Algorithm Visualization
Reflections and Future Directions

John Stasko

Information Interfaces Research Group
School of Interactive Computing & GVU Center
Georgia Tech

stasko@cc.gatech.edu
http://www.cc.gatech.edu/~stasko

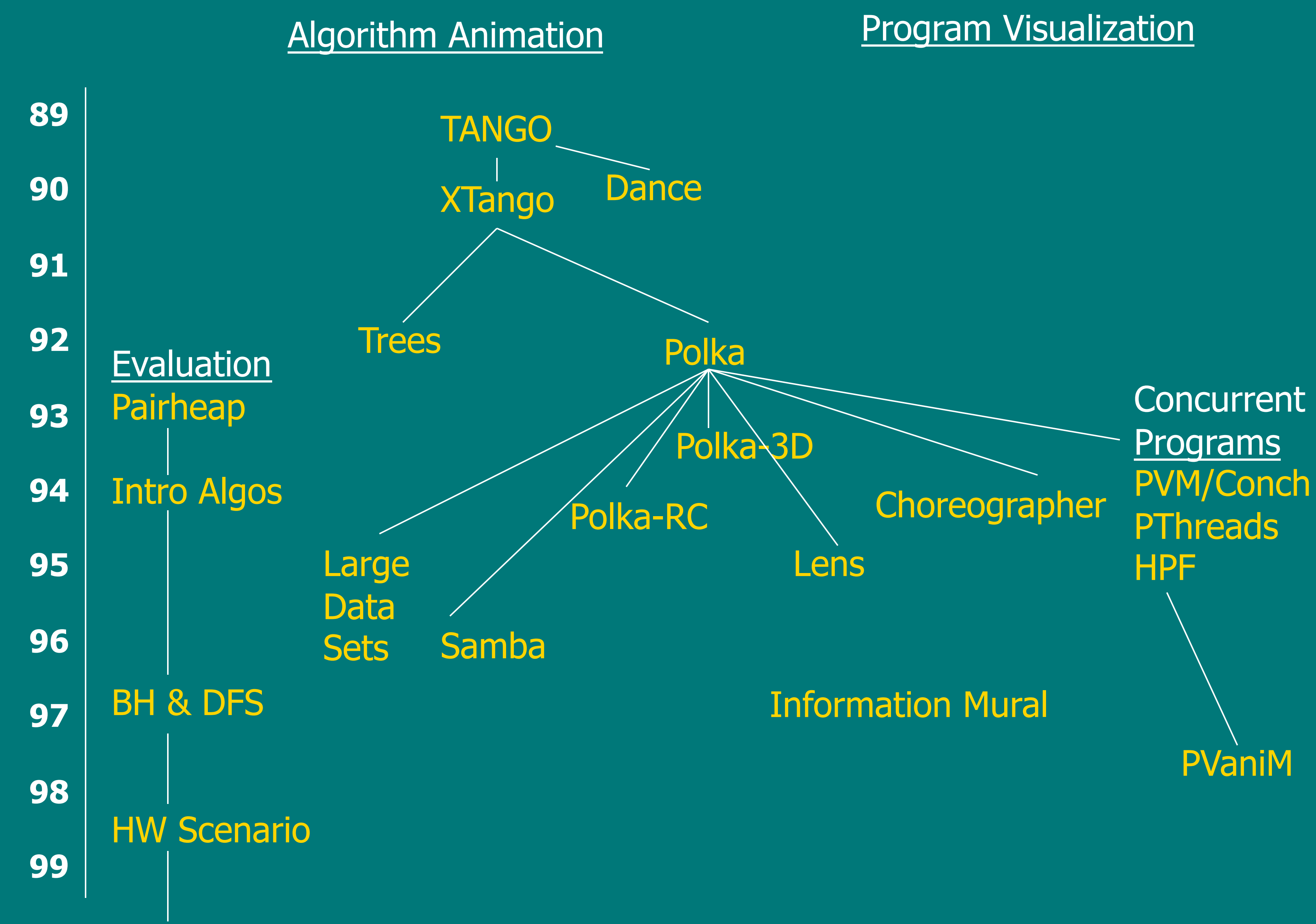
GRAPHICS VISUALIZATION & USABILITY CENTER

information interfaces

Georgia Tech College of Computing

Introduction

- BS Math 1983, Bucknell
- PhD CS 1989, Brown
  - TANGO algorithm animation system
- On faculty at Georgia Tech since ‘89


Algorithm Animation
Program Visualization
89
90
91
92
93
94
95
96
97
98
99
TANGO
XTango
Dance
Trees
Polka
Evaluation
Pairheap
Intro Algos
BH & DFS
HW Scenario
Polka-3D
Polka-RC
Lens
Choreographer
Large Data Sets
Samba
Information Mural
Concurrent Programs
PVM/Conch
PThreads
HPF
PVaniM

Algorithm Animation's History

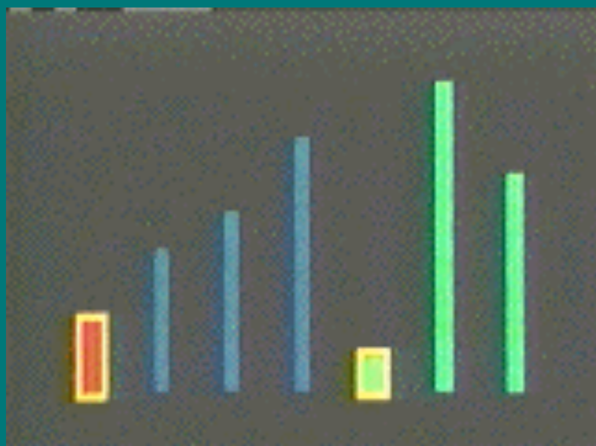
- Quick look back for those who may not be so familiar

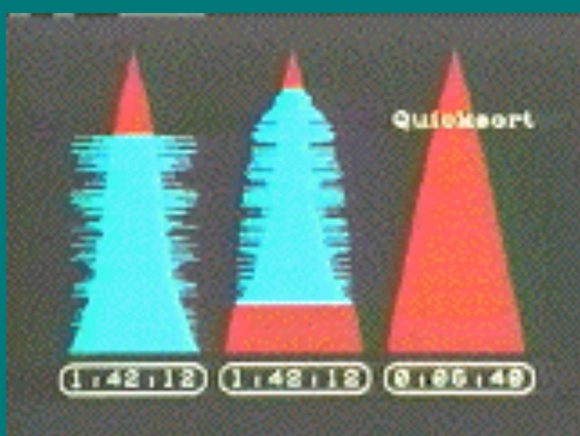
Sorting Out Sorting

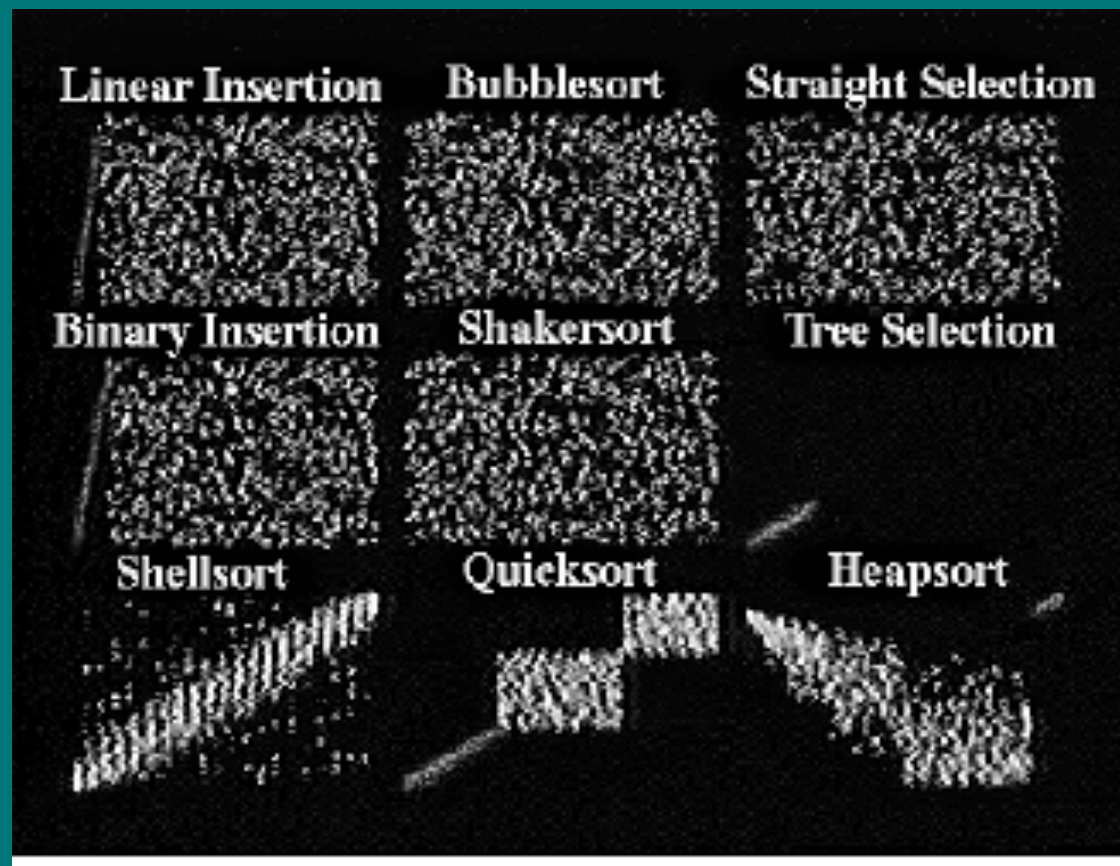
- Seminal work in area
- 30 minute video produced by Ron Baecker at Toronto in 1981
- Illustrates and compares nine sorting algorithms as they run on different data sets

Sorting Out Sorting

Video





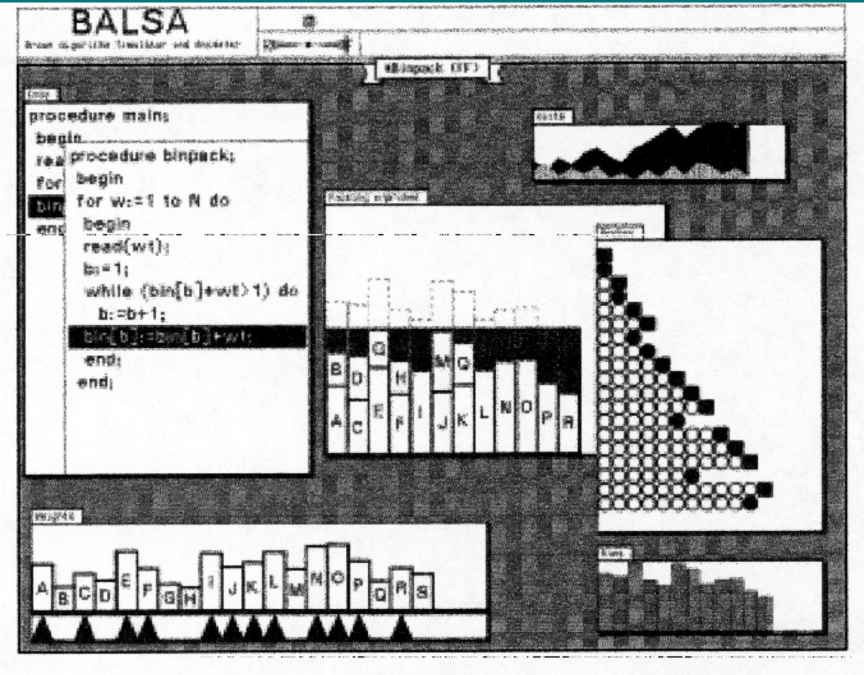


Early Algo Anim Systems

- Smalltalk-based systems, Animus
  - Ralph London & Rob Duisberg, Univ. of Washington
- BALSA
  - Marc Brown, Brown Univ.

Balsa

- Used in “electronic classroom” at Brown
- Introduced use of multiple views and interesting event model



M. Brown
Computer ‘88

Brown’s Electronic Classroom

Circa 1983

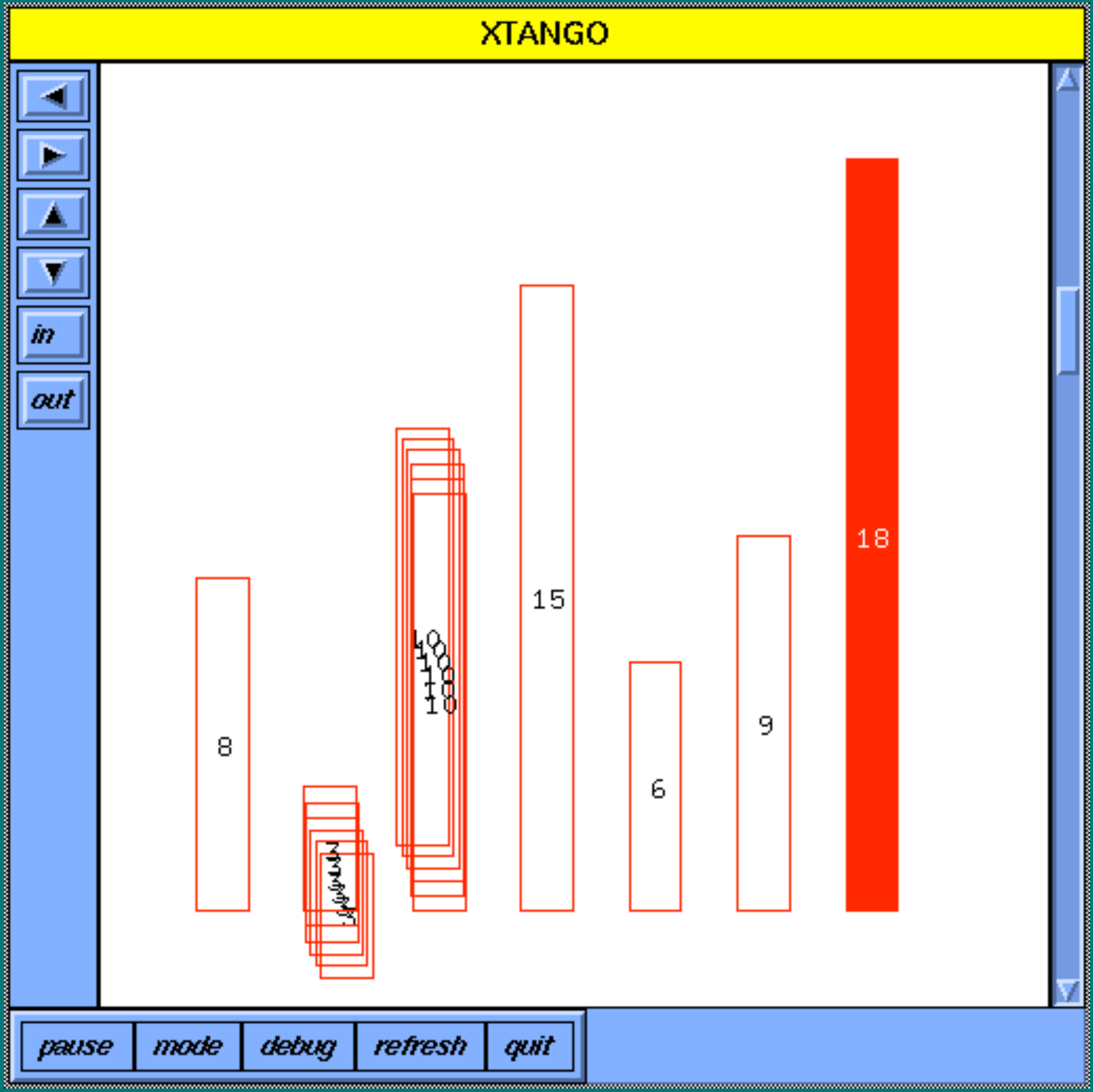
Apollo workstations

Tango

- Added smooth animation to algorithm animation
- Simplification of the design/programming process (easier programming model and framework)
- Formal model of the animation, the Path-Transition Paradigm

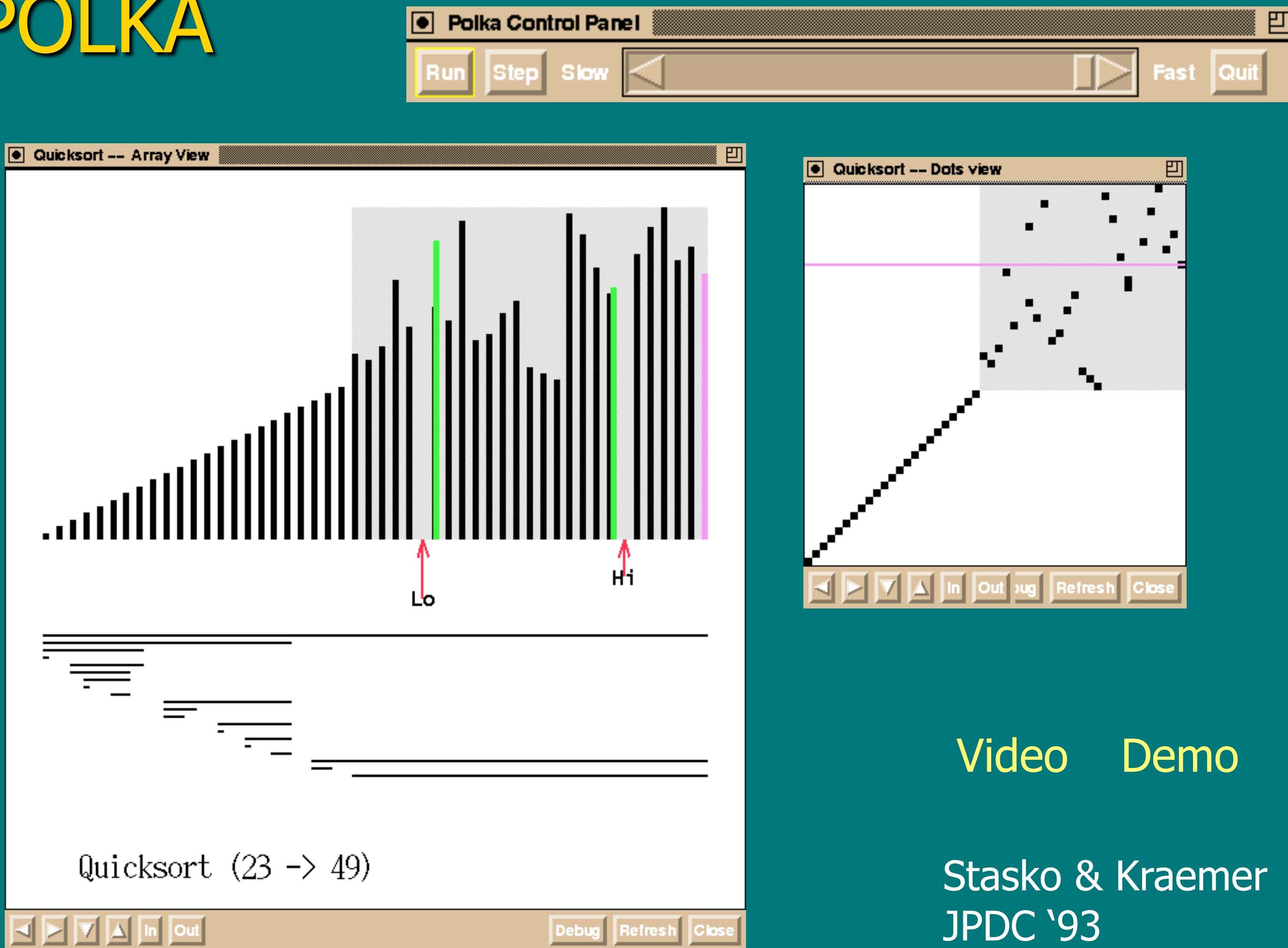
Tango

Multiple frames from bubblesort



Stasko
IEEE Computer '90
JVLC '90

POLKA



Video Demo

Stasko & Kraemer
JPDC '93

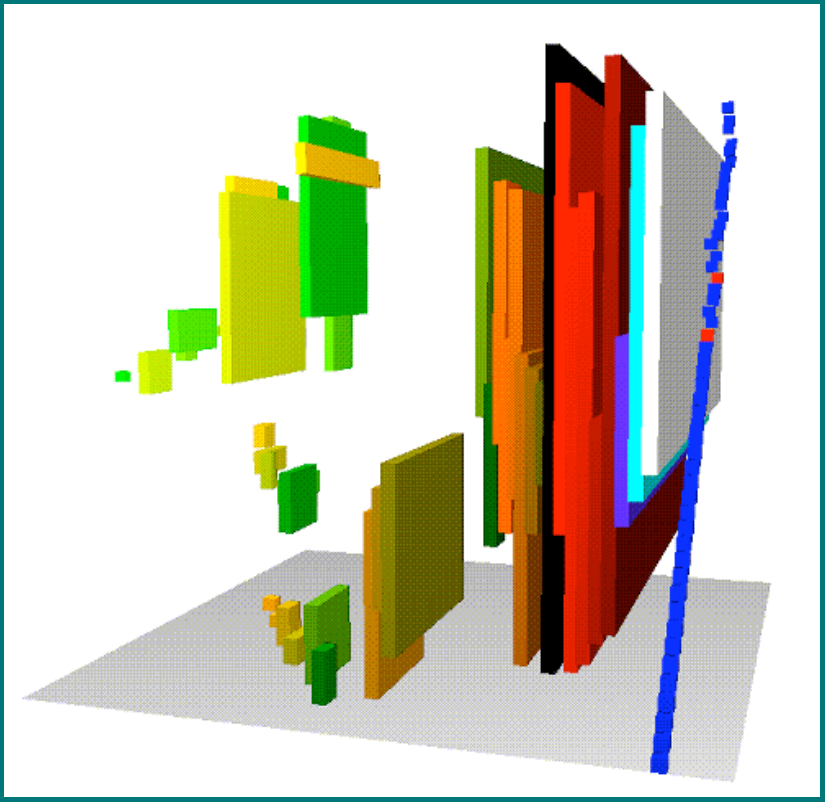
Polka-3D

- 3-D & VR version, same animation model

Quicksort side view

Blue dots are as in
2-d view

Colored planes
represent exhanges



Stasko & Wehrli
VL '93

Student-Built Animations

Stasko
SIGCSE '97

- Samba
  - Simple animation scripting language

```
circle 1 0.8 0.8 0.1 red half
line 2 0.1 0.1 0.2 0.2 green thin
rectangle 3 0.1 0.9 0.1 0.1 blue solid
text 4 0.0 0.0 0 black Hello
circle 6 0.3 0.3 0.2 wheat solid
triangle 7 0.5 1.0 0.6 0.8 0.4 0.9 cyan solid
bigtext 8 0.2 0.2 0 black Some Big Text
moveto 1 6
moverelative 3 0.05 -0.4
jumprelative 4 0.4 0.4
lower 1
color 6 blue
```

- Embed print statements in any program to generate

Transition

- In the mid ’90’s, my work began to transition from building systems to thinking about the pedagogical implications of the technology

Evaluation

- Big question:
  - Do algorithm animations help students learn and understand better?

- I think that this issue has dominated algorithm animation research of the last ~8 years

How to Evaluate?

- What's a good way to evaluate the pedagogical value of an algorithm animation?

Ponder

My Work

- Conducted a number of empirical studies to answer question
  - Compare learning with animation to learning without
  - An evolving methodology
- Concluded with a couple reflective articles

1. Pairing Heaps

- Classical experimental design
  - Exam scenario
  - Anim vs. no anim, followed by test
- Results: Just having animation doesn't make learning happen
- Reflection: Difficult for student to leverage animation of complex algorithm when they don't understand algorithm and visual mapping yet
- Reflection: The better students benefit more…

Stasko, Badre & Lewis
InterCHI '93

2. Introductory Algorithms
Sorting, graphs

- Lab style methodology
- Results: Benefits to animation, though uneven
- Reflection: Interaction is the key
  – Students who enter their own data sets into the algorithm benefited from animation

Lawrence, Badre, & Stasko
VL '94

3. Binomial Heap & DFS

- Classical experimental design with prediction added
  - Is animation like prediction?
  - Does it help the student to anticipate what will happen next and learn from that?
- Results: Mixed findings, some support
- Reflection: We need a new methodology

Byrne, Catrambone & Stasko
Computers & Ed '99

4. Binomial Heaps

- Change to HW scenario
  - Provide student with learning objectives, materials, and questions up front
  - Give unlimited work time
- Results: Significant benefit to animation
  - Students worked longer
- Reflection: Animations appear to help motivation
  - Make a complex algorithm less intimidating
  - Students with animation were “happier”

Kehoe, Stasko & Taylor
IJHCS ‘00

5. Samba

- Students built their own animations during algorithms class
  - Performance on final exam on questions about those algorithms was excellent
  - Anecdotal evidence that this helped understanding
  - Students spent a lot of time tweaking graphics

Stasko
SIGCSE ‘97

Unreported Work

- In-class assessment of quicksort
- Worked with focus groups of students
- Developed task analysis methodology for ensuring that animations were illustrating the "right" things

Little Victories...

From *******@felix.cc.gatech.edu Wed Feb 25 19:59:22 1998
Date: Wed, 25 Feb 1998 19:52:17 -0500
To: stasko@cc.gatech.edu
From: Susan ****** <*******@cc.gatech.edu>
Subject: cs3158

I know you always wanted feedback on using algorithm animations...
Well, I have another instance where I think it helped me...

I was taking a sample Computer Science GRE,
a question was asked that had five sorting algorithms as the
possible answers.

I was pondering the question and thinking about the algorithms
- some that we've been exposed to for years are easy to remember,
almost inherent. But then I came to "shellsort," and I stopped for
a second, going "aw geez... that was a complicated one..." but
then suddenly I could see that picture of the animation in my head
and could remember it as clear as a bell.

Thought you'd enjoy it....
Susan ******

Let’s Step Back a Moment

- Some important questions:
  - What does it mean to understand an algorithm?
  - What makes a good algorithm animation?
  - How should algorithm animations be used?
  - ...

Understanding

- What does it mean for a student to understand an algorithm?
- What are we evaluating in these studies?

Ponder

Understanding an Algorithm is

- Knowing which tasks, or types of tasks, for which the algorithm is suited or appropriate
- Knowing how to “run” the algorithm, knowing its procedures, operations, steps and chronology
- Knowing the algorithm’s purpose and what it can accomplish
- Analyzing the algorithm’s complexity and efficiency
- Having the ability to code or program the algorithm
- Being able to articulate about the algorithm in one’s own high-level (natural language) words
- Having the ability to transfer knowledge about it to another algorithm or domain
- Knowing where the algorithm fits into the bigger taxonomy/ methodology of algorithms in general

Typical Student Approach

- Focus group findings
- Two main learning approaches:
  - Conceptualize algorithm at a high level (English discourse). Students referred to this as being able to describe it in their own terms (top down)
  - Start with pseudo code and get good grasp of how it works mechanically, including tracing on data sets (bottom up)

Newstetter

Learning Order

- A. Knowing which tasks, or types of tasks, for which the algorithm is suited or appropriate
- B. Knowing how to “run” the algorithm, knowing its procedures, operations, steps and chronology
- C. Knowing the algorithm’s purpose and what it can accomplish
- D. Analyzing the algorithm’s complexity and efficiency
- E. Having the ability to code or program the algorithm
- F. Being able to articulate about the algorithm in one’s own high-level (natural language) words
- G. Having the ability to transfer knowledge about it to another algorithm or domain
- H. Knowing where the algorithm fits into the bigger taxonomy/methodology of algorithms in general

- A & C

  or

  B & F
- Other pair above
- E
- D
- H
- G

Ponder

Algorithm Animation's Value?

- For which of the learning objectives can algorithm animations help?
- Perhaps better: For which will algorithm animation have greater learning potential?

Assess AA Utility

- A. Knowing which tasks, or types of tasks, for which the algorithm is suited or appropriate
- B. Knowing how to "run" the algorithm, knowing its procedures, operations, steps and chronology
- C. Knowing the algorithm's purpose and what it can accomplish
- D. Analyzing the algorithm's complexity and efficiency
- E. Having the ability to code or program the algorithm
- F. Being able to articulate about the algorithm in one's own high-level (natural language) words
- G. Having the ability to transfer knowledge about it to another algorithm or domain
- H. Knowing where the algorithm fits into the bigger taxonomy/ methodology of algorithms in general

Ponder

What Makes a Good Algorithm Animation?

- What are the elements of an effective visualization/animation?
- How does one design a visualization that students can learn from?

Ponder

Good AA Design

- Very little work done on this topic
- Design has largely been a very individual, intuitive matter
- Involves graphic design, pedagogy, understanding of misunderstandings, story telling, artistic sense, …

Multiple Views

- Can be quite valuable but also problematic
  - We can only look one place at a time (people may not notice the text at the bottom of one important view)

Heapsort Case Study

- Taught psychology faculty colleague the heapsort algorithm using an existing animation
- Q & A session uncovered many problems with the animation
  - 20+ changes made

Heapsort Changes

Video Demo

- Show numbers appear in array view first (reading them in). Then place them in the tree view and (somehow) highlight the correspondences.
- Put better text region somewhere for telling viewer what is happening.
- Use circle with small lines out sides (indicating subtree) to tell where a heap is being checked.
- Change color of node "dropping down" in heapify operation. Rather than putting the "H" there.
- Put some text like "Comparing key values" and "Swapping with biggest" in tree view when that occurs. Similarly, put some text in array view to draw viewers attention to swaps there.
- Put up text to tell when valid heap has been built.
- Put up text to indicate when root and last leaf are being swapped.
- Change color of node when it has been moved out of tree and is in place (do in both views).
- Take circle away at this later stage of algorithm.
- Change color of node moving down in heapify.
- Pause longer after heapify finishes.
- Put text on both sides of line in array view, indicating regions that are sorted and unsorted respectively.
- Take out that "Elevate #" text at top.
- Move line over in array view to very left end.

The Learner is Your User

- Understanding an algorithm well handicaps you in designing an animation of it
- We should learn from HCI
  – Have learners as part of your design team
  – Iteratively prototype animation
  – Test it with other learners

Algorithm Animation's Place

- How should we use algorithm animations for learner benefit?
  - What's the scenario of use?
  - When?
  - Where?
  - How?

Ponder

Potential Use Scenarios

- In class
  - When and how?
- Lab
- Offline

Typical Class

- A discussion of the motivation for an algorithm and how it might be used in the real world. Frequently, analogy is used to give the students a feeling for the practical value of the algorithm
- A discussion of the high level methodology and general operations of an algorithm. Here, a very informal description is provided, attempting to give the students a "feel" for the algorithm
- A discussion of the detailed methodology and operations of an algorithm, with a focus on tracing the algorithm in pseudo-code, understanding its control flow and memory usage
- A discussion of the algorithm's computational complexity, that is, an analysis of its space and time requirements
- A discussion revolving around the presentation of an algorithm animation illustrating the algorithm's methods and operations

Representation

- Note that each lecture activity involves a different kind of representation
- Each uses a set of artifacts and discussion terms from a different domain/language

Components

- Motivation/Example
- High-level methodology
- Detailed methodology
- Complexity analysis
- Algorithm animation

- Real-world things, artifacts, problems
- English discourse
- Pseudo code or programming lang.
- Math notation
- Graphics and visualization

Reflections

- Running the sessions amazingly instructive to me
  - Often saw just how an animation wasn't helping a student
- Two papers reflect on the evaluation findings of many researchers
  - "Meta-Eval of AV," Hundhausen, Douglas, & Stasko, *JVLC*, June '02
  - "Algo Vis," Stasko & Hundhausen, appears in *CS Education Research*, '04

Results

- Findings definitely mixed, but some studies do show benefits
  - ---> Potential for pedagogical aid is there, but just can't throw animation at algorithm and expect it to help
  - Interaction & engagement is key
    - Blindly watching algorithm animation not really helpful
    - Student must interact with animation and be engaged
  - Animations help motivation, can make algorithm less intimidating

Future Work

- What's still missing in algorithm visualization research?
  - Further evaluation (somewhat)
  - Simpler animation construction (somewhat)
  - Design principles
  - Giant web index
  - Integrated curriculum
  - Interactive classroom blackboard tools

Controversial Thoughts

- Age breeds cynicism
- Let me push a few buttons

Teach, Don't Rationalize

- The people who seem to benefit most from algorithm animations are those who already understand the algorithm
- **Maybe** algo anim helps instructors present better (or think they present better) and that's why they like it
  - But do students learn better?

Ponder

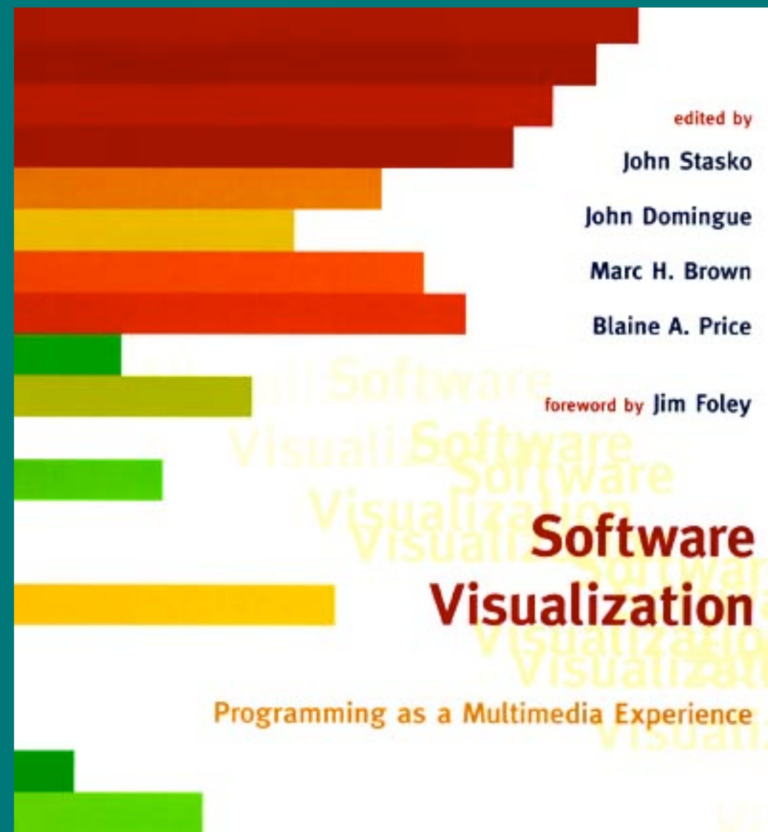
No Good Use Scenario

- It takes too much time to use an algorithm visualization well in class
- Difficult to get students to use them offline

Ponder

Software Visualization

- Good introduction to field
- Many AA chapters
  - History
  - Design
  - SoS
  - Evaluation



Acknowledgments

- Research supported by National Science Foundation and the GVU Center

- More info
  - http://www.cc.gatech.edu/gvu/ii
  - http://www.cc.gatech.edu/gvu/softviz
  - stasko@cc.gatech.edu

Thanks for Your Attention!

- Questions?