

# **Turning the tables: Investigating characteristics and efficacy of student-authored animations and multimedia representations**

Teresa Hubscher-Younger, Rensselaer Polytechnic Institute  
N. Hari Narayanan, Auburn University and National Science Foundation

## **Abstract**

Explanatory representations such as animations enable students to analyze and understand difficult concepts. Typically experts produce animations and explanations. Seldom do students themselves, as part of their learning process, create such representations. We summarize three studies of learning complex concepts (computer algorithms) from expository representations. The first is an observational study of college students learning algorithms from representations provided by their instructor and textbook. The study revealed a tendency of students to converge on one explanatory representation as the primary means to understand the concepts. This premature convergence blinded them to the limitations of that representation and impeded learning. So we investigated learning and representational diversity that result when students author their own representations to explain concepts to themselves and to their peers. In two subsequent studies we found that authoring representations significantly improved learning. Students constructed a richer set of representations than those found in typical instructional materials, in terms of media content and diversity of perspectives. Our findings suggest that a peer-to-peer approach to learning, in which students create and evaluate their own animations and multimedia representations, can be quite effective.

## **1. Introduction**

Providing multiple perspectives on complex concepts can promote deeper learning. This principle, in its simplest form, can be seen at work if one opens up any popular textbook on a complex topic in science or engineering. One will find textual descriptions of concepts and ideas interspersed with symbolic (e.g. equations) and graphical (e.g. diagrams or photographs) representations. Animations have the capacity to present dynamic aspects of a situation directly and explicitly. It is another way of presenting multiple perspectives to a learner, one that is particularly well suited to verbally explaining and visually illustrating complex procedures (e.g. computer algorithms) and dynamic processes (e.g. meteorological events). In fact, animations can be considered to be a special case of multiple representations in which a series of pictures are fluidly presented, along with other representations such as sounds and spoken words.

There is a wealth of literature on the educational benefits of animations as well as multiple static representations. For instance, Cox and Brna (1995) found that while only 17% of students solving analytical reasoning problems tended to use multiple representations, those who did performed better. In our work on learning in the complex

domain computer science, we found that students learn more about an algorithm from interactive visualizations that presented a suite of multiple representations - animations integrated with hypertext explanations - than from lectures and descriptions of algorithms in a style that is typically found in textbooks (Hansen, Narayanan & Hegarty, 2002). We also discovered from ablation experiments that learning is negatively impacted when animations are selectively removed from this suite of multiple representations (Hansen & Narayanan , 2000).

Providing multiple perspectives on a complex concept involves more than providing multiple representations of the same information. Learning is enhanced when representations that differ not only in modality but also in information content, with both redundancy and differentiation of information, are made available to the learner. For instance, pictures, by their very nature, present a different perspective on the subject matter than text. Mayer (2001) found that comprehension improved when text and graphics were presented side by side instead of one after the other. This is one possible reason for the effectiveness of animations in explaining dynamic concepts and processes. Animations typically employ both verbal narratives and fluid visuals, presented concurrently, to explain things that change over time.

Another argument for the potential benefits of exposing a learner to multiple perspectives arises from Lakoff and Johnson's theory of concept formation (1999). They argue that metaphor is the basis for all concept formation. According to their theory, abstract concepts are derived from multiple metaphors interacting and emphasizing different aspects of the concept. This suggests that analogical or metaphoric explanations can produce a richer understanding of an abstract concept than literal explanations. While we are not aware of research that directly tests this conjecture, our ablation experiments with algorithm visualizations revealed that students who viewed animated analogies of algorithms prior to viewing literal animations of the algorithms learned more than students who only saw the literal animations (Hansen & Narayanan, 2000). For example, an algorithm for sorting numbers can be illustrated with an analogy of organizing playing cards. But note that, like the algorithm, the analogy is also dynamic. It is difficult to capture the essence of such an analogy with a static picture or a verbose description as well as an animation can. Animations are multiple and fluid representations that can not only illustrate a complex and dynamic process to a learner, but also show dynamic analogies and metaphors that allow learners to make links between the dynamics of the process or concept being learned and their existing background knowledge or everyday experience.

Despite these advantages, traditional classroom instruction on algorithms mostly relies on static representations created by experts – text and diagrams found in textbooks, and verbal descriptions and pictures spoken, written or drawn by the teacher. Another issue is that animations or other kinds of multiple representations may not always lead to successful learning. For instance, Lowe (1999) found that novices viewing weather animations focused more on perceptually salient aspects than on thematically significant aspects. Novices may not have sufficient knowledge to identify and attend to thematically salient aspects of a representation. Another possibility is that even when multiple

representations are presented to learners, they may choose not to use all available representations in the learning process. Even if they do, the added cognitive effort of translating between representations may negatively impact learning (Ainsworth et al 1998). Similarly in the case of animations, the cognitive effort of mapping between the pictorial elements of the animation and the domain concepts they represent may impede learning. Surely, there are many aspects of learning from animations and multiple static representations that are in need of further research.

These problems can perhaps be avoided by having learners themselves construct, share and discuss expository representations of the concepts they are learning. For example, if students are asked to build animations of weather phenomena they are learning about, they are likely to make aspects of the meteorological domain that are thematically salient to them also appear perceptually salient in their animations. If they create multiple representations to explain complex concepts, and share, discuss and evaluate each other's representations, all representations may be equally understood and inter-representation translation may no longer be a problem.

Furthermore, constructionist theories of learning hold that learning will be deeper if students develop and share their own diverse understandings of a concept. If all students gain their understanding from the same expert-created representations, they will likely develop a uniform understanding rather than a diverse set of insights. This raises the question of whether the diversity of representations that students learn from can be increased by having students create explanatory representations instead of relying solely on instructor and textbook provided representations. Extant literature contains many studies of the efficacy of animations and other kinds of representations created by expert teachers and researchers. However, studies on the characteristics and efficacy of student-created representations are much less numerous. Therefore, the focus of this chapter is on characterizing student-authored expository representations and the learning that results from authoring and evaluating representations. We use the term "representation" broadly, to include both dynamic representations, such as animations, and static ones, such as pictures; thus our focus is not on learning from animations per se, but on learning from multiple representations that include animations. We describe investigations of whether college students of computer science naturally tend to use all available representations while trying to understand an algorithm, and the characteristics of expository representations that they themselves create to explain algorithms to their peers.

The rest of this chapter presents summaries of three studies that investigated the following questions. When students engage in their natural or habitual learning practices and study an algorithmic concept, do they effectively use all available representations? The first study reported here uncovered students' tendency to converge on one explanatory representation rather than use all available representations. It also revealed a limitation of the most common representations students use in learning: static representations such as text and pictures found in textbooks, and words and diagrams spoken, written or drawn by the teacher. Such representations may be inadequate, in comparison to dynamic representations (i.e. animations), to accurately and completely portray all aspects of a complex dynamic process.

If instead of studying representations provided by a teacher or textbook, students are asked to create, share and critique explanatory representations of a complex concept amongst themselves, what kinds of representations will they build – animations, static graphics, textual narratives or combinations of these styles? How diverse will these representations be? If this exercise in constructivism is repeated multiple times in class over a semester, how will the nature and styles of representations change over time? Will students learn from the activity of authoring and evaluating explanatory representations? The second and third study addressed these issues.

## **2. Student use of multiple representations in traditional instruction**

We address the aforementioned questions in the context of undergraduate student learning in the domain of computer algorithms. Algorithms are fundamental and abstract procedural concepts in computer science. These concepts are generally considered to be difficult to learn and teach. Because data and algorithms do not have physical manifestations, it is difficult for someone to “observe” an algorithm or to “show” someone how the procedure operates on and transforms data. To explain these concepts, textbooks and teachers often rely on verbal explanations and graphical representations of data, and how it changes over time as a result of an algorithm operating on it. This difficulty has also given rise to researchers developing various kinds of algorithm animations during the last two decades (see Hundhausen, Douglas & Stasko, 2002 for a survey). Despite the fact that first such animations were designed in the early eighties, algorithm animations remain in the realm of research and are not yet widely used in college level algorithm instruction. The teaching of algorithms at the undergraduate level still primarily depends on static pictorial and textual representations available in textbooks and created by teachers. So our first study looked at the way multiple representations drawn from traditional instruction are used by a group of college students majoring in computer science during an algorithm studying exercise.

### ***Study I***

#### **Method**

This was a qualitative observational study of natural or habitual learning practices of students engaged in studying an algorithm. Sixteen students from a university department of computer science and software engineering who were enrolled in an introductory algorithm analysis course volunteered to participate in the study, in return for extra course credit. Their sex, race, work experience, previous schooling, nationality and age varied considerably. Based on their reports that they typically study for this course in small groups, they were split into six groups of two to three students each. Students were grouped together based on what times they were available for the study, and so we grouped students together who had never met and who did not normally study together.

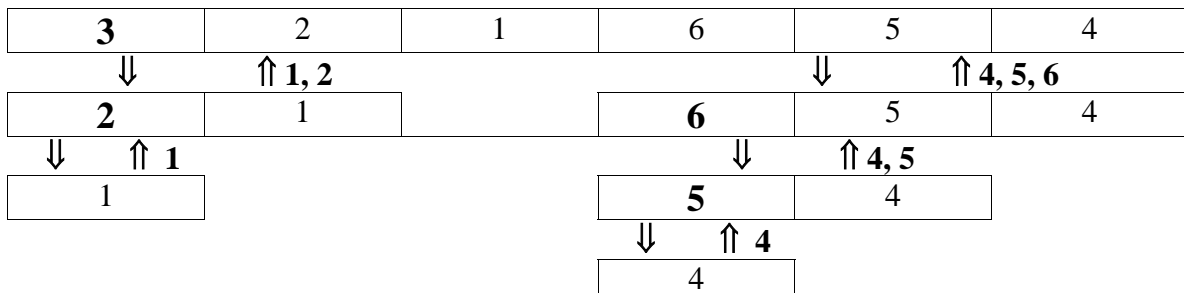
The course instructor was videotaped explaining the Quick Sort algorithm. The videotaped lecture resembled the lectures the students normally attended, i.e. it was

presented by their instructor, who used only a white board and markers, his usual method of presenting the material. This videotape was shown to the students during the study session. The students also brought the course textbook, which they normally use to study for the course, to the session. The instructor also provided a printed lecture summary. This summarized the material he presented in the lecture, and also contained a step-wise description of Quick Sort. This summary was given to the students before they watched the lecture. Thus, the following multiple representations were available to the students: descriptions and diagrams explaining the algorithm in the textbook, the instructor's lecture, a diagram he drew on the whiteboard during the lecture, the notes students made while watching the videotaped lecture, the teacher's lecture summary and a step-wise description of the algorithm.

The experimenter ran a session for each of the six groups separately. In each session, after an introduction to the study, the group watched the videotaped lecture. They were allowed to take notes. After watching the video, they were given a set of questions about the algorithm and asked to study the algorithm using any of the materials available and then answer the questions. While the students were studying the algorithm and answering questions, they were videotaped. The experimenter also observed student activities and interactions, and made notes. After the session, the experimenter collected any notes or pictures students made and conducted a short interview of the group or the individuals in the group separately. These interviews were used to collect the students' thoughts about the session and to understand how they normally study algorithms.

## Results

We expected to see students using multiple representations that were available to them in order to answer the questions. Videotapes, observations and interview responses revealed that students in this study chose to consider a single pictorial representation presented by the instructor, a recursion tree diagram (Figure 1), over all the other representations presented or available to them for understanding the algorithm. This graphical representation became central to their understanding of the algorithm, and they acted as if they would be able to understand everything about the algorithm by trying to understand the algorithm using this one representation, and subsequently reason using this representation to come up with answers to the questions. Consequently, they failed to correctly answer questions that required information not available in this diagram.



**Figure 1:** A recursion tree diagram showing how the Quick Sort algorithm makes recursive calls with different inputs extracted from the original input.

The instructor explicitly said while presenting this representation that all he was showing using this representation was one, albeit central, aspect of the algorithm, its pattern of recursive executions. The lecture and the pseudocode given to the students covered other equally critical aspects of the algorithm. Nevertheless, students did not pay attention to the limitation of this graphical representation, that it only presented a partial (i.e. only shows snapshots of one aspect of the algorithm's dynamics) and high-level (i.e. without details of dynamic operations on individual data items carried out by the algorithm) static view of the algorithm's execution.

All students realized when they encountered questions about algorithm steps not explicit in the recursion tree diagram, that their understanding of the algorithm was incomplete. But instead of reaching out to other representations and explanations available to them, they struggled to construct answers to such questions based on their understanding of the algorithm from the recursion tree diagram. The information to answer these questions was available to them from various representations in their textbook. Most students observed did not seek out information from the textbook to supplement their understanding from the instructor provided representation. They convinced each other that they did not need to use other available representations to answer the questions, and on occasion even decided not to accept correct answers that some group members derived from other representations.

### **Discussion**

One detriment to learning revealed by this study was *premature convergence*: students converged on one representation and understanding of the algorithm too early in the learning process, without considering other available representations. We suspect that all groups converged on one diagram as their representation of choice because it was presented and discussed extensively by the instructor in the videotaped lecture (thereby making it an authoritative representation), not because it provided the most complete explanation. This observation is consistent with Milgram's studies on student obedience to authority (Milgram, 1963, 1965).

An undesirable result of this convergence was that despite realizing that information needed to answer some questions could not be derived from this one representation, students attempted to construct answers rather than seek additional information from other representations available to them. Mulholland and Eisenstadt note a similar phenomenon happening with novice users of software visualizations (1998). They argue that when students discover a mismatch between a software visualization and their assumptions, they work to reinterpret the visualization, so that it is consistent with their expectations, rather than investigate the disparity further. Clearly, the tendency to privilege one representation over others is problematic.

A second impediment to learning, *groupthink* (Janis, 1967), was manifested by all of the groups. Often the groups would convince themselves that explanations based on a faulty understanding of the algorithm were in fact correct.

Even when students were observed accessing other representations such as descriptions and diagrams from their course textbook, it was clear that they became frustrated in trying to integrate those with the instructor's descriptions and diagram. Thus, *information integration* – successfully combining information from multiple representations -- was observed to be another learning difficulty.

### **3. Turning the tables: Having students author and evaluate their own animations and multimedia representations**

One approach to addressing the three problematic aspects of student learning practices identified in the previous study – premature convergence, groupthink and information integration – is for instruction to emphasize one complete and accurate representation of each concept. However, it is almost impossible for one static representation – a textual description or a diagram – to capture all aspects of complex and dynamic concepts such as algorithms accurately and completely. Any single representation of a complex algorithm, even when accurate, is likely to be incomplete. This is precisely why students' choice of a single representation to focus on in the previous study led to incomplete learning. On the other hand, narrated animations can simultaneously explain and illustrate the detailed and time-varying behaviors of even the most complex algorithms. Perhaps students might have learned more, and answered the questions more accurately, had the instructor used an animation instead of a static picture like Figure 1 to illustrate the behavior of the Quick Sort algorithm over time. This provides a strong case for the development and integration of animations into algorithms curricula, and for using animations as the central representations in lectures.

Despite the expressive power of animations, research in the field has shown that algorithm animations (Byrne, Catrambone & Stasko, 1999), and animations in general (Tversky, Morrison & Betrancourt, 2002), do not always provide learning benefits. Our prior research (Hansen, Narayanan & Hegarty, 2002) demonstrated that animations have to be embedded within a larger context of multiple explanatory representations in order to be educationally effective. However, interactive visualizations in which multiple synchronized animations are integrated with hypertext explanations, and facilities for asking students questions and allowing them to make predictions are provided, are extremely time-consuming to design and produce (Hansen, Narayanan & Hegarty, 2002).

Another approach is to encourage students to first *individually* create, and then *collectively* share and evaluate their own animations and other kinds of explanatory representations of concepts prior to teaching them the concepts in a classroom lecture. Since this makes a plethora of representations created by a group of peers available to each student, no single representation is likely to be considered more authoritative than the others. This can possibly counteract student's tendency to seek out and latch onto one representation. Since representation authoring is done individually and subsequently discussed and evaluated by the entire class collectively, groupthink may be less of a problem. The act of individual authoring, and the common prior knowledge, culture and social experience shared by students, can possibly make integrating information from

one's own and others' representations easier than integrating information from representations created by experts outside the peer group.

Learning from a large set of diverse student-created representations may lead to a richer, more connected concept of an algorithm than learning from a small set of instructor and textbook provided representations. As Cox and Brna (1995) argue, multiple representations are more effective for problem solving than relying on a single representation for answering all questions, and people vary considerably in how able they are to use and understand different types of representations. So a diverse set of representations is likely to better match the different cognitive and learning styles of students.

We implemented such an approach in an algorithms course and conducted two studies to investigate student learning and characteristics of student-created representations. In these studies, students authored, shared and evaluated each other's algorithm animations and other multimedia algorithm representations. To facilitate the sharing and evaluation of representations, we designed a computer supported collaborative learning tool called CAROUSEL. This tool allows students to store and share their representations with each other and to evaluate these representations. More details on CAROUSEL can be found in (Hübscher-Younger & Narayanan, 2003). Students could upload their representations, in the form of text, images, audio files or animations, to CAROUSEL, which displayed these items on the web so that other students could log in and view each representation and rate it using a set of pre-determined criteria.

Given the increasing prevalence of animations in electronic learning materials and the web, students are likely to be familiar with animations as explanatory representations. Existing software such as Macromedia Flash makes authoring simple animations relatively easy. Moreover, as discussed in the introduction to this chapter, animations are well suited to explaining the dynamics of algorithms. Therefore, we expected that students would predominantly create and share animations and other multimedia representations quite different from the static textual and pictorial representations found in textbooks and classroom blackboards.

## *Study II*

### **Method**

Twelve students from the same university department who were enrolled in an introductory data structures and algorithms course volunteered to participate in the study, in return for extra course credit. The study occurred over four weeks with three different algorithms (Fibonacci Number Series algorithm, Selection Sort algorithm and Merge Sort algorithm) presented as course assignments with three identical stages. In the first stage, the student participants were given a printed step-wise description of an algorithm, and asked to study and understand it themselves. Following this, each participant was asked to create an explanatory representation, i.e. one that explains the algorithm to someone else. This representation could involve text, pictures, video, animations, sounds or speech

in any combination. The student then used CAROUSEL to store the representation and to exhibit it to all other students in the volunteer group. This activity took place over a one-week period.

After all representations were online and visible to everyone in the group, the students started the second stage, in which each student reviewed and evaluated all representations (except one's own) using CAROUSEL. To evaluate representations, students used a form in CAROUSEL asking them to rate the following six characteristics of each representation using a Likert scale of 1 to 5.

- Usefulness (How central was this representation to your understanding of the algorithm?)
- Understandability (How easy was this representation to understand?)
- Salience (How well did this representation point out the important features of the algorithm?)
- Familiarity (How familiar were you with the content of the representation?)
- Pleasure (How much did you enjoy the way this representation communicated the algorithm?)
- Contiguity (How well did this representation connect with the other representations of this algorithm?)

These raw ratings and their average values provide a set of peer-measures for the *perceived quality* of a representation. This activity took place over a one-week period. After the week for the evaluating task was over, the students took a post-test to test their understanding of the algorithm. This was the third stage. Any classroom discussion of an algorithm used in this study occurred only after this posttest.

## **Results**

A total of 36 representations were created by 11 of the 12 study participants. Students constructed a variety of representations, differing both in style and content. Some students submitted pure text containing elaborate metaphorical stories that illustrated what an algorithm computes, such as a story about a shopkeeper who used the Fibonacci series to balance his shelf of statues (Figure 2). Others produced entertaining animations that illustrated the mathematical basis of an algorithm, such as the “Dancing Hampsters” (spelling in context) showing the Fibonacci series (Figure 3). Most representations were of a walkthrough style, giving an example of a data set and showing how it would change over time as the algorithm operated on it.

---

## Fibonacci algorithm representation 7

### The Marble Statues

There was once an old salesman who had just acquired a large inventory of marble statues. Almost all the statues were different sizes, ranging from very small to some that were so big the salesman could hardly lift them.

Thinking to make his fortune with these statues, the salesman rented a small shop on the main street, and bought a shelf to display some of his statues to the public. Unfortunately, once he got the shelf to his shop he realized that although the shelf was very strong, it wasn't very stable. He found that if the items on the shelf were unbalanced, then the shelf would eventually begin to lean until it finally toppled over.

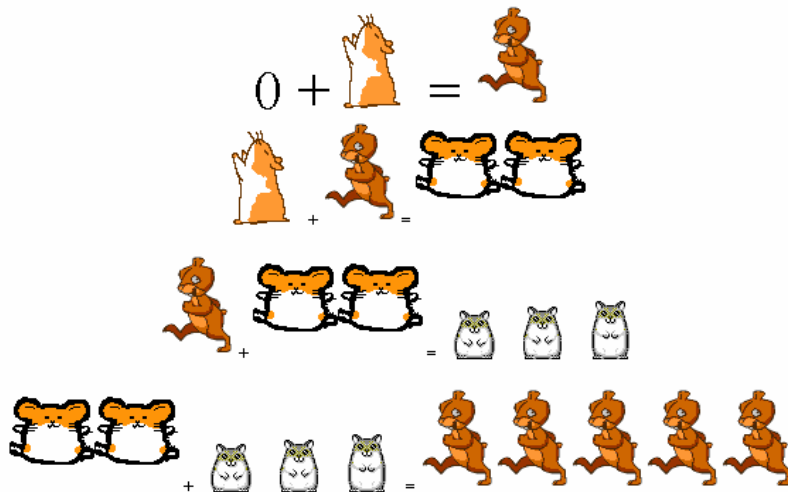
Unable to afford a new shelf, the salesman hit upon an idea. He searched through his inventory of statues until he found two one pound statues. He placed one statue on each end of the shelf to keep it balanced, but after looking at it for awhile decided that the shelf looked to bare. So he searched through his inventory until he found a two pound statue. He placed the two one pound statues on one end of the shelf, and balanced it out with the two pound statue on the other end. Now satisfied, he made up a sign, and opened his shop for business.

Later that day, a woman came into his shop and was quite taken by the two one pound statues on the shelf. She wanted to buy both statues, but couldn't afford them. Eventually, she made up her mind, and decided to just buy one of the statues. The salesman gladly wrapped up the statue for her and took her money.

Although he was happy to have finally made a sale, this caused a bit of a problem for the salesman. Now his shelf was no longer balanced, and he could already see that it was leaning slightly. He searched and searched through his inventory for another one pound statue to replace the one he had sold, but the closest he could get was a three pound statue.

**Figure 2** Part of a metaphoric story representing a recursive algorithm for calculating the Fibonacci Number Series.

*Now that the audience is prepared,*  
***ON WITH THE HAMPSTERS!!!***

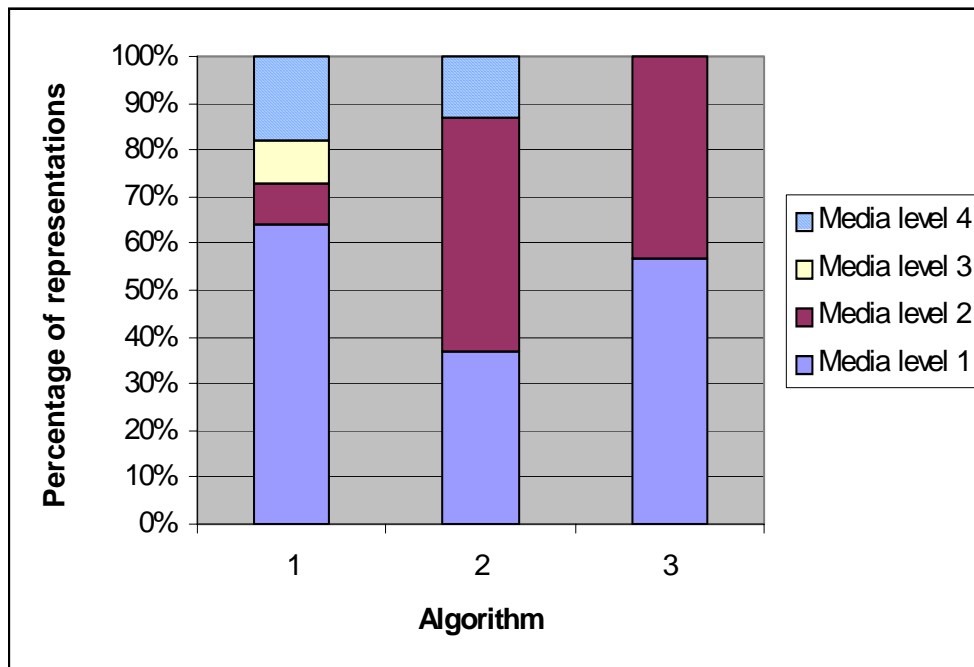


**Figure 3** An entertaining animation produced to represent a recursive algorithm for calculating the Fibonacci Number Series.

For two of the three algorithms that were used in this study, positive correlations between creating and sharing a representation and posttest scores were found ( $r=.635$ ,  $p=.07$ ;  $r=.663$ ,  $p=.05$ ). This suggested a positive relationship between authoring/rating representations of algorithms and understanding the algorithms; however, this does not necessarily indicate a strong causal relationship.

Media use by students was categorized using integers 1 to 4 with 1 being the use of only text, 2 being the use of graphics and text, 3 being the use of 2-D animation or other graphics/text/speech combinations and 4 being the use of 3-D animation or hypermedia (i.e. hyperlinked graphics/text/speech combinations). Multiple logistic regression analysis techniques were then employed to look at how the use of these different kinds of media in the representations affected student ratings of representation characteristics. Media use had a marginally significant effect on student ratings of all six characteristics: usefulness, understandability, salience, familiarity, pleasure and contiguity,  $G^2$ s (3, 237)= 7.8, 15.4, 6.6, 7.6, 7.8 and 15.2, respectively,  $p$ s<0.1.

Another interesting result was a convergence observed over the four weeks in representation characteristics. For the first algorithm approximately 64% of the representations were text only, 9% were text and graphics, 9% included animations and sound and 18% had more complex media. For the second algorithm, the number of text-only representations decreased (37%), those with graphics increased (50%), and the use of animations and complex media decreased (13%). For the third and last algorithm, only text representations (57%) and representations with graphics (43%) were used (Figure 4).



**Figure 4** This graph shows the percentage of representations created for each algorithm at each media level in Study II. The media levels are as follows: Level 1 is text only; Level 2 contains text and graphics; Level 3 contains text, graphics, sound or 2-D animation or sound; and Level 4 uses 3-D animation or hypermedia.

The experimenter rated all representations on a scale of 1 to 5 with 1 being a rating for representations that are least like a textbook or classroom explanation and 5 being a rating for representations that are most like a textbook or classroom explanation. These ratings increased over time with each new algorithm: the first algorithm had an average rating of 3.4; the second 3.9; and the third 4.7. So, over time, students created representations that were increasingly similar to conventional or familiar styles.

Furthermore, the average of all the ratings the students gave each representation was significantly positively related to the rating of how similar that representation was to a textbook or classroom explanation ( $F(1, 24)=3.9, p=.06$ ). Multiple linear regression analysis techniques were used to explore how the ratings of the representations' similarity to textbook or classroom explanations were related to the student ratings of different characteristics. The similarity ratings' relations to students' ratings of usefulness, salience and contiguity were positive and significant ( $F(1,24)=6.5, 6.0, \text{ and } 10.6$  respectively,  $p<.05$ ).

## **Discussion**

Discussion of the results of this study can be structured around four questions. When students are given the freedom to create their own expository representations instead of being handed expert-created representations, do they generate a diverse set of multiple representations? Do students use a variety of media in their representations? How does media use impact perceived quality of representations? Are student-authored representations similar to, or different from, conventional or familiar styles? How does similarity to conventional or familiar styles impact perceived quality of representations? Do they learn from authoring and evaluating each others' representations? Our expectations were that students would create a wide variety of representations with animations and multimedia, and that they would rate such media-rich representations as being higher quality. We assumed that the representations they generated would contain both conventional and non-conventional styles without either kind being dominant, and expected to find learning benefits from the acts of creating and evaluating representations. We did not have a priori expectations regarding the impact of conventional styles on perceived representational quality.

For the first algorithm assignment, students created representations that contained text, graphics, animations and sound. They employed a variety of styles, perspectives and media in their representations. Media use had a significant effect on student ratings of the six characteristics: usefulness, understandability, salience, familiarity, pleasure and contiguity. Adding graphics to text had a positive effect on all rating dimensions except familiarity. Adding animation or sound to representations with text and graphics had a positive effect on all rating dimensions except contiguity. Interestingly, adding more complex media types such as hypermedia and/or 3-D animation always led to a large negative effect on the ratings. Student-authored representations were initially very different from conventional styles found in textbooks. But over time, students created representations that were increasingly similar to familiar styles (i.e. text and graphics,

with no animations). The rating of this similarity to conventional representations was positively related to students' overall ratings of representations and their specific ratings of usefulness, salience and contiguity. In other words, how similar a representation was to a textbook or classroom explanation positively influenced not only the overall rating of that representation, but also student ratings of how useful that representation was to their understanding of the algorithm; how well that representation pointed out the salient features of the algorithm; and how well it was contiguous with (built upon) the other representations for that algorithm. Correlations between creating and rating representations and posttest scores suggested that authoring and rating representations is positively related to learning (for two of the three algorithms students saw in this study).

### ***Study III***

#### **Method**

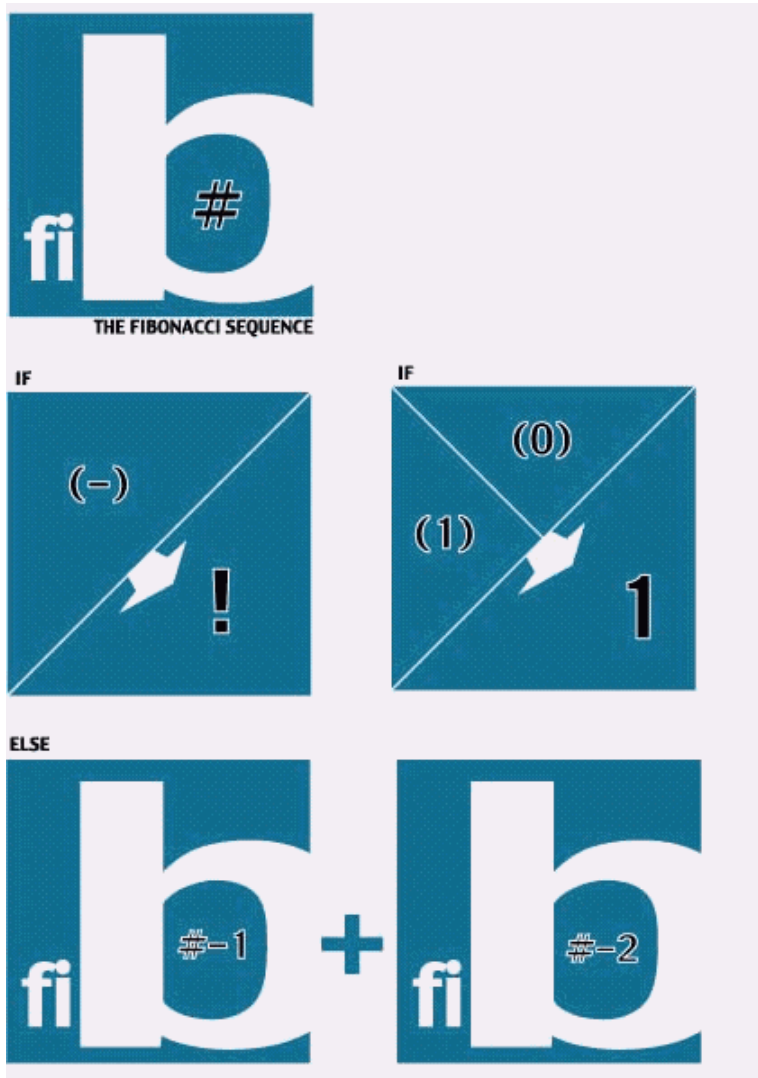
This was a replication of study II with more algorithms, more students and over a longer period of time. There were three other differences. First, the *contiguity* rating was replaced by *originality* (How much did this representation differ from the other representations?), since students reported having a difficult time understanding what exactly they were supposed to be evaluating and we felt the contiguity rating might have encouraged students who initially produced animations and other kinds of representations to converge over time to static representational styles in Study II. Another reason for this convergence might have been CAROUSEL's display of average ratings each representation received on each characteristic, compelling students to mimic styles that received high average scores in a previous assignment. So the system was revised to hide this information. To prevent students from mimicking representational styles of top students in the class, the system was changed to hide author information and present the representations anonymously for peer evaluation. Finally, a pretest was added to each algorithm representation assignment.

Sixty students in an introductory algorithm analysis course participated in this study. The study was conducted over 12 weeks with nine algorithm representation assignments: Fibonacci algorithm, Exponentiation algorithm, Binary Search Tree Node Insertion algorithm, Leftist Heap Merge algorithm, Selection Sort algorithm, Merge Sort algorithm, Quick Sort algorithm, Disjoint Set Find algorithm and Depth-First Search algorithm. Each assignment included taking a pretest and the three stages of study II.

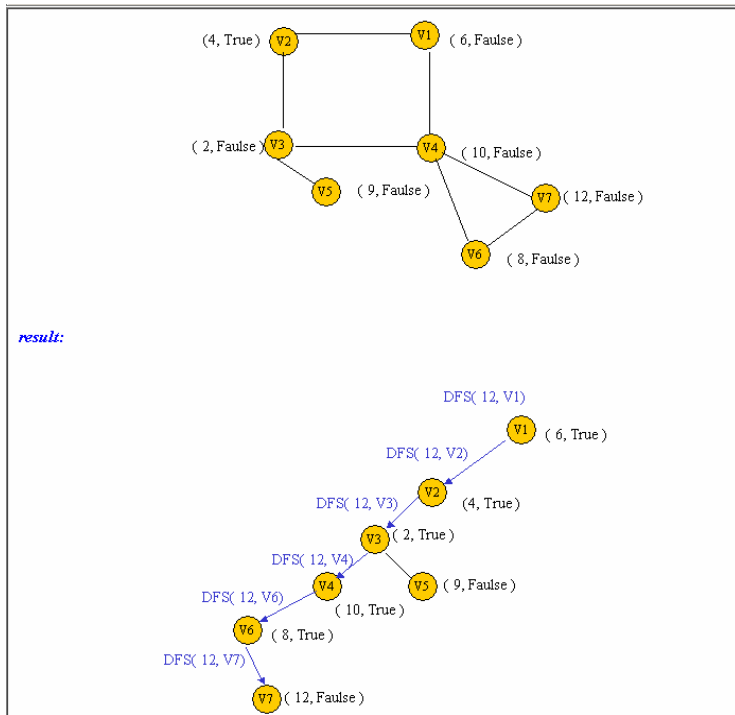
#### **Results**

A total of 196 representations were created by 36 of the 60 participants. Walkthrough style representations were prevalent (60%). However, there was not a convergence to this style over time, and there was no significant difference in the number of such representations across the assignments. There were plenty of other styles of representations that focused on aspects of algorithms that had not been represented at all in the first study, such as control flow charts, interactive representations that illustrated

algorithm efficiency, creative representations of key aspects of an algorithm (Figure 5) and graphical representations of execution results (Figure 6).



**Figure 5** A creative representation of how to recursively compute the Fibonacci Number Series.

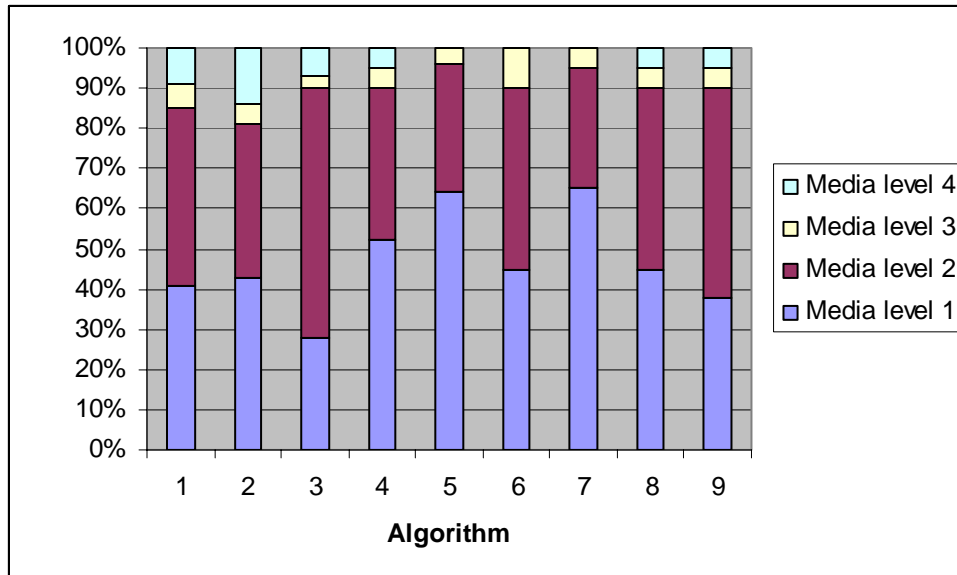


**Figure 6** A graphical representation of Depth-First Search algorithm execution.

Creating and evaluating algorithm representations aided learning. Students involved in the study improved their scores from pretest to posttest by 30% on average. Test scores were normalized by dividing the raw score by the maximum score attained by any student on a test for a particular algorithm. Some students did not create representations, as participation in any part of the activities associated with each algorithm was voluntary. The mean for the normalized posttest scores for the students who did not create a representation was 46% and for the students who did create one was 57%. The mean for the normalized learning scores for the students who did not create a representation was 25% and for those who did create one was 31%. These differences were significant: ( $F(1,327)=14.4, p<.001$ ) for posttest scores and ( $F(1,327)=3.63, p=.058$ ) for learning scores. Learning score of a student is his or her pre to post test score improvement.

When multiple linear regression analysis techniques were used to look at how whether someone did or did not create a representation and the algorithm being covered affected normalized learning scores, it was found that the model was significant ( $F(9,318)=3.37, p<.001$ ). Creating a representation had a significant positive effect on learning when the choice of algorithm was controlled for ( $F(1,318)=5.025, p=.026$ ).

The representations in this study converged less on a type, style and choice of topic than in study II. There was not a significant difference between the media the students chose to work with during each assignment and certainly no trend towards a particular style as can be seen in Figure 7.



**Figure 7** This graph shows the percentage of representations created for each algorithm at each media level in Study III. The media levels are as follows: Level 1 is text only; Level 2 contains text and graphics; Level 3 contains text, graphics, sound or 2-D animation or sound; and Level 4 uses 3-D animation or hypermedia.

Media use had a significant effect on all six dimensions of student ratings: usefulness, understandability, salience, familiarity, pleasure and originality,  $F(3, 205) = 12.9, 9.6, 9.16, 10.6, 29.07, 37.79$ , respectively,  $p < 0.0001$ .

As in study II, the experimenter rated each representation for similarity to conventional or familiar styles on a scale of 1 to 5. While the representations significantly differed in this similarity rating across the 9 algorithm assignments ( $\chi^2(32, N=206) = 62.4, p = .001$ ), there was not a trend toward the representations becoming more similar to the representations used in class. The average ratings of similarity for the 9 algorithms were 2.47, 2.96, 2.64, 2.9, 3, 2.95, 3.1, 3.25 and 2.47 respectively.

Multiple linear regression analysis techniques were used to explore how the ratings of the representations' similarity to textbook or classroom explanations were related to the average student ratings of different characteristics. The similarity ratings' relation to students' ratings of usefulness, understandability and salience were positive and significant ( $F(1, 205) = 4.56, 5.25$  and  $9.65$  respectively,  $p < .05$ ). The similarity ratings' relation to students' ratings of pleasure and originality were negative and significant ( $F(1, 205) = 4.09$  and  $16.6$  respectively,  $p < .05$ ).

## Discussion

Did students employ a variety of styles and perspectives in their representations in this study? We expected to see diverse representations that included animations as well as other media, and predicted that the changes we made in the method would prevent

convergence toward conventional styles. Indeed, the representations in this study converged less on a type, style and choice of topic than in Study II. Students used more styles and conventions in their representations than students in the first study. Their representations included animations, sound, graphics and text. They also represented aspects of algorithms that were not covered by any of the representations from the first study. However, there was not a convergence to any particular style, such as the walkthrough style, over time. Thus, students maintained their individual styles throughout the course of the study instead of converging. This is a clear difference from the previous study. We believe that convergence was prevented and representational diversity preserved by the use of the dimension “originality” instead of “contiguity” in the rating scheme, and due to hiding the authorship and ratings of representations during the course of the study.

Did students use a variety of media in their representations? How did media use impact perceived quality of representations? Consistent with our expectations, students did use a variety of media, including 2-D animations, and maintained their individual styles thorough the nine assignments. Media use had an effect on all six dimensions of student ratings: usefulness, understandability, salience, familiarity, pleasure and originality. Adding graphics to text improved the rating of all characteristics. Adding sound and/or animation to a representation with graphics and text improved average ratings of pleasure, originality and understandability. Adding hypermedia (representations produced in this study did not include 3-D animations) improved usefulness, understandability, salience and familiarity ratings but decreased pleasure and originality ratings.

Did students create representations that are similar to conventional or familiar styles? How did similarity to conventional or familiar styles impact perceived quality of representations? As predicted, there was not a trend toward the representations becoming more similar to the representations used in class. The average of all the ratings the students gave each representation was not significantly positively related to the rating of how similar that representation was to a textbook or classroom explanation, unlike what we found in the previous study. The similarity ratings’ relation to students’ ratings of usefulness, understandability and salience were positive and significant, as reported in the Results section. In other words, student ratings of how useful the representation was, how understandable it was, and how well it pointed out the important aspects of the algorithm were positively influenced by how similar that representation was to their classroom and textbook conventions. The similarity ratings’ relation to students’ ratings of pleasure and originality were negative and significant. That is, students rated a representation higher in pleasure and originality, if it differed from their classroom conventions. It seems familiar styles and conventions did influence how students perceived the quality representations. Unconventional representations were considered to be more original and pleasurable, but conventional representations were considered to be more useful, understandable and salient.

Did students learn from authoring and evaluating each others’ representations? We expected, based on the results of Study II that authoring and evaluation would lead to better learning than evaluation alone. Students who participated in this study improved

their score from pretest to posttest by 30% on average. Creating and evaluating algorithm representations clearly aided learning. While all participants rated representations, some students only evaluated others' representations and did not create any of their own, since participation in any part of the activities associated with each algorithm was voluntary. The mean for the normalized posttest scores for the students who did not create a representation was 46% and for the students who did create one was 57%. The mean for the normalized learning scores for the students who did not create a representation was 25% and for those who did create one was 31%. So creating and evaluating representations led to more learning than evaluating alone.

#### **4. General Discussion**

Research on learning from multimedia and animations in the domain of algorithms has for the most part focused on expert created animations (Naps et al., 2002). Even when investigating student created representations, researchers (e.g., Stasko, 1997; Hundhausen & Douglas, 2000) have not focused on quantitative as well as qualitative analyses of whether students learn from constructing, sharing and evaluating each other's representations. Neither have they attempted to characterize the nature of student-authored representations in terms of diversity, by focusing on the variety of static and dynamic multimedia representations students might create instead of solely on animations. These aspects are unique to the research presented in this chapter.

This research was motivated by a study in which we observed the learning strategies of computer science undergraduates studying algorithms in groups and with multiple available representations. This study revealed a tendency of students to focus on a single explanatory representation provided by the teacher as the means to learn an algorithm, even though other representations were available to them. This led us to explore the benefits of having students create, share and critique their own expository representations such as pictures and animations in two subsequent empirical studies. In these studies students represented algorithms with animations and a variety of other media and styles. Their representations included metaphoric stories involving only text, entertaining animations with sound and three-dimensional graphics, as well as more conventional text and graphics combinations that walked a reader through an example and explained the steps of an algorithm. Students evaluated each other's representations by rating them on Likert scales of six characteristics: usefulness, understandability, salience, familiarity, pleasure and contiguity/originality.

Analyses of pretest and posttest scores indicated that representation authoring and rating helped students develop a deep understanding of algorithms. Both students who constructed their own representations and evaluated those of others and students who only evaluated others' representations learned from these activities. But those who authored representations learned more than those who did not.

Use of multiple media in representations had a significant effect on the ratings representations received on all six characteristics. In general, we found that students rated representations with text and graphics higher than text only representations, and rated

representations with sound and/or animation higher than representations with only text and static graphics. But no consistent evidence indicating that the addition of more complex media further improved perceived quality of representations was found. In short, students rated media-rich representations such as animations higher than unimodal (text) and bimodal (text + graphics) representations.

Despite the positive effect that adding animation to an explanatory representation had on student ratings, and despite the initial set of student representations containing close to 30% animated representations (algorithm 1, Study II, Figure 4), students eventually converged to a static representational style (text and diagrams explaining the algorithm in a walkthrough style, similar to representations in their textbook) over the course of several weeks and algorithm assignments in the first study that looked at student-constructed representations. Thus, there was a loss of diversity in style and perspective over time. We suspected that this occurred because students might have mimicked the representational styles of top students in the class or styles that received high overall ratings, and because one rating dimension (contiguity) emphasized conformity. So we changed this rating dimension to one that emphasized non-conformity (originality), prevented students from seeing the ratings representations received in prior assignments until the end of the semester, and hid author information. As expected, convergence did not occur in the next study (Study III).

While the representations students created in Study III were more diverse, and their styles did not converge to a textbook-like style over time, a majority of the representations remained static in format and walkthrough in style. Representations with animations stayed at or below 20% for all algorithms (Figure 7). This small percentage was contrary to our expectation of higher production of animations. This expectation was based on the advantages of animations (discussed in the Introduction section) as explanatory representations of algorithms and other complex and dynamic concepts and procedures. Students produced animations at a lower rate despite the fact that animations provide opportunities for explanation of dynamic processes that are not available with static representational forms. It was surprising that an intrinsically dynamic form of representation (i.e. animation) was used so little by the student authors to explain an intrinsically dynamic process (i.e. algorithm).

There are several possible explanations. The most obvious one is that constructing animations, even with excellent software support, takes more time and effort than writing text and drawing pictures. An alternate explanation is that perhaps students felt that walkthrough style representations that employed explanations and illustrations conveyed key aspects of algorithms well enough, and the additional expressive power of an animation was not worth the cost of producing one. A third explanation is that even when the learning activities involve only representations created by themselves and their peers, students are influenced by conventional and familiar representational styles (what they see in the textbook and hear in the classroom). This influence could be seen in the representations students authored and in how they rated their peers' representations. Student authored representations received an average rating of at least 2.5 on a similarity (to conventional and familiar styles) scale of 1 to 5 in every algorithm assignment in both

Study II and III. In the third study, a majority of the representations employed a walkthrough style – walking the reader through an example illustrating the operations of an algorithm on a sample data set – that is commonly employed in textbooks and by instructors. Students in general rated these representations higher in usefulness, understandability and salience. On the other hand, representations differing from conventional and familiar styles received higher ratings of originality and pleasure.

In spite of the predominance of conventional styles, students did produce representations with diverse styles and perspectives. There were metaphorical stories, entertaining animations, creative graphical representations of the steps, executions and results of algorithms, algorithm efficiency comparisons, web calculators that showed algorithm results, and so on. Giving students free rein to create their own explanatory multimedia generated animations and other varieties of explanations that are absent in typical algorithm instruction based on textbooks and lectures.

Our findings have some pedagogic implications for teaching and learning complex and dynamic concepts. The activities of creating, sharing and evaluating explanatory representations do facilitate learning, and ought to be encouraged. Intentional design choices in instruction and supporting technology can counteract students' tendency to converge on a representation or style, and lead to improved diversity of representations in both style and content. A large percentage of representations created by students in our studies conformed to the style used by textbook authors and instructors. This is not necessarily detrimental to learning, but we suggest that it is important for students to create and/or peruse representations that employ styles and perspectives different from what is seen in the textbook or provided by the instructor. Thus for example, encouraging students to create their own animations (which are not found in textbooks or commonly employed by teachers) of complex procedural concepts may enhance learning in an otherwise traditionally delivered course.

### **Acknowledgements**

This material is based upon work supported by the National Science Foundation (NSF) under grant REC-9815016. This chapter was prepared with NSF support while the second author was serving at NSF. However, any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not reflect the views of NSF. Authors are grateful to the editors for reviewing the chapter and providing constructive criticisms that helped significantly improve the clarity and flow of ideas.

### **References**

Ainsworth, S.E., Bibby, P.A., & Wood, D.J. (1998). Analysing the costs and benefits of multi-representational learning environments. In H. Spada, P. Reimann, P. Bozhimen & T. de Jong (Eds.) *Learning with Multiple Representations*, Elsevier Science, pp. 120-134.

- Byrne, M. D., Catrambone, R. C. & Stasko, J. T. (1999). Evaluating animations as student aids in learning computer algorithms. *Computers & Education* 33: 253-278.
- Cox, R., & Brna, P. (1995). Supporting the use of external representations in problem solving: the need for flexible learning environments. *Journal of Artificial Intelligence in Education* 6 (2): 239-302.
- Hansen, S. R. & Narayanan, N. H. (2000). On the role of animated analogies in algorithm visualizations. *Proceedings of the Fourth International Conference of The Learning Sciences*, Lawrence Erlbaum Associates, pp. 205-211.
- Hansen, S. R., Narayanan, N. H. & Hegarty, M. (2002). Designing educationally effective algorithm visualizations: Embedding analogies and animations in hypermedia. *Journal of Visual Languages and Computing*, 13(3): 291-317.
- Hübscher-Younger, T. & Narayanan, N. H. (2003). Constructive and collaborative learning of algorithms. *Proceedings of the SIGCSE Technical Symposium on Computer Science Education*, ACM Press, pp. 6-10.
- Hundhausen, C. D. & Douglas, S. A. (2000). Using visualizations to learn algorithms: Should students construct their own, or view an expert's? *Proceedings of the IEEE Symposium on Visual Languages*, IEEE Computer Society Press, pp. 21-28.
- Hundhausen, C. D., Douglas, S. A., & Stasko, J. T. 2002. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing* 13: 259-190.
- Janis, I. (1967). *Victims of Groupthink: A Psychological Study of Foreign Decisions and Fiascoes*. Houghton Mifflin, Boston, MA.
- Lakoff, G. & Johnson, M. (1999). *Philosophy in the Flesh: The Embodied Mind and its Challenge to Western Thought*. New York: Basic Books.
- Lowe, R. K. (1999). Extracting information from an animation during complex visual learning. *European Journal of the Psychology of Education*, 14: 225-244.
- Mayer, R. E. (2001). *Multimedia Learning*. Cambridge University Press.
- Milgram, S. (1963). Behavioral study of obedience. *Journal of Abnormal and Social Psychology*, 67: 371-378.
- Milgram, S. (1965). Some conditions of obedience and disobedience to authority. *Human Relations*, 18: 57-76.
- Mulholland, P. & Eisenstadt, M. (1998). Using software to teach computer programming: past, present and future. In J. Stasko, J. Domingue, M. H. Brown & B. A. Price (Eds.), *Software Visualization: Programming as a Multimedia Experience*. Cambridge, MA.: MIT Press, pp. 399-408.
- Naps, T., Roessling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Kohonen, A., Malmi, L., Mchally, M., Rodger, S., & Valazquez-Iturbide, J.A. (2002). Exploring the role of visualization and engagement in computer science education. *ACM SIGCSE Bulletin*, Volume 35, Issue 2, June 2002.

Stasko, J. T. (1997). Using student-built algorithm animations as learning aids. *Proceedings of the 28th SIGCSE Technical Symposium on Computer Science Education*, ACM Press, pp. 25–29.

Tversky, B., Morrison, J. B. & Betrancourt, M. (2002). Animation: Can it facilitate? *International Journal of Human-Computer Studies*, 57: 247-262.