The Next Computer Art

Since the inception of computer art, there has been a consistent attitude of disregard for that medium from the traditional fine arts establishment [1]. Computer artists have responded by establishing their own competitions, starting their own galleries and finding new vehicles to voice their opinions and theories, this journal being a prominent example. Recently, however, a schism has developed among computer artists themselves. Ironically, it stems from a dispute between artists who rely for the most part on image capture, enhancement and composition using drawing and painting "tools" that simulate traditional artist's tools, and algorithmic artists who exploit the intrinsic capabilities of the computer by developing software to create art that can only be realized with the aid of a computer. Their dispute is centered around the role of the software specialist in the art-making process, and its underlying theme is the ideology behind such software.

Consider these observations by Nancy Hays, following an interview with artist Diane Fenster:

> From the beginning, Fenster said, she's been fighting the bad press engendered by original computer art, done by programmers rather than artists. People must view computer art based on content and artistic merit rather than just on the algorithms used, she said—there must be room for both. She accused the SIGGRAPH and Prix Ars Electronica competitions of a bias toward technique rather than content. The label is part of the problem—if judging is based on technique, it is not an art competition but a programming or engineering competition [2].

Now contrast these remarks with the words of John Lansdown, which appeared in a recent *Leonardo* editorial, arguing in favor of algorithmic art:

> I often find myself wondering what fundamental—rather than peripheral or incidental—role information technology has actually played in the works described. The test I conduct in considering this matter is two-fold: (1) Could this work have been done without the assistance of the computer? (2) Would it have been *thought of* without the existence of the computer? [3]

I agree with Lansdown's plea for more algorithmic art, and confess an aversion to the endless sea of "scan-and-collage" digital art currently engulfing the computer art scene, but I find there is more to the story than first impressions will allow.

Some time ago, I participated in several "bridge" projects uniting traditional artists and software experts [4], because I believed that the differences between digital artists following traditional paths and algorithmic specialists searching for new paths could be attributed to a misunderstanding between the two cultures, as evidenced by the following anonymous comments in an Association for Computing Machinery (ACM) newsletter:

> [Many] artists feel as if the programmers who wrote their software or code on a joint project are not worthy of note because they are seen merely as technicians, similar to "the guy who develops my film" and not equal contributors [5].

Currently, there seems to be little hope of reconciliation. Digital fine artists are trying to gain acceptance within the traditional fine arts world, while algorithmic artists, requiring a different audience, seem to drift aimlessly or suffer in silence. But new opportunities are looming on the horizon as the exponential growth of information technology creates a sophisticated, technically literate audience for the algorithmic practitioners. Based on my experiences giving algorithmic art presentations, I believe

this audience would flock to algorithmic art under the right circumstances. However, one harsh reality needs to be faced. Fenster makes a valid point, "People must view computer art based on content and artistic merit rather than just on the algorithms used" [6]. In fact, Lansdown's lone algorithmic example is precisely what Fenster would object to most [7].

We now come to the heart of the matter. Algorithmic artists have failed to distinguish between interesting visualization of numerical algorithms on the one hand, and algorithms developed for artistic creation on the other. A cadre of computer artists has always used techniques more fundamental to the discipline of computer science than to traditional fine art to generate their procedural art [8]. Mark Wilson's book [9] offered how-to instructions as well as inspiration to the programming community at large. His techniques for embedding drawing primitives in nested procedures that are invoked using a skillful combination of user-generated parameters and parameters controlled by a pseudo-random number generator met with considerable success. Roman Verotsko [10] has won acclaim and admiration for works on paper that are created by using a library of 1,300 hand-crafted drawing subroutines [11].

These artists, motivated by artistic content, labor in the face of the parallel explosion in fractal art, which furnishes my example of the kind of distinction that algorithmic artists have failed to make. Fractal art has been widely disseminated to the public, but has not spawned any technical movement within the computer art world. At one extreme are the works of Rhonda Roland Shearer, which appeal to the metaphor of the fractal image but without any understanding of chaos theory, iterated function systems or self-similarity [12]; at the other extreme are mathematically sophisticated treatments such as those found in the books by Barnsley [13] and Peitgen et al. [14,15], which I find difficult to view as efforts at artistic expression. In the middle are Pickover's attempts to promote fractal art from the moral high ground of "scientific visualization" by stressing the themes of pattern and beauty, which again seem to miss their mark [16]. Similar appeals to the themes of pattern and beauty—also coaxed passively from equations—whether in the context of linking art and science by following in the footsteps of Mark Wilson (see Molnar and Molnar [17]) or within the framework of design tools (see Passoja and Lakhtakiz [18]), or as a by-product of mathematical research (see Golubitsky and Frame [19]) offer further examples where contention arises between digital fine artist and algorithmic artist.

What is the litmus test? Jean-Pierre Hébert, an algorithmic artist who has not received the attention he deserves, is on the right track. His works possess the strong stylistic elements that must be present in successful algorithmic works, and he writes:

> The process is a convergence of mathematics and drawing. . . . This said, only the appearance of the process has been outlined. The mind and soul behind it remain, in fact, why and how the work is done, essential while the process remains circumstantial. Art—with computer [20].

One might argue that the schism to which Fenster and Lansdown draw our attention would simply disappear if we define "computer artist" in such a way that both the digital fine artist and algorithmic fine artist are seen to be engaged in the same paradigm—that of artist working at the meta-level. This philosophy was voiced by Rosebush in *The Proceduralist Manifesto*. There proceduralism is explained as

> [art] made by employing scripted, notational directions that specify processes and parameters; the picture is produced by executing these directions, rather than by drawing it directly [21].

Such rhetoric will not solve the dispute. Throughout the short history of computer art there have always been individuals and groups who militantly wished to define computer art as programs and programming. But even when the message is forceful, as the following quote from Craig Hickman's "Why Artists Should Program" illustrates, "I believe that the most exciting and important applications for computers in the arts will come from artists programming" [22]. It is naïve to think that more traditional digital fine artists will be so easily dismissed. What has altered the uneasy alli-

ance between digital fine artist and procedural fine artist and given hope to the algorithmic art movement is a new generation of programmer artists. This changing of the guard is signaled both in sentiment and in ideology by Lev Manovich:

> [We] need to completely reevaluate the term "computer art." The term presently refers to the making of art with the help of a computer, the art to be enjoyed by human observers. The artist is the one who makes the creative choices. This Romantic paradigm reaches its extreme in the recent trend of artificial life art, where the computer is programmed to simulate the laws of evolution, mutating images to create endless new combinations, while the artist assumes the role of God, selecting which of these images will survive [23].

Alvy Ray Smith, a SIGGRAPH computer art pioneer, also senses that significant change is in the wind: "Artists should be helping us explore complexity, showing us the meaning of computability" [24].

We have now reached the point where the algorithmic artist can be clearly differentiated from the digital fine artist, from the mathematical visualization specialist and from the programmer-technician. But before introducing these next computer artists, one more point needs to be addressed. Most algorithmic (i.e. programmer) artists do not view their art software as a reusable commodity. To generate their images they must constantly revise and reinvent their algorithms. Much of the appreciation of their art comes from recognition of their skill and originality in designing and manipulating their algorithms. Algorithmic artists do not create "tools" for digital artists, they create algorithms for making their art. With this understanding, I would augment Lansdown's test for evaluating (algorithmic) computer art by adding: Is the work recognizably computer generated? Are the algorithms of interest? Is the intent of the artist evident?

Manovich heralds the shift in emphasis from the digital fine artist who eschews programming to the software expert who creates a new paradigm of art by choice. The key to this paradigm is algorithms that generate images lying in an image space, coupled with search engines that explore the resulting non-linear image space. This is the basis for Artificial-Life (A-Life) art or genetic art. I suggested a formal model in my paper "New Directions for Evolving Expressions" [25], in which I named the two essential components as: (1) an encoding of all, or nearly all, images—the genotype parameterization—in the space and (2) a search algorithm for visiting all, or nearly all, visually interesting regions of the space.

The "proof of concept" was established by Richard Dawkins [26] who composed whimsical simulated organisms. Manovich's remarks about A-Life art, however, refer not to Dawkins but to William Latham [27], who, with the computer graphics expertise of Steven Todd, developed the "Mutator" system to explore three-dimensional, texture-mapped, and ray-traced images of artificially evolved creatures on an IBM supercomputer, and to Karl Sims [28], who, with the help of the graphics team at Thinking Machines Inc., developed and implemented on a Connection Machine supercomputer a rendering technique for searching an image space associated with mathematical expressions. Central to the art by choice process is the user interface that allows artists to select genomes that evolve and mutate through the use of evolutionary algorithms to produce the next set of genomes for viewing and culling. It is difficult to describe the experience of searching through hundreds of images per hour while trying to guide a search through a space of evolving images. Could it have been done without a computer? No! Would it have been thought of without the existence of the computer? No! Is the work recognizably computer generated? Yes! Are the algorithms of interest? Yes! Is the intent of the artist evident? Yes!

Such algorithms are not easy to implement. Few have tried. Therefore recent advances made by Steven Rooke [29] are to be commended. Equally impressive is the brilliant work of 1994 Prix Ars Electronica winner Michael Tolson who takes the idea to the next level. Tolson "breeds neural nets" to produce digital filters—image-processing algorithms—therefore, instead of searching within a space where each element

corresponds to a different image, Tolson searches within a space where each element corresponds to a different algorithm that can modify any collection of images [30].

To conclude this discussion, I will attempt to answer a few questions. Are algorithmic artists fine artists? I think the answer must be a resounding "yes." Should traditional artists have programmers at their beck and call? My answer to that is that algorithm development undertaken for the sake of computer art does need to be legitimatized and encouraged for a reason traditional artists should appreciate most—to empower people involved in art. Still unresolved then is the question of how computer art should be judged. Fenster nicely draws our attention to the dichotomy between art appreciated for its classical artistic content and art appreciated for its technical achievement. Must it be acknowledged that computer art will be judged with a bias based on technical understanding? Should there be two categories—art that presumes to make a statement on one hand, and art that flaunts technical innovation and achievement on the other? This seems unfair, unwise and unrealistic. Lansdown offers constructive criteria and I have added my own. I submit that it is reasonable to adopt these, or similar, criteria.

In closing, I want to note the words written by avant-garde enthusiasts of art by choice commenting first about its future promise:

> This new form of art, although controversial because the artist sort of lets the computer do all the work, is full of promises: For the first time, it is possible to convey the bottom of our dreams—for the first time it is possible to really visualize the creatures that live in there [31].

And second about its future significance:

> [If] you are an artist, AL [A-Life] opens a world of new experiences to you: it complements the traditional artistic techniques by extending the scope of art-as-it-is to the wider scope of art-as-it-could-be, where everything which is in your imagination, even deep down inside your subconscious, can be re-created [32].

Perhaps, then, the *next* computer art is the algorithmic art practiced by the artist engaging in art by choice.

Acknowledgments

I wish to thank the Visualization Lab at Texas A&M University for hosting a Fall 1994 sabbatical, where the first draft of this piece was completed, and Don House for his encouragement.

GARY R. GREENFIELD
Department of Mathematics & Computer Science
University of Richmond
Richmond, VA 23173
U.S.A.
E-mail: <ggreenfi@richmond.edu>.

References and Notes

1. James Elkins, "Art History and the Criticism of Computer-Generated Images," *Leonardo* **27**, No. 4, 335–342 (1994).

2. Nancy Hays, "An Artist's Thanks to the Medium," *IEEE Computer Graphics and Applications* **14**, No. 5 (1994) p. 7.

3. John Lansdown, "Why Not More Algorithmic Art?," *Leonardo* **31**, No. 4 (1998) p. 245.

4. Gary Greenfield, "Case Study: A Sculptor-Programmer Collaboration," Univ. of Richmond Technical Report No. TR-93-03, 1993.

5. Anonymous, "Just an opinion," *ACM SIGGRAPH Education Committee Newsletter* **4**, No. 4 (1994) p. 10.

6. See Hays [2] p. 7.

7. Lansdown refers in his *Leonardo* editorial to a 1972 number theoretic algorithm (from Roger Saunders) for visualizing a computation being performed and displayed on a cellular automaton. See Lansdown [3] p. 245.

8. For reasons of space, my examples are restricted to algorithmic art displayed as two-dimensional framed art, and therefore I have made no mention of visionary algorithmic work in the areas of web site design, multimedia, computer-assisted sculpture, virtual-reality installations or animation.

9. Mark Wilson, *Drawing with Computers* (New York: Perigee Books, 1985).

10. Roman Verotsko, "Epigenetic Painting: Software as Genotype," *Leonardo* **23**, No. 1 (1990).

11. Verotsko and five others—Charles Csuri, Helaman Ferguson, Jean-Pierre Hébert, Manfred Mohr and Ken Musgrave—maintain a web site at <http://www.solo.com/studio/algorists.html> for the Algorist School.

12. Rhonda Roland Shearer gave a talk entitled "How Fractals and Chaos Will Change Culture" at the Art & Mathematics Conference, 7–11 June 1993, State University of New York, Albany, New York, U.S.A. See also Rhonda Roland Shearer, "Chaos Theory and Fractal Geometry: Their Future Impact on the Future of Art," *Leonardo* **25**, No. 2 (1992).

13. Michael Barnsley, *Fractals Everywhere* (New York: Academic Press, 1988).

14. Heinz-Otto Peitgen and P.H. Richter, *The Beauty of Fractals: Images of Complex Dynamical Systems* (New York: Springer-Verlag, 1986).

15. Heinz-Otto Peitgen and Dietmar Saupe, eds., *The Science of Fractal Images* (New York: Springer-Verlag, 1988).

16. C. Pickover, "Beauty and the Bits", in *Mazes of the Mind: Computers and the Unexpected* (New York: St. Martin's Press, 1992).

17. F. Molnar and V. Molnar, "Noise, Form, Art," *Leonardo* **22**, No. 1, 15–20 (1989).

18. D.E. Passoja and A. Lakhtakiz, "Carpets and Rugs: an Exercise in Numbers," *Leonardo* **25**, No. 1, 69–71 (1992).

19. M. Golubitsky and M. Frame, *Symmetry in Chaos: A Search for Pattern in Mathematics, Art and Nature* (Oxford, U.K.: Oxford Univ. Press, 1992).

20. Jean-Pierre Hébert, "Visual Proceedings," *Computer Graphics Annual Conference Series* (1997) p. 34.

21. Judson Rosebush, "The Proceduralist Manifesto," *Leonardo* **22**, No. 5 (1989) p. 55.

22. Craig Hickman, "Why Artists Should Program," *Leonardo* **24**, No. 1 (1991) p. 49.

23. Lev Manovich, "The Engineering of Vision and the Aesthetics of Computer Art," *Computer Graphics* **28**, No. 4 (1994) p. 262.

24. This remark was made by Alvy Ray Smith, panelist, "Computer Technology and the Artistic Process—How the Computer Industry Changes the Form and Function of Art," panel discussion presented at SIGGRAPH '94 conference, Orlando, FL, U.S.A. (29 July 1994).

25. Gary Greenfield, "New Directions for Evolving Expressions," in R. Sarhangi, ed. *1998 Bridges Conference Proceedings* (1998) pp. 29–36.

26. Richard Dawkins, "The Evolution of Evolvability," Christopher Langton, ed., *Artificial Life* (Reading, MA: Addison Wesley, 1989) pp. 201–220.

27. Steven Todd and William Latham, *Evolutionary Art and Computers* (San Diego, CA: Academic Press, 1992).

28. Karl Sims, "Artificial Evolution for Computer Graphics," *Computer Graphics* **25**, (1991) pp. 319–328.

29. D. Voss, "Sex Is Best," *Wired* (December 1995) pp. 156–157.

30. S.I. Gallant, *Neural Network Learning and Expert Systems* (Cambridge, MA: MIT Press, 1993).

31. E.W. Bonabeau and G. Theraulaz, "Why Do We Need Artificial Life?" *Artificial Life* **1**, No. 3 (1994) p. 322.

32. Bonabeau and Theraulaz [31] p. 323.

In Memoriam

John Lansdown, long-time *Leonardo* Editorial Advisor and supporter of *Leonardo,* died 17 February 1999 at the age of 70. An architect, Lansdown became interested in and pioneered the use of computers to aid architectural drawing during the early 1960s, creating CAD (computer-aided design) applications. Lansdown was a founding member of the Computer Arts Society and participated on the boards of a number of computer organizations and publications, authoring numerous books and articles on the topic of computers and art. He co-founded a company responsible for innovations in computer animation for film and television and created and exhibited algorithmically generated images and animations. Lansdown was Professor Emeritus at Middlesex University's Centre for Electronic Arts, U.K., having been a member of its faculty for more than 10 years.