

# The Effects of Algorithm Visualizations with Storylines on Retention: An Experimental Study

Christopher D. Hundhausen<sup>†</sup>, Robert Patterson\*, Jonathan Lee Brown<sup>†</sup>, Sean Farley<sup>†</sup>

<sup>†</sup>*Visualization and End User Programming Laboratory, College of Electrical Eng. and Computer Science*

*\*Department of Psychology*

*Washington State University*

*Pullman, WA 99164 USA*

*{hundhaus, jbrown, sfarley}@eecs.wsu.edu, rpatter@mail.wsu.edu*

## Abstract

*Algorithm visualizations graphically illustrate how algorithms work. In prior ethnographic studies of a computer science course in which students were required to construct and present their own algorithm visualizations, we observed that visualizations based on storylines tended to stimulate increased audience interest and involvement. This observation, coupled with the empirical research that substantiates the value of stories as mnemonic devices, raises an interesting research question: Do visualizations with storylines actually help students remember the procedural behavior of an algorithm better than visualizations that do not involve storylines? To investigate this question, we conducted an experimental study that compared the memorability of algorithm descriptions involving differing degrees of spatial and verbal embellishment. The study failed to detect significant differences. We reflect on our lack of significant results, and suggest two alternative paths for future research into the value of story-based algorithm visualization.*

## 1. Introduction

Algorithm visualizations graphically illustrate how algorithms work. Traditionally, computer science educators have used algorithm visualization (AV) technology as visual aids in lectures, or as the basis of interactive laboratories. More than 20 experimental studies, summarized in [1], have aimed to substantiate the pedagogical effectiveness of AV technology. While the results of these studies have been widely mixed, one clear trend has emerged: learners who use visualizations as part of active learning exercises consistently outperform learners who passively view visualizations as though they were movies.

Given this trend, we have been exploring AV-based learning exercises in which students not only construct their own visualizations of the algorithms under study, but also present those visualizations to their peers and instructors for feedback and discussion. In prior ethnographic studies of a computer science course that included visualization construction and presentation assignments [2], 11 of 83 student groups constructed visu-

alizations that portrayed the target algorithm in terms of a story. Somewhat surprisingly, we observed that these story-based visualizations tended to stimulate markedly increased audience interest and involvement, as compared to more traditional geometric visualizations. This observation, coupled with the legacy of research that substantiates the value of stories as mnemonic devices (see, e.g., [3]), raises an interesting research question: Do story-based visualizations actually help students remember the procedural behavior of an algorithm better than algorithm descriptions that do not involve stories?

## 2. Experimental Study

To investigate this question, we carried out an experiment in which first-semester computer science students learned the Bubblesort algorithm through alternative descriptions of the algorithm containing various levels of spatial (visualization) and verbal (story) embellishment.

### 2.1 Design

The experiment considered three specific hypotheses:

- H1: Learners who learn an algorithm with the help of a story-based description will remember the algorithm better than learners who learn the same algorithm with a non story-based description.
- H2: Learners who learn an algorithm with the help of a visualization will remember the algorithm better than learners who learn the same algorithm with an equivalent purely textual description.
- H3: Learners who learn an algorithm with the help of a story-based description that well matches an accompanying visualization will remember the algorithm better than learners who learn the same algorithm with the help of a story-based description that poorly matches the accompanying visualization.

To test these hypotheses, we employed a between-subjects,  $2 \times 3$  mixed factorial design. As illustrated in Table 1, we obtained five treatment groups by crossing two independent variables:

		SPATIAL EMBELLISHMENT		
		None	Congruous	Incongruous
VERBAL EMBELLISHMENT	Simple	I (10)	II (15)	Not Tested
	Story	III (13)	IV (13)	V (16)

**Table 1.** Summary of Experimental Design. The Roman numerals denote treatment labels (referred to in the text below), while the numbers in parentheses indicate the number of participants in each treatment.

1. *Verbal embellishment.* The textual description of the algorithm used to learn the algorithm. Two variable levels were considered by our experiment:
  - a. *Simple*—a simple description not containing a storyline, and
  - b. *Story*—a story-based analogy that both motivates the algorithm, and functionally relates its procedural steps and logic to a real-world scenario.
2. *Spatial embellishment.* The spatial representation (visualization) that accompanies the textual description used to learn the algorithm. Our experiment considered three levels of this variable:
  - a. *None*—no spatial embellishment at all;
  - b. *Congruous*—a spatial embellishment that is consistent with the accompanying verbal description; and
  - c. *Incongruous*—a spatial embellishment that is inconsistent with the accompanying verbal description.

To measure learners' recall of the algorithm, we adopted three dependent variables:

1. Accuracy on three detailed algorithm tracing tasks;
2. Accuracy on a fill-in-the-blank post-test; and
3. Participants' self assessment of their understanding and memory of the algorithm (on a scale of 1 to 10).

## 2.2 Participants

We recruited 90 participants out of the Spring, 2004 offering of the introductory ("CS 1") computer science course at Washington State University. Participants received course credit for their participation.

Participants were assigned to conditions based on the course lab meeting in which they were enrolled. We screened participants based on pretest tracing tasks similar to those that they would complete within the study. Participants who demonstrated any prior knowledge of the target algorithm (Bubblesort) were discarded from our analysis. In the final analysis, the data of 67 participants (4 female, 63 male) were included.

## 2.3 Materials and Tasks

Participants in each treatment group learned the Bubblesort algorithm through alternative descriptions of the algorithm. The "Simple Verbal" treatments (I and II in Table 1) were given a standard Bubblesort description devoid of an underlying story. Here is an excerpt:

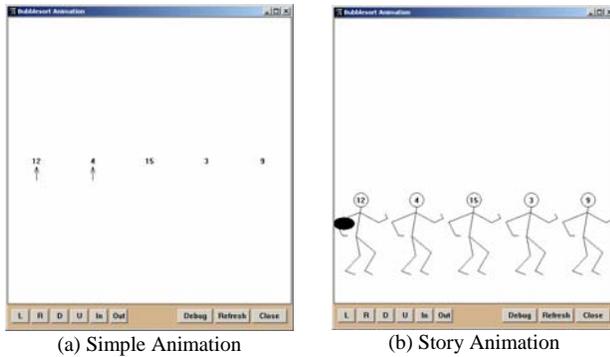
Given a list of  $n$  items, the Bubblesort works by making  $n - 1$  iterations through the items. In each of these  $n - 1$  iterations, the largest item in the unsorted portion of the list finds its rightful place in the list through successive exchanges with smaller items. Each iteration of the algorithm works from left to right, comparing pairs of adjacent items. ... If a pair of adjacent items are out of order, they trade places. If a pair of adjacent items are in order, they are left alone.

In contrast, the "Story Verbal" treatments (III, IV, and V) were given a description of the Bubblesort that portrayed the algorithm in terms of a game of American football. Here is an excerpt:

To illustrate how the Bubblesort algorithm works, we'll explain it in terms of a game of American football. In this game, a line of players of varying weights represent the list of items to be sorted. At the beginning of the game, the players are lined up in a row facing the "goal," which is to their right. The object of the game is to "score" (i.e., reach the goal) by advancing the ball from left to right to the end of the line of unsorted players.

Participants in the "Congruous" and "Incongruous" treatments (II, IV, and V) were instructed to study a computer-based Bubblesort animation operating on five self-selected input data sets. Built on top of the POLKA [4] algorithm animation framework, these animations portrayed the algorithm using the same spatial imagery used in the accompanying algorithm description (Treatments II and IV), or spatial imagery different from that used in the accompanying algorithm description (Treatment V). A snapshot of the "simple" visualization used by Treatments II and V appears in Figure 1a; a snapshot of the "story" visualization used by Treatment IV appears in Figure 1b. In contrast, participants in the "No Spatial" treatments (I and III) were not provided with an animation; instead, they were asked to use pencil and paper to hand-simulate the algorithm for five self-selected input data sets.

Following the learning session, participants completed a series of three tracing tasks in which they had to specify, for each of three input data sets (in order, reverse order, and random order), the specific comparisons and exchanges that took place in each outer loop iteration, along with the state of the array after each outer loop iteration. In addition, they completed a 14-question posttest (not analyzed here) consisting of a mix of true-false and fill-in-the-blank questions that tested their ability to make generalizations about the Bubblesort procedure. An exit questionnaire (not analyzed here) asked participants to self-rate, on a scale of 1 to 10, their understanding of the Bubblesort algorithm, and the difficulty of remembering the algorithm.



**Figure 1.** Snapshots of Bubblesort Animations used by participants in Treatments II, IV, and V

## 2.3 Procedure

The study was spread over two sessions. In the initial session, which lasted roughly 90 minutes, participants began by filling out an informed consent form and screening questionnaire. Next, participants completed a pretest consisting of tracing tasks similar to those in the posttest. Following that, they read the Bubblesort algorithm description, and were given 45 minutes to interact with the animation (Treatments II, IV and V) or hand-trace the algorithm (Treatments I and III). Then, they were given 10 minutes to complete each of three posttest trace tasks. Finally, they filled out the exit questionnaire.

In the follow-up session scheduled a month later, participants were given 15 minutes to complete each of three trace tasks isomorphic to the trace tasks in the initial session. In addition, they completed the follow-up written post-test, and the follow-up exit questionnaire.

## 2.4 Results and Discussion

Table 2 presents each treatment group's mean percentage correct across the three trace tasks, along with its mean "memory degradation ratio," obtained by averaging each participant's follow-up percent correct by his/her initial percent correct. As these data indicate, the initial and follow-up mean percentages correct were quite similar across the five treatments, with values ranging from 59 to 73 percent. Not surprisingly, according to ANOVAs, no significant differences exist with respect to either initial ( $df = 66, F = 0.21, p = 0.93$ ) or follow-up ( $df = 66, F = 0.56, p = 0.69$ ) trace task accuracy. These results suggest that participants generally performed rather poorly on the trace tasks, regardless of the degree of verbal and spatial embellishment in the learning materials. The most common misunderstanding we observed had to do with the termination condition of the algorithm: many participants incorrectly thought that the algorithm terminates once the data values are in order.

Moreover, according to a Kruskal-Wallis ANOVA, no significant differences exist among the treatment groups

Treatment	Initial Session	Follow-up Session	Ratio
	% Correct	% Correct	Follow-up ÷ Initial
I	65.8 (21.5)	59.2 (22.2)	0.89 (0.36)
II	65.0 (32.7)	64.3 (31.3)	1.20 (0.92)
III	73.4 (24.3)	69.8 (14.0)	1.12 (0.70)
IV	65.4 (28.1)	71.4 (25.9)	1.04 (0.43)
V	69.3 (30.0)	71.7 (26.2)	1.18 (1.00)

**Table 2.** Mean Percentages Correct and "Memory Degradation Ratios" by Treatment (Standard deviations are in parentheses)

with respect to their retention of the Bubblesort algorithm ( $df = 66, H = 1.27, p = 0.87$ ). In fact, in four out of the five treatment groups, the follow-up traces that participants performed one month after the learning session were, on average, more accurate than the original traces they performed right after the learning session. One possible explanation of this unexpected result is that, even though it did not cover the Bubblesort algorithm, the additional month of computer science instruction increased participants' general understanding of the kind of iteration present in the Bubblesort, enabling them to compensate for algorithm details they may have forgotten.

In sum, it appears that, given a brief, 45 minute learning period, many introductory computer science students may fail to develop a correct mental model of the Bubblesort algorithm, regardless of the level of verbal or spatial embellishment in the learning materials. It could be the case that a 45-minute learning session was simply too short for the hypothesized effects to develop. A follow-up study could explore this possibility by lengthening the learning session. Alternatively, it could be the case that the real advantage of story-based visualizations lies not in their memorability, but in their ability to stimulate meaningful discussions about algorithms. In future research, we plan to investigate this issue in greater detail through post-hoc video analysis of student/instructor discussions of traditional and story-based algorithm visualizations.

## 3. Acknowledgment

This work was supported by the National Science Foundation under grant no. 0133212.

## 4. References

- [1] C. D. Hundhausen, S. A. Douglas, and J. T. Stasko, "A meta-study of software visualization effectiveness," *Journal of Visual Languages and Computing*, vol. 13, pp. 259-290, 2002.
- [2] C. D. Hundhausen, "Integrating Algorithm Visualization Technology into an Undergraduate Algorithms Course: Ethnographic Studies of a Social Constructivist Approach," *Computers & Education*, vol. 39, pp. 237-260, 2002.
- [3] J. L. Santa, A. B. Ruskin, and A. J. H. Yio, "Mnemonic systems in free recall," *Psychological Reports*, vol. 32, pp. 1163-1170, 1973.
- [4] J. Stasko and E. Kraemer, "A methodology for building application-specific visualizations of parallel programs," *Journal of parallel and distributed computing*, vol. 18, pp. 258-264, 1993.