8 Summary of Contributions

In this thesis, I have presented and analyzed several important issues pertaining to algorithmic animation. The theoretical part of the thesis deals with cognitive aspects of algorithm visualization, with the design of graphical elements, and the definition of an architecture for algorithmic animation, both for high-quality production and for the animation of sketches. The practical part of the thesis consists in two hand-crafted animation productions, the development of Chalk Animator, a system for algorithmic animation in the E-Chalk system, and the design and development of the Flashdance algorithm animator. As the thesis shows, there is still much work to be done until we can really reach the ultimate goal of creating intelligent systems capable of animating algorithms automatically.

This thesis is a step towards that goal – its main contributions are the following:

☼ *Analysis and comparison of families of algorithmic animation systems*

Most of the comparisons and historical analysis on software visualization found in the literature are somewhat superficial, or outdated. The survey in Chapter 2 covers the historically most relevant systems, is up to date, and shows the differences and similarities of the various approaches to algorithmic animation.

☼ *A new visualization method for heaps*

In Section 3.4, I presented a new graphical visualization for heaps that goes beyond the one introduced by Brown [91, 93]. The logical structure of a heap is visible, as a tree, together with its physical structure, as an array. The visualization suggests immediately how to compute the indices of the parent and children nodes. Some properties of heaps are also immediately visible from the visualization. This visualization leads immediately to a new way of animating the Heapsort algorithm, which is used in the production mentioned next.

☼ *Resorting Out Sorting – Novel animations of sorting algorithms*

In Chapter 3, I discussed several abstract principles related to graphical displays of quantitative information and to animation systems. Starting from a very general perspective, which considers the theories advanced by researchers of human learning, I proceeded to develop animations of sorting algorithms. “Resorting Out

Sorting" (Section 3.6.1) is a collection which covers five sorting algorithms, most of them animated in ways never before used by other systems.

☼ *Data Structures and Algorithms Production*

A complete production of an interactive electronic learning unit was developed for use in the classroom (Section 3.6.2). The unit illustrates how to embed animations in textual descriptions and it has been in use in several German Fachhochschule for almost two years. The unit was developed as part of the "Virtuelle Fachhochschule" project at the Technische Fachhochschule Berlin.

☼ *Animation of sketches by handcrafting*

A new way of animating algorithms is introduced in this thesis: the animation of sketches drawn on the blackboard (Chapter 4). The thesis analyzes the way educators perform "static animations" on the blackboard and shows how to exploit the macro facilities of the E-Chalk system to the same effect. This technique could be also called "animation on the fly".

☼ *Animation of sketches by instrumented algorithms*

A scripting language for algorithmic animations has been proposed in this thesis (Chapter 4). Programs are instrumented and the resulting event list is processed in order to produce animations for the E-Chalk system. I wrote a renderer that produces macro animations in E-Chalk. These animations can be annotated with audio explanations, and can be exported as Applets to the Web. Several examples show the flexibility and quality of my approach (Section 4.12).

☼ *Animation of sketches by interaction*

Chalk Animator, an animation framework for the interaction protocol of E-Chalk, allows an algorithmic animation to receive data from the electronic chalkboard in form of graphical input from the user. The graphical input can then be used as the data for the animation. With this system the chalkboard becomes effectively alive. Suddenly, sketches drawn by the lecturer can be animated on the screen, allowing her to better illustrate algorithms. The learning effect is enhanced by the surprise of the students who see the chalkboard become dynamic.

☼ *Execution of handwritten code*

The E-Chalk handwriting recognition engine developed by E. Tapia was used to recognize a simple programming language (Section 5.3). Tiniest BASIC was developed for this experiment, I wrote an interpreter which was integrated in the E-Chalk system. This is the first time that an electronic whiteboard system incorporates the possibility of executing handwritten code. I also showed how pseudocode could be interpreted and how annotations could be used to produce semi-automatic algorithmic animations.

☼ *Flashdance – A virtual machine for Algorithmic Animation in Flash*

High quality animations for the Web can be produced using the animation virtual machine proposed in this thesis (Chapter 6). By using the animation engine of Macromedia Flash, the algorithmic animator needs only to implement the abstract operations needed for the animation. The Flash animation engine takes care of the rest. Animations produced by the system can be exported as Flash movies but also as collection of slides. Such slides can be also used in E-Chalk. The algorithmic animation virtual machine is reversible and allows hierarchical stepping, that is, the user can advance the animation at different speeds and can skip animation detail.

☼ *Overlays in views*

The animation engine provides reversibility and overlays in views (Section 6.9). Overlays had never been used before in algorithmic animation, but they are the equivalent of hierarchical stepping in the space domain. By switching overlays on and off, information about the algorithms being monitored can be shown, or can be hidden. This allows the user to explore the same algorithm from different perspectives.

☼ *Evaluation of algorithmic animation for education*

The animations produced for this thesis are not just an academic exercise – they have been used in the classroom at four different institutions: at the Fachhochschule Kiel, the Technische Fachhochschule Berlin (Section 3.6.2), the Freie Universität Berlin, and at Princeton University (Section 6.9). The use of the animations by the students in Kiel and Berlin was observed and was evaluated using programming homeworks (Section 7.2). The student feedback was very favorable to the use of animation in the classroom.

It is my hope that the work described in this thesis and the contributions just listed above can have impact in future development of algorithmic animation systems.