SALSA and ALVIS: A Language and System for Constructing and Presenting Low Fidelity Algorithm Visualizations

Christopher Hundhausen
Laboratory for Interactive Learning Technologies
Department of Information and Computer Sciences
University of Hawai'i
Honolulu, HI 96822
hundhaus@hawaii.edu

Sarah Douglas
Human-Computer Interaction Laboratory
Department of Computer and Information Science
University of Oregon
Eugene, OR 97403–1202
douglas@cs.uoregon.edu

Abstract

Computer science educators have traditionally used algorithm visualization (AV) software to create graphical representations of algorithms that are later used as visual aids in lectures, or as the basis for interactive labs. Based on ethnographic field studies we have conducted in an undergraduate algorithms course, we advocate an alternative teaching approach in which students use simple art supplies to construct and present their own visualizations to their peers and instructor for feedback and discussion. To support this approach, we have built SALSA and ALVIS, a prototype language and system that enable students to (a) quickly construct rough, unpolished ("low fidelity") visualizations in much the same way they would do so with simple art supplies, and (b) interactively present those visualizations to an audience. Our prototype pioneers a novel technique for programming visualizations based on spatial relations, and a novel presentation interface that supports reverse execution and dynamic mark-up and modification.

1. Introduction

Computer science instructors have traditionally used algorithm visualization (AV) software to construct algorithm visualizations (AVs) that are later used either as visual aids in lectures, or as the basis for interactive labs, e.g., [1]. Building on [2], and inspired by social constructivist learning theory [3], we have explored a teaching approach in which students use simple art supplies to construct their own AVs, and then present those AVs to their instructor and peers for feedback and discussion [4,5]. In this paper, we present SALSA and ALVIS, a computer-based language and system that support the creation and presentation of rough, unpolished ("low fidelity") AVs much like those built out of simple art supplies. Because it is derived from empirical studies of how students construct and present AVs out of simple art supplies [4,6], our prototype is better tailored to AV creation and presentation than commercially-available animation packages.

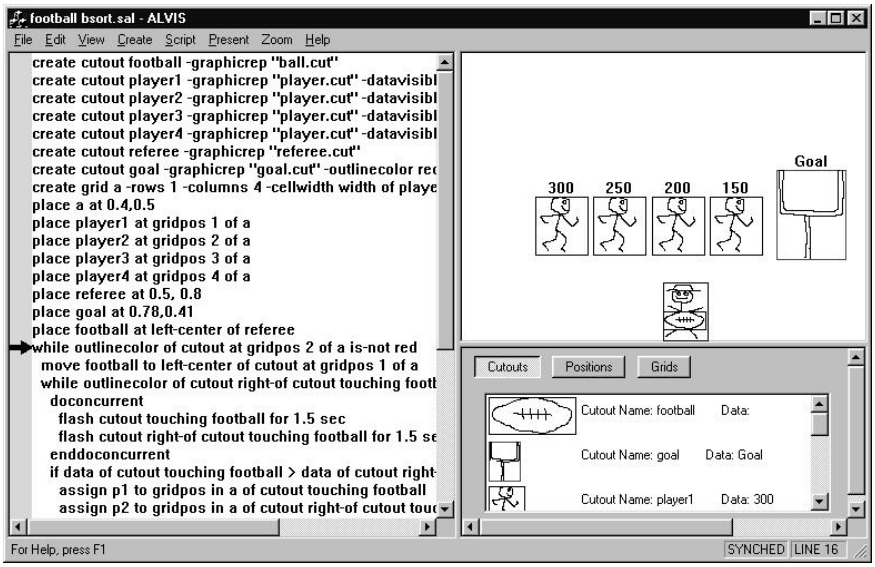
2. Design requirements

Based on ethnographic studies we conducted in a junior-level algorithms course [4], and on detailed studies of how students construct AVs out of art supplies [6], we derived the following design requirements for our system:

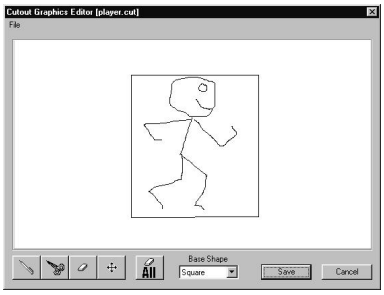
R1:	*Users must be able to create, systematically lay out, and animate simple objects containing sketched graphics.*
R2:	*Users must be able to construct storyboard objects by cutting and sketching; they must be able to position objects by direct placement.*
R3:	*Users must be able to create storyboards using spatial relations, not Cartesian coordinates.*
R4:	*The system must support an execution model based on spatial, rather than algorithmic logic.*
R5:	*The system must enable users to present their storyboards interactively. This entails an ability to execute storyboards in both directions; to rewind and fast forward storyboards to points of interest; and to point to, mark-up, and modify storyboards as they are being presented.*

3. Prototype Language and System

To explore the design space circumscribed by the above requirements, we have developed a prototype language and system for creating and presenting "low fidelity" AVs. The foundation of our prototype is SALSA (Spatial Algorithmic Language for StoryboArding), a high-level, interpreted language for programming low fidelity AV "storyboards." Whereas conventional AV



(a) Snapshot of a session with ALVIS



(b) Cutout graphics editor

(c) Execution control interface

(d) Presentation interface

Figure 1. The ALVIS interactive environment

technology requires one to program an AV by specifying explicit mappings between an underlying "driver" program and the AV, SALSA enables the layout and logic of an AV to be specified in terms of its *spatiality*—that is, in terms of the spatial relations (e.g., *above*, *right-of*, *in*) among objects in the AV.

The second key component of the prototype is ALVIS (ALgorithm VIsualization Storyboarder), an interactive environment for programming in SALSA. Figure 1a presents a snapshot of ALVIS, which has three main regions:

1. *Script View* (left). This view displays the SALSA script presently being explored; the arrow on the left-hand side denotes the line at which the script is presently halted—the current "insertion point" for editing.
2. *Storyboard View* (upper right). This view displays the storyboard generated by executing the current script up to the insertion point marked by the arrow.
3. *Created Objects Palette* (lower right). This view contains an icon representing each created object.

ALVIS strives to make constructing a SALSA storyboard as easy as constructing a storyboard out of simple art supplies. In ALVIS, users create storyboards by using a graphics editor (Figure 1b) to cut out and sketch visualization objects, which they lay out in the *Storyboard View* by direct manipulation. They then specify, either by direct manipulation or by directly typing in SALSA commands, how the objects are to be animated over time.

Likewise, ALVIS strives to make presenting a SALSA storyboard to an audience as easy and flexible as presenting an "art supply" storyboard. To that end, ALVIS supports three features that are taken for granted in "art supply" presentations, but that are notably absent in conventional AV technology. First, using ALVIS's execution interface (Figure 1c), a presenter may reverse the direction of execution in response to audience questions and comments. Second, ALVIS includes a "mark up pen" (third tool from left in Figure 1d) with which the presenter and audience members may dynamically annotate the storyboard as it is executing. Finally, presenters and audience members may dynamically modify a storyboard as it is executing by simply inserting SALSA commands at the current insertion point in the script.

See [4], available at http://lilt.ics.hawaii.edu/~hundhaus/dis, for further information on SALSA and ALVIS.

4. References

[1] M. H. Brown, *Algorithm Animation*. Cambridge, MA: The MIT Press, 1988.

[2] J. T. Stasko, "Using student-built animations as learning aids," in *Proc. ACM Technical Symposium on Computer Science Education*. New York: ACM Press, 1997, pp. 25-29.

[3] J. Lave and E. Wenger, *Situated Learning: Legitimate Peripheral Participation*. New York: Cambridge U. Press, 1991.

[4] C. D. Hundhausen, "Toward Effective Algorithm Visualization Artifacts: Designing for Participation and Communication in an Undergraduate Algorithms Course." Ph.D. Dissertation (Tech. Rep. No. CIS-99-07), Department of Computer and Information. Science, University of Oregon, Eugene, 1999.

[5]. C.D. Hundhausen and S.A. Douglas, "Using Visualizations to Learn Algorithms: Should Students Construct Their Own, or View an Expert's?," in *Proc. 2000 IEEE Symposium on Visual Languages.* Los Alamitos, CA: IEEE Comp. Soc. Press, 2000.

[6] S. A. Douglas, C. D. Hundhausen, and D. McKeown, "Toward empirically-based software visualization languages," in *Proc. 11th IEEE Symposium on Visual Languages*. Los Alamitos, CA: IEEE Computer Society Press, 1995, pp. 342-349.