A COMPUTER ANIMATED EXPLANATION OF INTERNAL SORTING METHODS

R. Baecker* and D. Sherman**

Dynamic Graphics Project, University of Toronto

** Also with Human Computing Resources Corporation, Toronto*
*** Currently with Blake, Cassels & Graydon, Toronto*

ABSTRACT

Program animation is the use of computer animation to visualize the behavior of algorithms and computer programs. In our presentation we shall discuss and show an experimental computer-animated film production about a variety of internal sorting methods. The algorithms and their relative efficiency are illustrated and compared visually in a number of ways.

RÉSUMÉ

L'animation de programmes consiste en l'utilisation de l'animation par ordinateur pour visualiser le comportement des algorithmes et des programmes. Dans notre présentation, nous montrerons et discuterons un film animé par ordinateur décrivant plusieurs méthodes de tri interne. Les algorithmes et leur efficacité relative sont illustrés et comparés visuellement de façons variées.

Program visualization is the use of graphics to facilitate the understanding of programs by people. Graphics is to be interpreted in this context in a very broad sense, including the arts of animation, moviemaking, graphic design, and typography, and including the technology of interactive computer graphics. Program visualization is relevant to the production and maintenance of reliable software, and to the teaching of numerous aspects of computer science.

Program animation is the use of computer animation to visualize the behavior of algorithms and computer programs. The animation is typically generated semi-automatically from the computer program as it executes.

We shall describe and show an experimental computer-animated film production about a variety of internal sorting methods. *Sorting Out Sorting* is a 30-minute colour sound film which explains through dynamic graphics nine different internal sorts: three Insertion Sorts -- Linear Insertion, Binary Insertion, and the Shellsort; three Exchange Sorts -- the Bubblesort, the Shakersort, and the Quicksort; and three Selection Sorts -- Straight Selection, Tree Selection, and the Heapsort.

The algorithms and their relative efficiency are illustrated and compared visually in a number of ways: The title introducing each algorithm is first animated quickly using the technique. The method is then presented in detail through its operation on a simple example. Each group of three sorts is next compared with dynamic graphs illustrating their performance. They are then further compared in a race on a randomly chosen sequence of 250 items. The film concludes with a grand race of all nine algorithms running in parallel on a randomly chosen sequence of 2500 items.

Bibliography

Baecker, Ronald M., "Towards Animating Computer Programs: A First Progress Report", *Proceedings of the Third National Research Council Man-Computer Communications Seminar*, Ottawa Ontario, May 1973, pp. 4.1-4.10.

Baecker, Ronald M., "Two Systems Which Produce Animated Representations of the Execution of Computer Programs", *ACM SIGCSE Bulletin*, Volume 7, Number 1, February 1975, pp. 158-167.

Baecker, Ronald and Sherman, David, *Sorting out Sorting*, 30 minute colour sound film or video cartridge, Dynamic Graphics Project, Computer Systems Research Group, University of Toronto, 1981.

Hopgood, F.R.A., "Computer Animation Used as a Tool in Teaching Computer Science", *Proceedings of the IFIP Congress*, Applications Volume, 1974. pp. 889-892.

Knowlton, Kenneth C., *L6: Bell Telephone Laboratories Low-Level Linked List Language*, black-and-white sound film, Bell Telephone Laboratories, Murray Hill, N.J., 1966.

Knuth, Donald E., *The Art of Computer Programming*, Vol. 3: *Sorting and Searching*. Addison-Wesley, 1973.

Lorin, Harold, *Sorting and Sort Systems*, Addison-Wesley, 1975.

Wirth, Nicklaus, *Algorithms + Data Structures = Programs*, Prentice-Hall, 1976.

Yarwood, Edward, "Toward Program Illustration", M. Sc. Thesis, Department of Computer Science, University of Toronto, 1974.