

PARAGRAPHS ON COMPUTER ART, PAST AND PRESENT

Frieder Nake
Informatik
Universität Bremen
P.O. Box 330 440
Bremen
Germany
nake@informatik.uni-bremen.de
www.agis.informatik.uni-bremen.de

Sol LeWitt published “Paragraphs on Conceptual Art” in *Artforum*, June 1967. They became an influential theoretical text on art of the twentieth century. They played the role of a manifesto even though they appeared when their topic – concept over matter – had already existed for about a decade. Digital computer art had had its first exhibitions in 1965. It seems it never produced a manifesto, with the exception, perhaps, of Max Bense’s “Projects of generative aesthetics” (1965, in German). Since computer art is a brother of conceptual art, it is justified in a late manifesto to borrow the style of the old title.

An art movement that is by now 45 years old and that has gone through a tremendous development, cannot create *ex posteriori* its manifesto. However, *A Software Manifesto* was also written only *after* software and information technology had become the most important technology of our times. The following paragraphs may be read as a belated manifesto.

1. There are no images now with no traces of digital art. Digital Art exists as computer art, algorithmic art, net art, web art, software art, interactive art, computational art, generative art, and more. When it made its first appearances, in Stuttgart and New York, the name “computer art” was thrown against art history and into the faces of art critics. It was a proud name and a bad one. “Algorithmic art” would have been the correct term. The superficial “computer art” disguised the revolutionary fact: the algorithmic principle had entered the world of art.
2. The algorithmic principle is the principle of computability. Whatever exists in the domain of computability, exists insofar as it is computable. Alan Turing, Alonzo Church, and others had in the 1930s saved mathematics as the only discipline of the human mind that can say clearly what it says. Those heroes had

clarified the concept of computability. They had thus created a new basis for mathematics. Soon after, the computer appeared as the machine to turn science into engineering. There had, of course, before been devices for mechanical calculation, but no computing automaton.

3. The computer was about twenty years old, when computer art appeared. Was this late? Was it early? It was in time. The times were times of deep social unrest. The hypocrisy of the war generations came under attack, and Karl Marx had a revival. Ideas of the cultural revolution from far East conquered young minds. The algorithmic revolution started its long march overturning all of the technological infrastructure.
4. Three years later, computer art was recognized internationally. In 1968, two exhibitions became the forerunners of the development of digital media. One was called *Cybernetic Serendipity. The Computer and the Arts*, at the Institute of Contemporary Art in London. The other one was *Tendencies 4. Computers and Visual Research*, at Galerije Grada Zagreba in Zagreb, Croatia. *Serendipity* established the event component of digital media, and linked to the computing industry. *Tendencies* established the research component of digital media, and linked to the world of art.
5. Computing machinery in its form as digital media incorporates three great principles: computability, interactivity, and connectivity. Computability appears in the arts as algorithmic art. Interactivity appears as interactive installation. And connectivity appears as net art or software art.
6. Earliest computer art (as digital art, and not as electronic art) is art from a distance. It is art done by brain, and not by hand. It liberates the artist from the limits of handicraft skills. It automates the production of the perceivable, material component of the work. The artist in algorithmic art creates an entire class of individual works. He or she is an artist insofar as she works in the realm of possibilities and potentials, not of realities and facts. The work of art in algorithmic art is the *description* of an infinity of possible works. They all share some common features that the mind can discern, even if the eye cannot see any similarities. The description is a sign of signs.
7. Computer art is almost entirely happening in the semiotic domain.
8. Algorithmic art has denies the concept of a masterpiece. This is to say that in algorithmic art, there cannot be a masterpiece any more in the traditional sense of the word. Each and every individual piece of algorithmic art is no more than only one instance of the potentially infinitely many from the class of works

defined by the algorithm. The tragedy is that the algorithm itself does not often show visual qualities. Its qualities are the potential to generate visual works. But each of its visual products is a shadow only of the algorithm. It is one of its traces, a left-over, a consolation for those who need to see rather than think. If you want to find the masterpiece, you must compare algorithms. Critics and art historians are not prepared to do this. Nor is anyone else.

9. Computer art is conceptual art. However, concepts in computer art are different from concepts in conceptual art. They here appear as operational descriptions. Algorithms are descriptions. They are finite descriptions of infinite sets. They are static descriptions of dynamic processes. These descriptions are operational and executable. I.e. they are text *and* machine, at one time. When the algorithmic artist designs a work (an algorithm), he writes a static text. He may print the text on paper. This shows the text quality of the work. It is a quality for the human reader to perceive and acknowledge. But the text's description is also operational. It can be executed by a computer. When the computer executes the description, it reads it in its own, peculiar way: it realizes exactly what the description requires it to do, and nothing else. Reading always is interpreting. The computer, when reading the operational text, interprets it. Absolutely different from our interpretation, the computer's interpretation is a determination: no freedom allowed. The computer interprets by determining the one and only one interpretation that makes algorithmic sense.
10. Computer art is concept art insofar as it describes an idea and does not show the material work. However, since its descriptions must be operational or computable, the concept can be carried out immediately, without mediating media. If the conceptual artist ever wanted to realize his description of an idea, he would need media of an appropriate kind to do so.
11. Emerging at the same time as conceptual art, algorithmic art as the elder or younger brother clearly went beyond the confines of conceptual art. Concepts and conceptualizations had always been present in art since, without an idea, without a concept, art would not emerge (at least not in modern times). Conceptual art was another step in the continued modern reduction of the work of art. This reduction reached the point of the concept or idea itself. No work exists without a concept at its root. In conceptual art, the concept is considered more important than its realization. Algorithmic art goes the other way. Ideas and their descriptions, in algorithmic art, must be codes that incorporate their own execution. Where conceptual art dances around the possibility of, perhaps, realizing a piece and drawing pleasure from imagining it, algorithmic art immediately delivers the conceptualized piece free of charge. It could go on

realizing works of the same concept for centuries. Harold Cohen, Manfred Mohr, and Roman Verostko, and dozens of other algorithmic artists know this.

12. The algorithm is the concept in its strictest form of description. Conceptual art usually is “free from the dependence on the skill of the artist as a craftsman”. Some years before Sol LeWitt wrote this in 1967, algorithmic art had already eliminated the skilled craftsman. We see: algorithmic art is the final form of art in times of industrial production. Beyond all craftsmanship and aura, the work is produced automatically. This comes at the price of the artist turning himself into an engineer. What the futurists and others may have dreamed of vaguely, becomes deed in the algorithmic age.
13. “The idea becomes a machine that makes the art.” Exactly this, dear Sol LeWitt, had happened some years before you wrote it. Insofar, algorithmic art is the mother of conceptual art. Art critics and others should finally become aware of this when trying to study a phenomenon so alien to them.
14. The greatest idea before its time, in computer art, is its *generative* approach. The algorithmic artist does nothing that is not generative. At a second look, nothing is so great about this. Once an artist decides to use a computer for his production, he is bound to design a program (an algorithm). Without an algorithm for art, no algorithmic art. It is as simple as this. The artist turned algorithmic is a generative artist by birth. Not necessarily does he know this. Therefore, the philosopher had to tell him. For this end, Max Bense wrote his short essay, *Projects of generative aesthetics*. In a book of 1974, I mathematically defined the concept of an analytic aesthetics. A synthetic aesthetics is, of course, the inverse of an analytic aesthetics. A generative aesthetics is a computable form of a synthetic aesthetics.
15. The generative principle had existed since Noam Chomsky had studied generative grammars. To him, they were devices not only to describe syntactic structures of sentences in a language. A generative grammar was also capable of producing syntactically correct phrases of a language. Of course, they played an eminent role in the definition of programming languages. Max Bense only borrowed the attribute “generative” from Chomsky.
16. The generative principle was soon forgotten again as a machinic device for the creation of art. Only recently the generative approach has re-emerged. It is now applied in an almost trivial way whenever a computer is used in the course of some creative work. There is generative design, generative architecture, generative art, generative music, and probably more. The question has become: How can you do anything now *without* a computer?

17. The first artists who used digital computers, and thus discovered for art the algorithmic principle, had before been working as mathematicians or engineers. This was just what they had studied. They had access to computers and were, most likely, computer experts, which was a very special kind of expertise in the early 1960s. Almost all critics, journalists, culture types by the time loved to accuse and insult those pioneers of a new kind of art. The pattern of their degrading comments was: “Quite nice, but boringly geometric and constructive. It needs a real artist to create something fine and remarkable.” Pioneers not often reach great accomplishment. They rather create the vision. It was appalling to see how little the experts of art understood when the machinic principle of computation reached out for art.
18. Earliest computer artists had to tame their naked machines. They kept totally erotic relations with them. Their programs were written in machine language (which is immediate binary code). They had to test and run their programs without support by an operating system. Soon enough, this situation improved, and Algol60 or Fortran became the programming languages of choice. Is this important? Yes, it is. The software and hardware support compares with the brushes, oils, acrylics, and whatever other materials and tools an artist may be using outside the digital world. A language of choice would now be Processing or one of the scripting systems. Such symbolic devices still keep the artist at a distance from his visible work. They still require the kind of thinking that is genuinely new, and was an idea before its time. The artist who wants to do computer or algorithmic or interactive or net art and, therefore, wants to program the computer, must learn to *think how the machine would think if it could*. Read this statement twice, memorize it, and then start doing it. An entirely new world will open up for you. It is the world of digital media whose forerunner was algorithmic art. It is the postmodern. You are leaving industrial production. You enter post-industrial performance.
19. Photoshop is a great piece of software. It is a huge collection of marvellous functions in the guise of tools. Other programs are as great. They open options that most of us have never dreamed of. They open the world of the digital Sunday painter.
20. Artistic production requires activities of selection and composition. The artist selects her materials and tools, forms and colors, her elements with which to work. She does so even if not explicitly. The artist works with her elements by combining, connecting, arranging, positioning, transforming them: she composes. Viewed more abstractly, the artist has at her disposal primitive (or elementary) signs (in their material state). She is free to compose them into

groups and systems of signs of signs, and supersigns, etc.: signs of signs of signs ... – always only on the material, syntactical level. Low and high level compositional decisions influence the gradual building of agglomerations of signs up to the highest levels. Decisions in detail, concerning only tiny parts, gain influence in an often miraculous way up to the top of the work. Such a semiotic view of the generative process is helpful when writing programs for art.

21. Computer art shares with conceptual art (and some others) a neglect of materiality. In fact, the revolutionary step at the very beginning of algorithmic art was the total loss of the material dimension. Only after the program had been developed, tested for correctness, and run for production, could the artist finally, if he so wanted, see what he had achieved. The drawing automaton generated, from an abstract encoding of the drawing, a concrete paper version in ink. Drawing with the brain becomes possible only if all material aspects and components are given up. Early computers did not have display units. They may have been interactive in some way. Such interaction was, by the time, not mediated by icons or indices we have become so familiar with to observe whenever we open our notebook. Interaction was mediated by symbols to think. Meanwhile, materiality has returned in form of the “graphic user interface”. It is a fluid kind of materiality. The principle of algorithmic, of interactive, of software art is still omnipresent: “Think of infinite sets, not of their individual representatives.”
22. Was there any individual inventor of computer art? Certainly not. Machines, devices, institutions existed, people worked there with their differing backgrounds and interests, in their various situations and diverse contexts. The idea to make experiments in two dimensions instead of one, with open ends instead of pre-set goals, in a playful spirit instead of consecutive logic, such an idea emerged here and there until it was realized and proven as viable. Such processes took place between 1962 and 1965 at several places in the world, and nobody should be surprised if some day a new name appears besides those that are usually credited for having been the first pioneers.
23. Did Max Bense, the philosopher and writer of concrete poetry, invent or predict computer art? Would this not be a beautiful idea? You may occasionally find indications in this direction in the literature. The title of one of the four volumes of his *Aesthetica* was *Programmierung des Schönen* (Programming the beautiful). The books were first published between 1954 and 1960. The term “program” was used here, as in similar contexts, in a more general meaning of the word than “computer programming”. Generally doing something according to some programmatic rule, principle, or method differs from the program on a computer precisely in the radical requirement of computability. Claude Monet

worked programmatically in many of his series of paintings depicting one and the same object, but under varying conditions. So did Josef Albers in the long series of his *Homage to the square* paintings. Chaotic as the results may look, Jackson Pollock was following programmatic schemas in his drip paintings. The list could be extended, and it could give rise to similar lists in music or in poetry. The principles of series, of experiment, of construction, permutation, variation, transformation have all been used in art long before computers had arrived. The technique of perspective projection, so important in the Renaissance, is proof of a strict constructive schema that all painters had to learn, and from which they decided to deviate when they thought it was necessary. So what came into the world of art in the early 1960s, is the principle of computability. Max Bense was too much a philosopher to surrender to restrictions of computability.

24. Behind computability we discover mathematics. Throughout the centuries, a friendship has existed between art and mathematics. The two stand for the two most basic capacities and activities of the human mind and of human practice: counting and drawing. Both are ways of abstracting from what we experience directly, i.e. bodily. When counting, humans reduce the world to numbers. When drawing, they reduce the world to shapes. In numbers and shapes, the digital and the analog aspects of the world appear. They are aspects only, not objective givens. We decide to look this or that way. Humans did this early in their existence as humans, early in cultural history. The walls of caves show numbers (as groups of strokes), and they show shapes. With the algorithmic description of an operation to be carried out by machine, art gained the option to incorporate mathematical processes into artistic creations. This is a strong idea. Before its time? Certainly before its time if we look at quantities.
25. It took twenty years into the history of the digital computer before digital art appeared. Not really a long time. To force a computer to create two-dimensional drawings, the machine for calculating must be told drawing. The question must be solved, how to draw when the machine at your disposal can only calculate. As we know, this has been solved completely.
26. Randomness is essential for the aesthetics of computer art. It has been suggested that probability distributions and random numbers simulate in a computer program the artist's intuition. By intuition the artist takes decisions during the creative process. An external observer has no chance to say what the reason was for a particular intuitive decision. Other than by random choices, there is no way algorithmically to simulate such internal decisions. Used in clever ways, randomness must not be the same as throwing dice. It can be controlled in many ways such that macro-structures emerge inspite of randomness on low levels of a program. Randomness is much broader a concept than uniform probability.

27. Generative methods can be designed as structured sequences of decisions. Top level decisions concern global aspects of the visual work. Lowest level decisions concern primitive aspects of the visual work. Any number of intermediate levels can be inserted. Conditional probabilities can be used to create local control dependent on neighborhoods. Hierarchical data structures can be used to move up and down between global and local aspects and levels of an image.
28. Early computer art was revolutionary but, at the same time, traditional. It was traditional insofar as it resulted in paper work to be put up on the walls of a gallery. Why use the most modern technology in order to generate the most traditional formats of the art world? Apparently, the activists of the time were hoping for recognition in the art world by sticking to traditional forms. Computer art was, at the same time, revolutionary on all other accounts. It was a radical turn to an aesthetics of the object. The individual human subject simply did not exist anymore, once he or she had set the boundary conditions for the image to be computed. Computer art, in its early years, was radically rational. It was done in thinking, not in dreaming. Computer art remains a rational art of the object even though its appearances of today have hardly anything in common with those of its early years.
29. Computer art left its McLuhan phase when it took on the form of interactive installation. In its McLuhan phase, a new medium – despite its revolutionary break – still has an old medium as its contents. Algorithmic art as paper work on a wall is of this kind. Computer art gained its inherent historic height only when it took to interactive works. The computer is the machine for precise and rapid repetition with small changes. Only when this essential feature is exploited, does computer art become more than a gimmick. In interaction, this is the case. It is also the case in animation. Digital art has become an integral part of virtually all films. The virtues of algorithms for this genre were demonstrated very early by A. Michael Noll, John Whitney, Sr., Ken Knowlton, Bill VanDerBeek, and others. Interactive works needed more developed technology. They are the truly new genre. In them, digital art triumphed. Connectivity is celebrated in the rapid world of software art.
30. The earliest pioneers of algorithmic art worked in solitude. But very soon, cooperation came up. During some great happy years at Bell Telephone Laboratories, exciting things were happening that came out of a group of engineers, mathematicians, and artists. During a few years of the 1960s, they demonstrated many of the technical possibilities that had become available and were waiting to be used. In 1966, also with support from Bell Telephone, *Experiments in Art and Technology*, headed by Billy Klüver and Robert

Rauschenberg, put up a spectacular series of events in New York, but with moderate success only. Even though they continued for some years, cooperation turned out to be not a trivial task if the product was to be of high quality in both aspects, aesthetically and technologically. The amount of shared time, of disciplinary autonomy, and of allowing oneself deeply getting involved with the other's expectation, is tremendous. Early computer art has, perhaps, demonstrated, that *interdisciplinarity* is not really what is needed. Maybe, that's *transdisciplinarity*. It happens in the individual.