analysis was type of laboratory session, No Lab, Active or Passive. Cell means indicated that those students in the active condition had the highest scores on the free-response test. These results appear in Table 3. Analysis of Variance for the three possible lab conditions indicated that laboratory condition was significant ($F = 2.83$, d.f. 2,59, $p < 0.07$). Pairwise t-test were performed to determine where the difference in condition actually lay. The significant difference ($p = 0.05$) was discovered between the active and the no laboratory condition.

5 Discussion

Our experiment was interesting in several different aspects. First, it appeared that the example used (animation vs. slides) did not make a significant difference in teaching the algorithm. In fact, the animation group did slightly worse on the free response test when no lab was involved. Even though it is valuable to have a visual aid to concept formation, the animation, while enjoyable to the student, did not provide added clarity over the transparencies. Certainly, this result could be a factor of the particular algorithm we studied. Other algorithms may benefit more from animation.

A second aspect of interest was that the advantage of the interactive laboratory session was confirmed. We found that these students excelled when compared to those in the passive laboratory condition as well as when compared to those students who did not participate in a laboratory session at all. Of special interest is the fact that the intuitive advantage of the laboratory group was not statistically supported. Simply having a laboratory session was not enough to improve performance; the issue of control and interaction also was necessary. A strong indication of this result is that one valuable use of these animations is to make them available to the students outside the classroom setting. Such availability may be provided in either a closed laboratory or open laboratory setting where students would create sample data sets and observe the workings of the algorithms to be learned on these sample data sets. Our result suggests that active student participation is a key issue in this design process.

A third feature of interest was that while those in the active laboratory performed at a slightly higher level for both portions of the test, the difference was larger for the free response test than for the on-line test. The nature of the two tests is important in understanding this result. In general, the questions on the on-line test required recognition of the correct response rather than generation of a response. These questions might be described as being more on a procedural and operational level than on a conceptual level.

This speaks to the issue of what types of learning are most affected by the use of the animations. A previous hypothesis was that these animations may aid in concept formation. The results support that hypothesis.

5.1 Conclusions

The results of this experiment indicate that an advantage was shown for students who interacted with the algorithm animation in the laboratory session. The advantage was more marked for those questions that required knowledge at a deeper level (the free-response test). Questions on this test required drawing conclusions from the questions asked, as well as demonstrating a holistic version of the algorithm. Students who received the laboratory session also performed better on the on-line true/false or multiple choice questions, but not at significant levels. This finding indicates that the animation session is more crucial for those conceptual questions than for the more basic operational level of question.

Study of the results for the two laboratory conditions indicated that students who are in the active condition and create their own data sets for the algorithm achieve higher scores than those who observe prepared data sets. The result suggests that a study on the effect of student involvement in the learning process should be conducted.

6 Implications

With the new curriculum for Computer Science (ACM IEEE 91) comes a focus on breadth of learning and exposure of the beginning computer science student to many varied concepts and areas of computer science. Animations may well be used to enhance this breadth of exposure and to enable the student to grasp an understanding of the field through an understanding of its underlying algorithmic processes.

Additionally, one emphasis of the new curriculum is to include more closed laboratory sessions during the courses of the computer science major. Animated algorithms could well be employed to serve as a portion of these closed laboratory sessions and to provide enhancement and reinforcement to lecture and textbook material.

Strong implications exist for application of these results in general areas of Human Computer Interaction. The design of animated help, animated interfaces, and other uses of animations can be guided by empirical results such as these which indicate the most effective use of animations.

References

[BH92] Marc H. Brown and John Hershberger. Color and sound in algorithm animation. *Computer*, 25(12):52–63, December 1992.

[BN93] Marc H. Brown and Marc A. Najork. Animating algorithms in 3D. In *Proceedings of the UIST 1993*, Atlanta, GA, November 1993.

[Bro88] Marc Brown. Exploring algorithms using Balsa-II. *Computer*, 21(5):14–36, 1988.

[Bro91] Marc H. Brown. Zeus: A system for algorithm animation and multi-view editing. In *Proc. 1991 IEEE Workshop on Visual Languages*. IEEE, October 1991.

[BS90] Ronald Baecker and Ian Small. Animation at the interface. In Brenda Laurel, editor, *The Art of Human-Computer Interface Design*, pages 251–267. Addison-Wesley, Reading, MA, 1990.

[BSM91] Ronald M. Baecker, Ian Small, and Richard Mander. Bringing icons to life. In *Proceedings of the ACM SIGCHI '91 Conference on Human Factors in Computing Systems*, pages 1–6, New Orleans, LA, May 1991.

[FS88] Richard M. Felder and Linda K. Silverman. Learning and teaching styles in engineering education. *Engineering Education*, 78(7):674–681, 1988.

[Glo92] Peter A. Gloor. AACE - algorithm animation for computer science education. In *Proceedings of the 1992 IEEE Workshop on Visual Languages*, pages 25–31, Seattle, WA, September 1992.

[Law93] Andrea W. Lawrence. *Empirical Studies of the Value of Algorithm Animation in Algorithm Understanding*. PhD thesis, Georgia Institute of Technology, 1993.

[PE91] Susan Palmiter and Jay Elkerton. An evaluation of animated demonstrations for learning computer-based tasks. In *Proceedings of the ACM SIGCHI '91 Conference on Human Factors in Computing Systems*, pages 257–263, New Orleans, LA, May 1991.

[Ree85] S. K. Reed. Effects of computer graphics on improving estimates to algebra word problems. *Journal of Educational Psychology*, 77:285–298, 1985.

[SBL93] John Stasko, Albert Badre, and Clayton Lewis. Do algorithm animations assist learning? An empirical study and analysis. In *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems*, pages 61–66, Amsterdam, Netherlands, April 1993.

[SK93] John T. Stasko and Eileen Kraemer. A methodology for building application-specific visualizations of parallel programs. *Journal of Parallel and Distributed Computing*, 18(2):258–264, June 1993.

[Sta90] John Stasko. TANGO: A framework and system for algorithm animation. *Computer*, 23(9):39–44, 1990.

[Sta92] John T. Stasko. Animating algorithms with XTANGO. *SIGACT News*, 23(2):67–71, Spring 1992.

[Suk88] Piyawadee Sukaviriya. Dynamic construction of animated help from application context. In *Proceedings of the '88 ACM SIGGRAPH Symposium on User Interface Software and Technology*, pages 190–202, Banff, Alberta, Canada, October 1988.

[SW93] John T. Stasko and Joseph F. Wehrli. Three-dimensional computation visualization. In *Proceedings of the 1993 IEEE Symposium on Visual Languages*, pages 100–107, Bergen, Norway, August 1993.