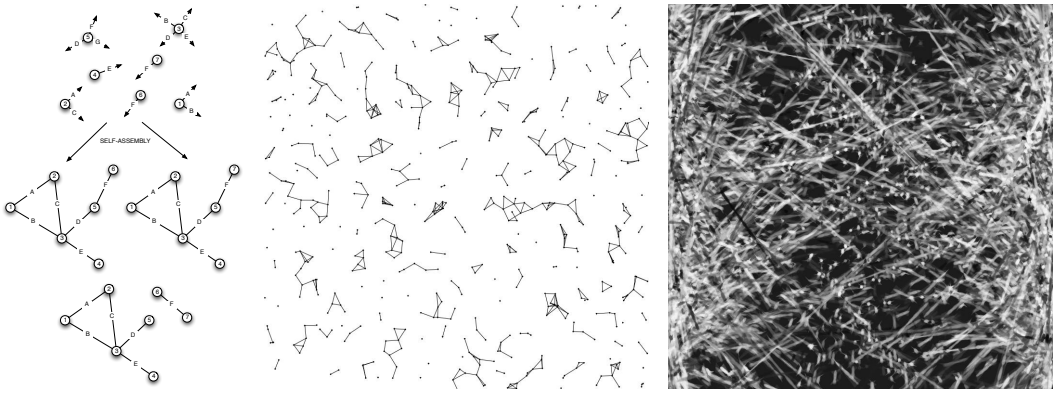
Self-Assembled Art

Russell Deaton*
Computer Science and Computer Engineering
University of Arkansas



(a) Self-Assembly (b) Simulation of Self-Assembly (c) Visualization of Self-Assembly

Keywords: algorithmic art, simulation, visualization

1 Introduction

Code (computer software and the technologies that it enables) is changing fundamentally how human beings interact with each other, and think about themselves and the world. It is a medium through which artists are increasingly expressing themselves. Code can serve as the tool which the artist uses to produce their work, or more interestingly, the artist takes the role of programmer and designs and implements an algorithm that generates the work of art. Thus, the artist's ideas are filtered and constrained through the filter of code, whose limitations and capabilities shape and inform the consequent artistic vision. For example, Casey Reas through his {Software} Structures takes verbal descriptions of processes to produce visual components and turns them into programs[Reas 2009]. In what follows, a Turing-universal model of some natural and man-made phenomena, Self-Assembly, is adapted to the automatic creation of visual art.

2 Self-Assembly

Self-assembly is a process of construction through specific, localized interactions among component parts that result in organized structures[John and Bar 2005]. Self-assembly is capable of producing complex structure and function (*e.g.* ribosomes), and seems to be ubiquitous in the world around us. Self-assembly frequently is mentioned as a potential method for nanoscale manufacturing, and is capable of universal computation[Winfree et al. 1998]. Self-assembly can be modeled as a collection of agents that interact with each other through very specific, localized interactions. Other systems, from physical to theoretical, share this fundamental character. In these systems, there is some potential for "programming" the interactions to achieve a structure with some usefulness, application, or aesthetic appeal, be it through learning, evolution, or design (Figure (a)). Agents are labeled with $P = \{1, 2, 3, 4, 5, 6, 7\}$, and interactions between agents with $K = \{a, b, c, d, e, f\}$. Compatible interactions, *i.e.* the labels match, form edges between particles producing a graph.

3 Self-Assembled Art

To investigate the ability of Self-assembly to produce visual works, a simulation was programmed in the Processing system. A simulation of the system is show in Figure (b). Agents move about a two-dimensional field through random Brownian motion. When they encounter other agents, and if they have compatible interactions, the agents "bind" together. To produce the visualization, different types of interactions were programmed to render different types of visual elements, for instance, trace the line between agents, draw an arc from the center of the line between agents to the 90° axis, or draw a circle at the center of the line connecting agents. This type of visualization produced the image in Figure (c).

4 Conclusion

Self-assembly is a computational model for a variety of man-made and natural systems. By choosing agents and interactions, the artist can program a framework for a particular work while the ultimate product remains nondeterministic. Thus, self-assembly is a constructor for not only man-made and natural structure, but also potentially for "art."

References

JOHN, K., AND BAR, M. 2005. Alternative mechanisms of structuring biomembranes: Self-assembly versus self-organization. *Phys. Rev. Lett. 95*, 198101.

REAS, C., 2009. Software{Structures}. `http://www.whitney.org/arport/commissions/softwarestructures/`.

WINFREE, E., LIU, F., WENZLER, L. A., AND SEEMAN, N. C. 1998. Design and self-assembly of two-dimensional DNA crystals. *Nature 394*, 539–544.

*e-mail: rdeaton@uark.edu

SIGGRAPH 2010, Los Angeles, California, July 25 – 29, 2010.
ISBN 978-1-4503-0210-4/10/0007