The AlgoViz Portal: Lowering the Barriers for Entry into an Online Educational Community

April 15, 2009

(NSDL 2009 solicitation proposal: Due April 15.)

Project Summary

Algorithm visualizations (AVs) are widely viewed as having the potential for improving computer science education. While there has been steady progress in developing, evaluating, and deploying AVs, the rate of AV use and overall impact has progressed little from the mid 1990s. Surveys of CS faculty show that impediments to successful use of AVs in the classroom include: difficulties in finding quality AVs on desired topics; difficulties in adapting AVs to a given classroom setting; and lack of knowledge on the best way to deploy AVs. Our study of roughly 450 AVs shows that too high a percentage of existing AVs are of little pedagogical value, and there is poor distribution of topical coverage. This indicates that many AV developers are not making use of lessons learned. Since effective ways are known to develop and deploy AVs, these difficulties could be overcome if this knowledge were distributed by an effective community of AV developers and users.

CCLI grant DUE-0836940, "Building a Community and Establishing Best Practices in Algorithm Visualization through the AlgoViz Wiki" has prototyped a series of online resources and initiated steps toward building a community of users and developers of AVs, with the goal of significantly increasing their use in the classroom. This NSDL project will supply critical technical infrastructure needed to go beyond these initial efforts. As a significant content provider and community within the Ensemble NSDL Pathway project, the proposed work will allow us to go beyond simply providing content and building the AV community. We will also provide an exemplar online education community that can be replicated by others.

The fundamental problem is not lack of AVs or lack of an AV collection. Rather, the need is to better focus the ongoing efforts of AV developers and users. We seek to establish a new model for online educational communities, based less on the "digital library" model of information gained by going to a site and searching. Instead, the focus will be on community-driven content through members' discussions, reviews, and ratings of content items. To gain a critical mass of community input, we must actively lower barriers to participation by incorporating information from both the content and the community into the notification streams that community members are already using. This includes both traditional dissemination mechanisms (such as websites for collections, forums, email, and online blogs and newsletters for notification) and newer notification streams (such as Twitter and social networking sites where users can subscribe to receive notification as part of their everyday online life). Thus our prototype portal (`http://algoviz.org`) will distribute information through numerous channels. As a direct outcome of this project, the AlgoViz community will better focus the future direction of AV development and use.

Intellectual Merit: Our efforts will improve the effective use AVs in the classroom. We will leverage the collective knowledge of the community to identify quality instructional materials, thus driving better AV development practices and better AV deployment. We will demonstrate a new model for digital libraries, that better supports the contributions of community-driven content.

Broader Impact: Improving on the general availability and quality of AVs can affect the educational outcomes for tens of thousands of students every year in computer science and related disciplines. Providing a clearinghouse of available AVs with community-based rating and review, and documenting best practices for their use will make it easier for instructors to adopt quality educational materials. Since the same mechanisms that work to lower barriers to information access for the AV community are likely to work for other online educational communities, our technical innovations will be directly applicable to a broad range of educational communities within Ensemble and within the broader NSDL.

1 Problem Statement

Algorithms and data structures form one cornerstone of an undergraduate computer science education. One potential technique for improving instruction in this critical area is to include algorithm and data structure visualizations and animations (hereafter referred to as "algorithm visualizations" or "AVs") into the curriculum. AVs have a long history in computer science education, dating from the 1981 video "Sorting out Sorting" by Ronald Baeker and the BALSA system [3] from 1984. Since then, hundreds of AVs have been implemented and provided free to educators, and scores of papers have been written about them [33]. Good AVs bring algorithms to life by graphically representing their various states and animating the transitions between those states. AVs illustrate data structures in natural, intuitive ways instead of getting bogged down in memory addresses and function calls.

AVs are attractive to educators, who nearly universally view them positively [21]. They are also consistently "liked" by students [10, 32]. But an important question persists: *Are they effective at educating computer science students?* There has been some debate in the literature as to whether AVs are effective in practice. Some studies have shown the classic dismissal for many technological interventions in education: "no significant difference" [10, 13, 16]. Other studies have shown that AVs can indeed improve understanding of data structures and algorithms [18, 4, 11, 9]. Many AVs exist and are widely available via the Internet. Unfortunately, the ones of high quality can be lost among the many that serve no useful pedagogical purpose (see [29, 7] and data shown in Section 4).

So we see that (a) while many AVs exist, relatively few are of true value, and (b) some AVs can be demonstrated to have pedagogical value, yet it is also quite possible to use them in ways that have no pedagogical effect. These results indicate that creating and deploying effective AVs is difficult. There is a growing body of literature that investigates how to create pedagogically useful AVs (for example, [14, 28, 21, 19, 26]). Yet, there is still much to be done before we are at the point where good quality AVs on most topics of interest are widely recognized and adopted.

Instructors report [21] that a major impediment to using AVs in class is the difficulty of finding effective AVs. We note that the need for an organized repository for AVs (with some review mechanism) has been recognized since at least 1996 [2]. Yet no significant repositories of AVs exist, and certainly none that impose quality control over their content. While the ACM Digital Library [1], IEEE Xplore [15], and CITIDEL/NSDL repository [5] are all huge collections, none appear to have significant numbers of AVs. Equally important, none provide good search tools for finding AVs. The bulk of their materials are papers (syllabi/lecture notes in the case of CITIDEL). They tend to organize by content topic (CITIDEL) or publication venue (ACM DL, IEEE Xplore). None support browsing for AVs separate from the (overwhelming) body of non-AV content. JERIC [17] and the SIGCSE courseware site [30] each maintain a collection of courseware links, including a small number of AVs. Broader courseware repositories include SMETE [31], and Connexions [6]. These are not well known within the CS education community, and they have few, if any, AVs. Further, none of the repositories mentioned here have much "web presence" with respect to whatever AVs they do contain. In all the hours that we have spent conducting Google searches for AVs, not a single AV within any of these repositories was discovered by that means.

The typical way for instructors to find AVs is to search the Internet. Assuming that a suitable AV on a specific topic exists somewhere online, we can hope that a standard Internet search will allow educators to find it. If so, this might alleviate the need to create and maintain specialized repositories or link collections for courseware in general, and AVs in particular. However, as the results from Section 4 indicate, much of what is available is of low quality and is poorly distributed among the topics of interest. Thus, instructors indicate that they are unable to find a quality AV relevant to a given topic [21], despite the fact that there are good quality AVs available on many topics [34]. Even if a suitable AV is found, the typical instructor is unaware of what is known about effective ways to use AVs in the classroom, or how the AV in question can be best used. The learning curve currently involved in selecting and adopting an AV deters many would-be users.

The bottom line is that AVs have not improved their penetration into CS courses beyond where they were in the mid 1990s. This is despite the fact that access to computers is now ubiquitous in the classroom, making courseware use in classrooms noticeably easier than in the 1990s. It is also despite the fact that the AV research community is making steady progress on understanding how to effectively create and deploy AVs. To quote [19]:

> Previous surveys show a significant disconnect between the intuitive belief that visualization enhances a student's learning and the willingness and ability of instructors to deploy visualization in their classrooms. A key impediment to the adoption of visualizations by instructors is the time required to learn, install, and develop visualizations and then integrate them into a course. Additionally, there is also a perceived lack of effective visualization development tools and software.
>
> Whereas studies have begun to show the conditions under which visualization enhances student learning, the overall educational impact of visualization is and will be minimal until more instructors are induced to integrate visualization techniques in their classes.

A key reason for this situation is that there exists no effective mechanism to transfer the fruit of AV research activities into the hands of developers and instructors. Since there is no organized AV community, there are no established best practices for developing AVs, no good collections of information about existing AVs and how to use them, and little shared information about where developers can best focus their efforts. A collection of existing AVs (such as we provide in the AlgoViz Wiki [34]) is necessary but not sufficient, since the fundamental problem is not the lack of AVs or lack of an AV collection. Rather, we need better focus for the ongoing efforts of AV developers and users. To do this, we must go beyond the "digital library" model of information gained by visiting a site and searching. We need to focus on the value-added content provided by an active educational community through discussions, reviews, and ratings of content. In this way instructors will not merely find an AV. They will find how to use the AV most effectively with less of a learning curve.

2 Solution Overview

Our overarching goal is to build a community of developers and users of AVs who together will improve CS education through the use of AVs. We have support for our initial efforts at community building through the NSF CCLI program under grant DUE-0836940, "Building a Community and Establishing Best Practices in Algorithm Visualization through the AlgoViz Wiki." That project focuses primarily on non-technical aspects of building our human capital, including developing a steering committee of senior researchers within the community, establishing an awards process to recognize good-quality AVs, establishing forums for communication, and expanding on the AlgoViz Wiki and the OpenAlgoViz SourceForge site.

The goal of this proposed NSDL project is to provide technical infrastructure to collect and disseminate community-driven content. The centerpiece will be the AlgoViz Portal, whose purpose is to unify our existing collection of online resources. Our existing resources through CCLI are not sufficient to go beyond a primitive linking of the existing sites (the Wiki, the SourceForge site, and community forums). Our plan with additional support from NSDL is to extend the portal to be a full-service site that serves the online portion of the needs of our community, and which can further serve as a model for similar online educational communities. The most significant aspect of this is that the portal will not simply be a place that users "go to" for AV information. The problem with that approach is that users have to explicitly make the effort to come to our site to determine when new (and relevant) content has been provided. **Instead, this project will focus on the use of new online technologies that target collecting and organizing content from the educational community, along with the means to convey that information back to the community, through channels that they already use on a daily basis. In that way, we hope to reduce the barriers to getting needed information out to potential users.**

We will develop the portal infrastructure needed to collect and organize community-driven content. As described in Section 4, we have already created the AlgoViz Wiki, the OpenAlgoViz SourceForge site, and community forums. The AlgoViz portal will begin with one-stop credentialing to access all of our resources through OpenID. We will engineer connections that allow the community to easily incorporate other sources of information into our collections, such as images and video to describe AVs (linked to host sites such as YouTube). This will improve the value of our collections. We will provide online infrastructure to support nominations and voting in support of an awards process that will raise awareness of quality AVs.

We will develop the portal infrastructure needed to disseminate community-driven content. The original goals of the CCLI project were to allow AV users to connect and interact, rate and review available AVs, and share activities and plans for how best to use AVs. The NSDL project will go the next step, by linking the AlgoViz resources back to other social networking support such as Twitter and Facebook. The more ways that AlgoViz interconnects with users' existing social networking and notification streams, the more likely that they will participate in this community. We will "push" content to the community through the social networking sites, whose key functionality in this context is that they channel disparate streams of notifications to users. In this way, users who choose to receive updates about the AlgoViz community will automatically stay up-to-date on what we have to offer.

We will provide technical support for a community of developers of AVs. While many good AVs are available, the need for more and higher quality AVs continues. Many existing problems can be mitigated by the existence of an easily accessible resource that provides AV developers with a way to communicate with users, and that carries the "corporate knowledge" of the user and developer community. We will demonstrate via the AlgoViz catalog how developers can provide connections to YouTube videos and screenshots to help users find useful AVs, and provide notification to interested users through the AlgoViz portal. We will provide information on evaluation of AVs. In short, we will connect developers with their users and with each other.

We will log the level of use for the various technologies that we employ. One important aspect of our efforts is that we do not know which of the many possible channels of information will prove to be the most popular with users. Fortunately, the technologies involved lend themselves to tracking their level of use. So, for example, we will know how many users receive notifications on AlgoViz via Twitter, and precisely which AlgoViz services they request. We will be able to report this information back to the broader educational community.

We will provide technical support for others to build similar educational communities. We will document everything that we do with the explicit intent of making it easy for other communities to copy our solutions. Most of our technology will be application of off-the-shelf software that is tailored to our needs. Where we need to develop custom solutions to join things together, we will make this material available via the OpenAlgoViz SourceForge site. We will provide a "handbook" for building an educational community, thereby broadening impact beyond our own community.

We will support Ensemble as content providers, technology innovators, and community builders. Being among the first group of content providers to the Ensemble NSDL Pathway project [8], and one of the most active communities, we have the opportunity both to use Ensemble support to leverage our efforts, and to add value to the entire Ensemble enterprise. We are helping to build the infrastructure within Ensemble that will provide access to the data collections contained in the AlgoViz Wiki and the SourceForge site (among others). These efforts will provide to Ensemble users effective means to browse our collections of AVs, AV development toolkits, the annotated bibliography, and other resources in the AlgoViz Wiki. Our infrastructure is then available to other communities both within Ensemble and within the broader online education community.

3 Proposed Work

3.1 A New Model for Educational Content Providers

The central idea of this project is to prototype a different use model for NSDL repositories. The old model is a "library" approach, where people come to search a collection. The problem with this model is that users do not know when content of interest has been added to the site, or otherwise get relatively little value from it. Couple with this the recognition that it is difficult for repository providers today to improve on Google searches for online content. The content is valuable, but the repository "services" typically can be provided by Google automatically.

The new model is to have the community add value beyond the base content. This comes as a byproduct of direct communication between community members. While forums are one obvious method for this, for many communities there is a deeper communication in terms of community evaluation of aspects of the content (community ratings and recommendations for content entries), and discussions about the content (sharing experiences on how to make the best use of it). This user-based, value-added material is not "content" under the old model, and is generally not captured by the metadata associated with a collection. Yet, in many educational contexts, the user feedback on items in the collection is just as important as the collection itself. The reason is that in educational communities, a major concern for instructors is deciding whether a given educational resource is of good quality, and how to use it.

Amazon is an example of a "collection" that incorporates significant value added by the community, in the form of ratings, reviews, and commentary. While Amazon's catalog of items for sale is intrinsically valuable, a great deal of value is added when users can see how a large group of other users rated a given item, or can read a review about an item. However, a key difference between Amazon and a typical educational community is scale. If only 1-2% of Amazon's community ever provides content, that is enough to generate a rich database of ratings and reviews. The typical educational community cannot reach critical mass on user-added content with only a 1-2% participation rate. However, there is reason to hope that the typical instructor interested in (for example) algorithm visualization is intrinsically more motivated to participate in providing value to the community than is the typical Amazon customer. The key is to make it sufficiently easy for users to do so, so that enough members do participate.

We believe that one way to increase participation is through notification mechanisms. We cannot expect instructors to routinely come to our site to determine if new content is available, since the overhead for doing so is too high relative to the likely value received. Instead, we need to notify members of our community whenever "interesting" content appears. To succeed, we need two things. First, this needs to be done in a way that is natural to any given individual, in that it fits with their normal information flow. For some that can be traditional email. But for many people today, notification streams come through social networking schemes such as Twitter or Facebook. Second, each user must have sufficient control in determining what they consider to be "interesting" so that they do not receive too much unwanted information.

There is a reason why many people opt to use these social networking schemes for notification, rather than receive all such information via more traditional email. That is because they are better tuned toward integrating many streams of small, targeted notification (a very short message, typically). As a consequence, if we are going to send out notification streams based on changes to the AlgoViz site content, the notices need to be fine grained, and targeted to only those recipients that want any given notification.

A key question that we must answer is: What services should an online educational community provide? Fundamentally, we seek to provide communications, discussion, search, and feedback (voting, rating, reviewing). The techniques for providing these services must be considered experimental within the context of an online educational community, since there are few existing models today. Thus, we cannot determine in advance which of these services, and which technologies that provide them, are going to be most successful. Therefore, key aspects of our work must be to (1) perform many experiments in how to provide information, and (2) track the performance of each such experiment, and update our services accordingly. Fortunately, much of the technology involved (such as Twitter and Facebook) provide good tools for tracking the level of use. We will regularly monitor the level of use for all services that we provide. And we will regularly document how we go about providing them, and routinely report their success rates, so that others can adopt our approaches as desired. One of our goals is to provide to other Ensemble communities information on what services are worth providing and how this can best be done. Besides usage tracking, another source of information that we will use to determine user satisfaction will be routine "mini surveys" to our users. These are brief (single question) surveys on specific aspects of use.

As indicated above, for our notification approach to succeed we must provide multiple levels of notification. Users need to feel in control of the level of notification, so that they can be assured of receiving only information of value to them. For example, a given user might want to know when certain AV topics are updated in the catalog, or new posts are made to certain subforums at the site, and not others. Thus, we will build in an extensive "profiling" mechanism whereby users can easily select what topics are of interest.

3.2 AlgoViz and Ensemble

The key proposed activities are to (1) build infrastructure to extend the value of the AlgoViz portal and related materials, and their contribution to Ensemble, and (2) support the adoption of our community-building infrastructure within the Ensemble meta-community, and beyond. We have agreements with the directors of the Ensemble project regarding this work. Support letters from Boots Cassel and Ed Fox, directors for Ensemble, are included with this proposal.

AlgoViz content collections will be modified to support the Open Archives Initiative protocol for metadata harvesting [23] so that Ensemble can automatically collect data from AlgoViz collections. The AV catalog is already structured as a series of key:value pairs. While we are reanalyzing our content to better support the Dublin Core, overall the structure of the catalog should require little revision to support harvesting. Where necessary, we will restructure content of our other collections to support harvesting, such as our research bibliography collection and the listings for AV development tools and development projects.

While we seek strong integration of our content within Ensemble, there are unique aspects to each educational community that should be preserved and capitalized on. For example, our initial choice of a Wiki (as opposed to a traditional website) for the AlgoViz site was driven in part by our goal of building community. Meaningful mechanisms for harvesting content from wikis back to Ensemble will need to be worked out. We now recognize that our support for community must go far beyond simply providing a Wiki site. Our focus on value added to the content by the community itself is one that we believe will serve many other Ensemble communities well. Harvesting this community-driven content presents new challenges for Ensemble.

One example of the community adding value is a community rating system for AVs in the catalog. Currently, quality assessment information in the catalog comes in four ways. 1) An overall ranking (Recommended, Has Potential, Not Recommended) for each AV. This is currently given by the Wiki "editors." 2) Ratings by users on a five-star basis, similar to product ratings at a site such as Amazon. 3) Description and review information for the AV. 4) User comments. While the present incarnation of the catalog gets the bulk of its review information from the current catalog editors, in the future the bulk of this material must come from the community to reach its true potential, through users submitting reviews and comments. This in turn will drive recognition of AVs and other content items that can then be sent as "notifications" to community members through their current notification streams, which will in turn prompt community members to add more value to the community through their own ratings and comments.

Other mechanisms for supporting the community through the AlgoViz Portal include:

- A rich set of discussion forums that allow instructors and developers to engage in public, threaded exchanges of information and views. If forum activity reaches critical mass, it could become a focal point for establishing a real sense of community presence. Thread topics can be indexed within Ensemble, and users can receive notifications of topics they have flagged.
- The annotated bibliography (see Section 4.2). We already have a review form mechanism in place that allows the community to contribute to this resource. The bibliography will naturally be incorporated into Ensemble as well.
- Information on the state of the practice for using AVs effectively. Such information can be provided through the Wiki by synthesizing existing research. It can be cast as a series of articles or records to allow indexing in Ensemble and suitable notifications to users.
- Resources for educators teaching courses on AV design and implementation. Already two instructors (Cliff Shaffer and Tom Naps) have taught graduate seminar classes where the students provided reviews and other updates to the Wiki as part of their classwork. We will encourage other instructors to make use of the community resources and have their students provide content.

3.3 Support for Developers

An important tool for AV developers is good examples of existing AVs, including source code. As part of the CCLI project, we have established the OpenAlgoViz site at SourceForge.net. This SourceForge site seeks to build a collection of AV implementations. They are open source, and are contributed by existing developers in the community. So far, we have contributions from Virginia Tech, the Jhavé [20] and GAIGS [22] projects from Tom Naps, and internationalization libraries from Guido Rossling. We are actively soliciting many more contributors. The site has at least two purposes: 1) Provide reference implementations for good quality AVs that others can use to see how things should be done. 2) Provide a hosting site for AV contributors. Potentially, our SourceForge site could become a major repository of AV source code in its own right.

Our examination of the software development practices employed by AV developers [7] indicate that much can be done to improve the state of the practice in AV implementation. The key is providing relevant information to developers. The AlgoViz Portal has the potential for providing many resources that could prove critical to efforts to improve AV design and implementation. This includes information on topics such as how to license sourcecode (most AVs currently provide sourcecode, but often without any license provided, which make reuse difficult); development tools and best practices; advice on which topics are most needed for further AV development; and advice on how to evaluate AVs.

Perhaps the most valuable resource that we can provide to AV developers is the opportunity for contact with AV users. This can occur directly through forum discussions, or indirectly through ratings and reviews.

4 Preliminary Results

We have built a substantial collection of information. But merely documenting the prior efforts of AV developers though a digital library collection of AVs is not sufficient to substantially increase use of AVs in the classroom. However, a base of useful content is necessary to provide incentive for users to join the community in the first place. Our existing collections of information serve as a valuable resource and the basis on which to develop community-driven content.

4.1 The Algorithm Visualization Catalog

Since Spring 2006 we have made a significant effort to catalog as many existing AVs as we could on topics related to data structures and algorithms. The results are housed at the AlgoViz Wiki [34] (the AlgoViz Wiki itself is further described in Section 4.2). We have developed the most extensive collection of links to AVs currently available, with roughly 450 links as of April, 2009. While it is by no means complete, this collection can serve as a representative sample of the total population of AVs accessible on-line. A number of interesting research questions can be addressed by analyzing the contents of the catalog. In this section we present some of our initial findings [29, 7].

How did we find them? We began with a list of all AV systems that we were aware of from our knowledge of the field. We developed a topic list based on our experiences teaching relevant courses. We considered what search terms would be most productive for locating AVs via Internet searches. We then performed searches using Google. We examined the pages we found to try to locate other AVs, since developers of a given AV often have others available. Whenever we came across a page that had links to collections of AVs, we would include any new ones not yet in our collection. We speculate that we have so far captured about half of what is publicly available, and have plausibly located the vast majority of easily found, better-known AVs. The search procedure models the process a diligent educator might employ for locating an AV. If an AV cannot be found by our method, it seems unlikely the typical instructor would locate it via standard Internet search engines. Our team is still actively collecting new links.

How many are available? The collection now contains links to around 450 AVs. Many of these are individual applets or programs, but a significant fraction appear as parts of integrated visualization collections (typically, individual applications that embody 5–20 distinct AVs, or toolkits that distribute a collection of 5–20 distinct AVs as a unit). If a given applet or program contains multiple AVs (for example, a single Java applet that embodies separate AVs for multiple tree structures), it is counted multiple times—once for each distinct visualization. There are well over 200 distinct programs or applets. These totals do not include older systems like XTANGO and Balsa. We have not yet integrated most non-Java AVs into the main catalog, although the Wiki tools collection includes pointers to these systems.

How are they implemented? Virtually all AVs and AV toolkits that we have located with creation dates since the mid-1990's have been implemented in Java. About two thirds of available AVs appear as applets directly embedded in web pages or initiated using Java Webstart. Most of the rest are Java applications that must be downloaded and opened locally, while only about five percent are implemented any other way. These include OS-specific binaries (e.g., Microsoft Windows), Shockwave, Flash, or JavaScript. These numbers are somewhat biased. There is a tendency for us to search for applets, since these are easier to find (including "applet" as a keyword often results in locating additional AVs on Google searches). AVs distributed as applets—i.e., that can be embedded directly in web pages—typically get more attention from potential users, since they need not go through the additional step of downloading and unpacking. We do a significant amount of cross checking by capturing links from AV link collections that we find to insure that our data collection process does not focus unfairly on individual applets. However, our catalog is currently known to be deficient in that it does not yet include many pre-Java AVs.

How are they disseminated? We found almost no AVs in large, organized courseware repositories. Many AVs are cataloged by link sites, meaning sites that (like our Wiki) attempt to link to collections of AVs that the site managers have considered worthy. Most of these link sites are small, perhaps linking to 20 favorite AVs for some course or textbook. There have been other efforts to produce comprehensive catalogs of AVs (the Hope College collection [12] and Guido Rößling's animation repository [25] containing approximately 230 AVs).

Who makes them? Roughly 40% of AVs are essentially single efforts by their respective authors. About 15% are provided by "small shops" that have created 5–10 AVs, often as individual Java applets. These might have each been created by the same individual over some number of years (typically a faculty member who is teaching relevant courses), or they might have been developed by a small number of students working for a faculty member. Nearly half of the AVs in our catalog are created by groups that have developed 15 or more AVs. Most such AVs are part of a system, but a significant minority are parts of collections we cannot characterize as a system. Note, though, that once a single AV from a collection or small shop was added to the Wiki, the entirety of that collection would typically be discovered and added. This ensures that we probably have AVs available from groups with at least a little visibility. The AVs we missed are presumably heavily skewed towards one-offs.

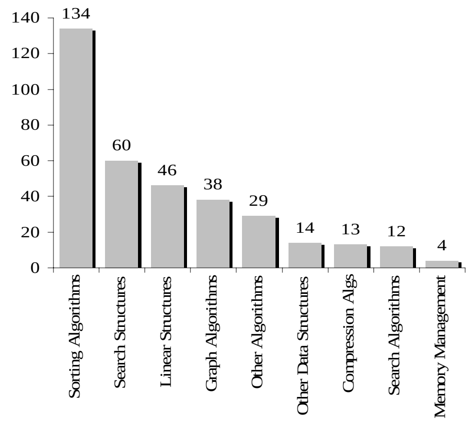

Figure 1: Histogram of major categories for AVs in the Wiki catalog.

What is the content distribution? Since we have limited resources, we have focused our study to topics typically covered in undergraduate courses on data structures and algorithms. While this concerns mostly lower division material, we also included some upper division topics like computational geometry, $\mathcal{NP}$-complete problems, dynamic programming, and string matching. A histogram of our top-level categories for grouping AVs is shown in Figure 1. We see that there is wide variation in coverage. Over 10% of all AVs that we have found are on linear structures such as stacks and queues, even though these probably present less difficulty to students than many other topics. Over a third of all AVs are on sorting algorithms. Sorting is an important topic in the undergraduate curriculum, but it is certainly being given too much attention by creators of AVs. Further, many of the sorting AVs are variations on the classic "Sorting out Sorting" video, and just show bars being swapped. In contrast, most specialized and advanced topics are poorly covered. There is certainly room for new AVs.

What is their quality? A significant fraction of the collected AVs either cannot be made to work easily, or appear to have no pedagogical value. Here, "has no pedagogical value" means that the AV gives the user no understanding whatsoever of how the data structure or algorithm being "visualized" works, and so will be of little use in the classroom. A growing body of work suggests observable characteristics which contribute to AV effectiveness [27, 28, 14, 21], and this work has guided the apportionment of labels in the catalog. We provide a coarse rating structure of "Recommended," "Has Potential," and "Not Recommended." While not all entries have been rated as of this writing, nearly two thirds have been given ratings. Slightly under half of these are ranked as "Has Potential" while over one third are ranked as "Not Recommended" and less than one fifth are ranked as "Recommended."

Even the better AVs tend to have serious deficiencies. Perhaps half of all AVs are actually animations—essentially, real-time-rendered movies with little or no meaningful interaction. While some animations are useful, in general, users of animations are relegated to being passive observers with no control over pacing (aside from animation speed), the data being processed, or the operations being conducted. A different type of deficiency often occurs with AVs of tree structures. Typically, these will show the tree that results from a user-selected insert or delete operation. Rarely do they illustrate at all, let alone effectively, how the insert or delete operation actually works. It is exceedingly rare that any AV goes beyond explaining how something works, to explaining implementation pros and cons or performance characteristics.

1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007
11	10	11	16	5	12	9	3	5	9	8	8

Table 1: Year of last change, by project

When were they made? Some well-known systems were developed in the early 1990s for creating AVs. However, most of these are now no longer available or so difficult to access due to changes in computer operating systems that they are not currently a factor in education. Considering primarily the development of AVs since the mid-1990s (i.e., Java), AV development as indicated by last-changed dates appears to be continuing relatively evenly over the years. Figure 1 shows a count of the last-change dates for those AVs in the catalog for which the information is available. These counts are "by project" rather than by individual AV, in that if a given AV or AV system provided multiple visualizations, then it only counts once in the histogram. Note that AV developers rarely do a good job of documenting their development histories, and so there is a lot of inference and some guessing on these dates. While these data show a dip during 2003 and 2004, and some slight reduction in numbers in the past three years, current production still does appear steady.

A decline in the creation of new AVs certainly cannot be explained by saturation of either topic coverage or content quality, since both are lacking. Ongoing projects appear not to be attuned to gaps in topical coverage. We speculate that students were readily available to create AVs during the Java boom of the mid to late 1990s, when Java and applets were new and the "in thing" for students to learn. But now there are other "in things" competing for students' attention. The nationwide drop in computer science enrollment might also mean that there are fewer students available to do such projects. This is not necessarily a bad outcome, as such student-driven "one-off" implementations tend to be of low pedagogical value. Fortunately, larger projects appear to be ongoing at similar rates to 10–15 years ago.

Will we find them again? Like everything on the Internet maintained by individuals, AVs have a turnover rate in their accessibility. To measure this, we track the status of the links on our Wiki. Each week, a snapshot of each link's accessibility is recorded. On July 31, 2006 the catalog had 251 links. By February 26, 2007, 40 of those had disappeared and 13 became redirects. Thus, for this seven month period, 21% were lost or moved. On November 27, 2006, the catalog held pointers to 382 links, 23 of which were not working. By February 26, 2007 (three months later), 84 were lost and 4 became permanent redirects. For this period, 61 of 359 (17%) of the links were lost or moved, while 3% reappeared. These numbers expose an extremely high turnover rate, and presents a major challenge to creating a useful index of AVs. Fortunately, many of the lost links were recoverable with some searching—they had been moved but their old links were not redirected. In June, 2008, of 414 entries in the catalog, 352 were found, 45 were redirected, and only 17 were lost (a loss rate of just over 4% over the past year). These better results are in part due to our efforts to chase down and restore bad links when possible. In February, 2009, an additional 18 AVs (about 5%) were lost, but 14 of these (12 from one project) were recovered by searching for them. In each case, the URL had changed with no redirection.

4.2 The AlgoViz Wiki

A Wiki is a user-editable website, usually supporting a simplified markup syntax and directed towards a particular topic. The AlgoViz Wiki is implemented using the MoinMoin Wiki Engine [35]. MoinMoin was selected because Dr. Edwards (consultant to this project) had already evaluated several Wiki packages and settled on MoinMoin for his other work. He administers the AlgoViz Wiki on his existing Wiki "farm." The Wiki concept is preferable to a standard HTTP server for this project because it allows for the community access and editing we desire (see Section 3 for more details).

Analysis Tools We have created various tools to automatically extract data from the AlgoViz Wiki, including a database query tool, a link checker, a counter for updating the top-level catalog counts, and a validator for checking the completeness and correctness of catalog entries. The query tool [24] loads each visualization entry and reformats its data as a row in a CSV (comma-separated values) dataset. The dataset is timestamped and archived. When the user selects a dataset and a column label, the tool returns a histogram of the visualization entries by that column. Our link checker searches the Wiki's catalog, recording the HTTP status code for each link. This allows us to track AV disappearances over time. The link checker's output (page name, URL, and status code) is stored as an XML file with a time and date stamp and weekly runs are archived to create an historical record.

Other cataloged software In addition to AVs, the Wiki contains a catalog of toolkits for authoring AVs and a small collection of visual debuggers. Many of the toolkits are associated with collections of AVs, so these are represented multiple times: once for each AV they contain and once on the page describing available toolkits.

Bibliography Another major component of the Wiki is an annotated bibliography containing over 80 research papers about AVs. One of the hardest things to do in any field of study is to acquire the knowledge and the zeitgeist of the field's community. We hope that the bibliography will provide a starting point for newcomers to the field and a repository of important work to which practitioners can refer. Many of the papers listed have summaries, and nearly all have BibTeX citations for easy referencing. A topical index helps to locate a particular paper.

Current Wiki traffic We have been collecting web server traffic statistics. After factoring out search engine spiders and other chaff as best we can, our data indicate that the Wiki serves approximately 30-40 page requests per day to Non-Virginia Tech users, with a steady rise in accesses. In the six month period ending with September, 2008, average page request rates had increased 122% over the same period the previous year, but this is largely due to initial advertisement for the site. February 2009 data show a 10% increase over the the prior year, a better reflection of a steady increase in use.

Thus, the Wiki has already established a growing degree of external visibility. Our greatest source of referring traffic comes from Wikipedia, where some articles on data structures refer to the Wiki. This suggests that one strategy for increasing visibility will be to increase the number of relevant articles at Wikipedia that reference the AlgoViz Wiki. Our second largest referral site is Google, presumably coming as the result of Google searches. In addition to referrals on AV topic searches, the AlgoViz Wiki is now the first listing for the Google search "Algorithm Visualization."

4.3 AlgoViz CCLI Phase 1 Project

Our ongoing CCLI project lays the groundwork for our community-building efforts. We began by establishing a steering committee to guide policies and the technical foundations of the project. This committee is drawn from people who have been active in the AV field, including AV developers, researchers, and participants in ITiCSE working groups related to AVs [21, 19, 26]. We held a one-day workshop for the committee in conjunction with SIGCSE'09. The most significant outcome from this meeting was a near-term "road map" of activities, some of which have lead to this proposal.

We have begun a portal project, whose goals are to support a single point of access to our various online resources (the Wiki, the forums, and the SourceForge site). This is the framework that this NSDL proposal intends to work on.

We have planned a major community-building activity for the next year, which is to solicit from the CS education community a set of nominations for high-quality AVs. We will then establish a voting procedure, and solicit CS educators to vote for the best AVs. We will arrange an award ceremony at SIGCSE'10. In this way, we anticipate raising visibility for our project, and draw attention to the existence of high-quality AVs. The portal will support the voting process.

5 Dissemination

The impact of this proposal goes beyond the typical dissemination of results via publications in conferences and journals, and workshop or tutorial presentations. AlgoViz and Ensemble are all about dissemination of information through their online presence.

Additional dissemination of results and knowledge from this work will come in the form of publications and workshops/tutorials at national or international conferences. We expect to continue publishing the results of this research at SIGCSE, ITiCSE, FIE, CCSC and/or ASEE, all recognized conferences that cover advances in computer science education. Relevant journals include *ACM Transactions on Computing Education* (previously, *JERIC*), Elsevier's *Computers & Education: An International Journal*, *IEEE Transactions on Education*, and Taylor & Francis' *Computer Science Education*. As we further develop "best practices" for AVs, it would be natural to give a tutorial on AV design and implementation at (for instance) the annual SIGCSE conference. We are well positioned to do so, as Dr. Shaffer has developed relevant materials for a graduate-level advanced topics course that he taught on AVs in Spring 2008. We held a birds-of-a-feather session related to development and use of AVs at SIGCSE 2008, and again at SIGCSE 2009. At SIGCSE 2010, we plan to hold an event to recognize award-quality AVs. It would be appropriate to write a book on AVs for the broader community based on the results of this project. We will also engage the SIGCSE listserv—not just advertising the Wiki there (which we have done), but also initiating and supporting discussions on various aspects of AVs.

6 Evaluation

Our evaluation measures come at two levels. First, we have a series of direct measures related to the AlgoViz Portal and associated sites. These measure things such as level of traffic, number of users, etc. They serve primarily as formative measures for site development. Goals are meant to be reached by the end of the two-year NSDL project.

- **Notification followers:** Notification mechanisms such as Twitter and Facebook provide statistics to track the number of people who sign on to receive notifications. We will track the numbers and the specific notification subareas that users sign on for. We will do this for each notification technology. Goal: At least 200 users receiving notifications.
- **Site traffic:** Tracking the total page hits, number of referring sites, and other access statistics for our assets including the AlgoViz Portal and the AlgoViz Wiki. By subdividing access

metrics based on the section of the site being used—the AV catalog, AV reviews, bibliography, developer forums, and pages on our own AV development work—it will be possible to gauge participation levels for both educators using AVs and developers working on new AVs. Goal: A 10-fold increase in site traffic over present rates.

- **Ratings and reviews:** In addition to basic traffic, we will also track the number of ratings and user reviews provided, together with the number of unique user accounts that participate in ratings and reviews. Goals: Complete all catalog entries with "editor" ratings and descriptions. User ratings for at least half the catalog entries.
- **Outbound links:** To measure the degree to which educators are using either Ensemble or the AlgoViz Portal to find AVs for their own courses, we will also measure the number of click-throughs on outbound links—that is, the number of times links directed to each external AV site are clicked by users. In addition to providing insight into community impact, this measure will also provide useful guidance on which AVs are most popular and are receiving the most attention. Goal: Evidence that at least a quarter of cataloged AVs have been accessed.
- **Inbound links:** We will track and measure the number of referring sites–that is, sites out on the Web that link directly to Ensemble or the AlgoViz Wiki. Goal: A 10-fold increase in inbound links over current levels.
- **Registration counts:** We will measure and track user demographics via registration information that participants create. Goal: At least 100 contributors.
- **Forums:** In addition to basic registration information, we will also track the number of registered users who participate in the various user forums, both for reading and posting. Goal: At least 100 contributors.

While significant, the direct measures above indicate only the health of our community sites. Arguably they measure the health of our online educational community. But they are not measures related to our ultimate goals, which are to increase the level of use for AVs in the classroom, and to improve the quality of AVs being developed. Those can be viewed as summative measures for our project. In addition, we also have taken the task of impacting other educational communities, through disseminating our community development methodologies.

- **New AVs created:** Throughout the project, as new AVs are discovered and cataloged, we will also track whether or not their creators are registered with the AlgoViz project. We will also track the new AV link contributions that our users provide. Over time, these measures will provide a clear picture of how strongly connected the developers of new AVs are with the AlgoViz project and its resources. Goal: Acquire clear baselines for current activity levels, and show a clear upward trend in activity after two years.
- **Reported AV use:** We will continually collect information about the level of use of AVs in Computer Science courses. We will collect information from the SIGCSE email list, the SIGCSE annual conferences, and our own forums and contacts. Goal: Acquire clear baselines for current levels of use, and show a clear upward trend in use after two years.
- **Impact on other Educational Communities:** We seek to disseminate our technical innovations and community-building successes to other communities within Ensemble. Goal: Have at least three other educational communities adopt our community-driven content methodologies by the end of the project.

7 Work Plan

The major budget request for this project is graduate research assistant support for a total of 24 months. This resource will be allocated to the following tasks at the indicated level of effort in terms of total months worked over the life of the project. Note that a timeline is not useful here as many of these activities are intermittent and interleaved.

- Initial access to notification mechanisms, tracking, and updating. 6 months.
- Develop customization for Ensemble access to the AV collections. 4 months.
- Ongoing support for the SourceForge site and forum. 2 months.
- Support for the catalog and AlgoViz collections, including maintaining harvester connections with Ensemble. 3 months.
- Set up automated collection of measures for site traffic (for the Portal, the Wiki, the SourceForge site, and the forums) and conduct other data collection as described in Section 6 for the purpose of project evaluation. 3 months.
- Provide and maintain all the mechanisms necessary to support community-based rating and reviewing, including awards voting and community interaction with the AV catalog and the bibliography. 3 months.
- Documentation of community site building technologies, and support for other Ensemble educational communities. 3 months.
- Other efforts related to advertising the Wiki and SourceForge sites to potential users. 2 month.

8 Broadening Participation: Minorities and Undergraduates

This project is a natural for recruiting undergraduate students for research projects. We have had many students work on AV-related activities over the years. For example, we had two undergraduate independent study students each year for the past three years, and at least one intern each year during the past three years. At Virginia Tech, we have had success in broadening participation in computing research by minorities. The PI has a strong track record working with female and minority students at VT (including two of the undergraduates doing independent study in the past three years on AV-related projects and two minority summer interns). The Department of Computer Science holds an annual research symposium for undergraduate research students. This is part of the Virginia Tech Undergraduate Research in Computer Science (VTURCS) program. The PI often requires that undergraduate students that work with us participate in VTURCS with a poster presentation. Our consultant, Dr. Edwards, is a co-director of VTURCS.

The CS department generally has had success recruiting African-American female students from neighboring Historically Black College and Universities (HBCU). We have three recent female African American PhD's in computing, (Tracy Lewis, at Radford University, Cheryl Seals at Auburn University, and Jamika Burge at Penn State). We currently are recruiting African-American students from Auburn University, as part of our collaboration with Dr. Juan Gilbert. The success of women and minorities in Computer Science at Virginia Tech are detailed at `http://www.womenandminorities.cs.vt.edu/`. 10 out of 37 PhD graduates over the past 3 years have been female. We currently have four Latino students in our graduate program (2 PhD students and 2 Masters). One of them, Ricardo Quintana-Castillo, is a recipient of an NSF Graduate fellowship.

Within our own CCLI project, our current graduate students are A.J. Alon (minority) and

Monika Akbar (female). We routinely participate in the REU program through our Center for Human-Computer Interaction. We are scheduled to have a minority undergraduate intern this summer working on AV-related projects.

9 Key Staff

PI: Cliff Shaffer, Professor of Computer Science at Virginia Tech, has interests in algorithms and data structures, visualization, problem-solving environments, and educational use of computers. He was the PI and lead designer for the highly successful **Project GeoSim** in the mid-1990s, which developed a series of simulations for geography education. He has extensive experience with designing problem solving environments for a number of scientific and engineering applications. He is currently a member of the CS departmental undergraduate program committee, and was chairman of the graduate program for eight years. He is the designer and course coordinator for CS2606: Data Structures and File Processing, and wrote the textbook used in that course. Role: Dr. Shaffer will provide overall project management, lead the efforts to create AV developers' guides, establish and interact with the steering committee, and oversee day-to-day management of the Portal, Wiki, and SourceForge site development efforts.

Senior Personnel: Stephen Edwards, Associate Professor of Computer Science at Virginia Tech, has interests in component-based software, automated software testing, and educational uses of computers. He is the lead designer of Web-CAT, an extensible, automated online program testing and grading system. Dr. Edwards is responsible for Web-CAT's adoption at other institutions within the academic community. He is also a member of his department's undergraduate program committee, and chair of the subcommittee on curriculum and courses. He is designer and course coordinator for CS 1705: Introduction to Object-oriented Development I (a CS1-level course using Java, objectdraw, and student testing activities). Role: Dr. Edwards maintains the servers for the AlgoViz Wiki system, and manages its technical aspects. He will be a consultant for portal development.

Graduate, undergraduate, and intern students This effort has historically had great success in recruiting undergraduate and graduate students to contribute. We regularly sponsor undergraduate research projects to develop AVs. The initial Wiki was developed by Matt Cooper for his MS thesis. We had two interns from India working on AV development during Summer 2007 and another coming in Summer 2008. Eight independent study projects were completed over the past two years on AV-related topics. Graduate Students A.J. Alon (minority) and Monika Akbar (female) are currently working on the AlgoViz project, and they will be joined in Fall by Michael Stewart.

10 Results from Prior NSF Support

NSF CCLI Phase 1 Award (DUE-0836940): *Building a Community and Establishing Best Practices in Algorithm Visualization through the AlgoViz Wiki.* PIs: Cliff Shaffer and Stephen Edwards. $149,206. for 01/2009 – 12/2010. This project is discussed in Section 4.3.

References

[1] Association for Computing Machinery. The ACM digital library. `http://portal.acm.org`, 2009.

[2] J. Bergin, K. Brodie, M. Patio-Martnez, M. McNally, T.L. Naps, S.H. Rodger, J. Wilson, M. Goldweber, S. Khuri, and R. Jimnez-Peris. An overview of visualization: its use and design: report of the working group in visualization. In *ITiCSE '96: Proceedings of the 1st Conference on Integrating Technology into Computer Science Education*, pages 192–200, 1996.

[3] M.H. Brown and R. Sedgewick. A system for algorithm animation. In *SIGGRAPH '84: Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, pages 177–186, 1984.

[4] M.D. Byrne, R.Catrambone, and J.T. Stasko. Do algorithm animations aid learning? Technical Report GIT-GVU-96-18, Georgia Institute of Technology, 1996.

[5] CITIDEL: Computing and information technology interactive digital educational library. `http://www.citdel.org`, 2008.

[6] Connexions scholarly content repository. `http://cnx.org`, 2008.

[7] M.L. Cooper. Algorithm visualization: The state of the field. Master's thesis, Virginia Tech, April 2007.

[8] Ensemble NSDL Pathway Project. Ensemble : A National Digital Library for Computing Education. `http://www.computingportal.org/`, 2009.

[9] S. Grissom, M.F. McNally, and T.L. Naps. Algorithm visualization in CS education: comparing levels of student engagement. In *SoftVis '03: Proceedings of the 2003 ACM Symposium on Software Visualization*, pages 87–94, 2003.

[10] J.S. Gurka and W. Citrin. Testing effectiveness of algorithm animation. In *Proceedings of the 1996 IEEE Symposium on Visual Languages VL '96*, pages 182–189, September 1996.

[11] S.R. Hansen, N.H. Narayanan, and D. Schrimpsher. Helping learners visualize and comprehend algorithms. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 13(3):291–317, 2000.

[12] Hope College. Complete collection of algorithm visualizations. `http://www.cs.hope.edu/~dershem/ccaa/ccaa`, 2006.

[13] C. Hundhausen and S.A. Douglas. Using visualizations to learn algorithms: should students construct their own, or view an expert's? In *Proceedings of the 2000 IEEE International Symposium on Visual Languages VL '00*, pages 21–28, September 2000.

[14] C. Hundhausen, S.A. Douglas, and J.T. Stasko. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing*, 13(3):259–290, 2002.

[15] Institute of Electrical and Electronics Engineers. Ieee xplore. `http://ieeexplore.ieee.org`, 2009.

[16] D.J. Jarc, M.B. Feldman, and R.S. Heller. Assessing the benefits of interactive prediction using web-based algorithm animation courseware. In *SIGCSE '00: Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education*, pages 377–381, March 2000.

[17] JERIC: Journal on Educational Resources in Computing. `http://www.acm.org/pubs/jeric`, 2008.

[18] A.W. Lawrence, J. Stasko, and A. Badre. Empirically evaluating the use of animations to teach algorithms. In *Proceedings of the 1994 IEEE International Symposium on Visual Languages VL '94*, pages 48–54, October 1994.

[19] T.L. Naps, S. Cooper, B. Koldehofe, C. Leska, G. Rößling, W. Dann, A. Korhonen, L. Malmi, J. Rantakokko, R.J. Ross, J. Anderson, R. Fleischer, M. Kuittinen, and M. McNally. Evaluating the educational impact of visualization. *SIGCSE Bulletin*, 35(4):124–136, 2003.

[20] T.L. Naps, J.R. Eagan, and L.L. Norton. JHAVÉ: An environment to actively engage students in web-based algorithm visualizations. In *SIGCSE '00: Proceedings of the thirty-first SIGCSE technical symposium on Computer science education*, pages 109–113, 2000.

[21] T.L. Naps, G. Rössling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S.H. Rodger, and J. Ángel Velázquez-Iturbide. Exploring the role of visualization and engagement in computer science education. In *ITiCSE-WGR '02: Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education*, pages 131–152, 2002.

[22] T.L. Naps and B. Swander. An object-oriented approach to algorithm visualization: easy, extensible, and dynamic. In *SIGCSE '94: Proceedings of the twenty-fifth SIGCSE symposium on Computer science education*, pages 46–50, 1994.

[23] Open Archives Insitute Protocol for Metadata Harvesting. `http://www.openarchives.org/OAI/openarchivesprotocol.html`, 2008.

[24] S. Ponce. AlgoViz graph creator. `http://research.cs.vt.edu/algoviz/gds_algoviz/AlgovizGraphCreator.html`, 2008.

[25] G. Rößling. Animation repository. `http://www.animal.ahrgr.de/animations.php`, 2008.

[26] G. Rößling, T.L. Naps, M.S. Hall, V. Karavirta, A. Kerren, C. Leska, A. Moreno, R. Oechsle, S.H. Rodger, J. Urquiza-Fuentes, and J. Ángel Velázquez-Iturbide. Merging interactive visualizations with hypertextbooks and course management. *SIGCSE Bulletin*, 38(4):166–181, 2006.

[27] P. Saraiya. Effective features of algorithm visualizations. Master's thesis, Virginia Tech, July 2002.

[28] P. Saraiya, C.A. Shaffer, D.S. McCrickard, and C. North. Effective features of algorithm visualizations. In *SIGCSE '04: Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, pages 382–386, March 2004.

[29] C.A. Shaffer, M.L. Cooper, and S.H. Edwards. Algorithm visualization: a report on the state of the field. In *SIGCSE '07: Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, pages 150–154, March 2007.

[30] SIGCSE Educational Links. `http://sigcse.org/topics`, 2008.

[31] SMETE Digital Library. `http://www.smete.org`, 2008.

[32] John Stasko, Colleen Kehoe, and Ashley Taylor. Rethinking the evaluation of algorithm animations as learning aids: an observational study. *International Journal of Human Computer Studies*, 54(2):265–284, 2001.

[33] Virginia Tech Data Structures and Algorithm Visualization Research Group. Algoviz wiki annotated bibliography of the AV research literature. `http://algoviz.cs.vt.edu/AlgovizWiki/AnnotatedBibliography`, 2009.

[34] Virginia Tech Data Structures and Algorithm Visualization Research Group. Data Structures and Algorithm Visualization Wiki. `http://algoviz.cs.vt.edu`, 2009.

[35] T. Waldmann and J. Hermann. MoinMoin Wiki Engine. `http://moinmo.in`, 2008.