

Epigenetic Art Revisited: Software as Genotype

Roman Verostko

Roman Verostko's 1988 paper on "Epigenetic art: software as genotype" was published in *Leonardo* in 1990. For the 1993 *Ars Electronica* exhibition catalogue, *Genetic Art / Artificial Life*, he summarized the substance of the paper as "Notes on epigenetic art". Here, tempered with 20 years' experience as an algorist and 40 years as an artist, he revisits his original views on coded artistic procedures.

The beginnings. For over 40 years, as an artist, I have sought to create works pointing to hidden or unseen reality. Early on I learned to wonder about the marvelous event of things existing "just the way they are" and whether there could be other ways "to be". Within commonplace phenomena I learned to see a marvelous world filled with mystery. My approach to art grew from this sense of wonder about most things and a reverence for the materials of earth. Eventually this wonder came to include circuit boards, computer languages, and the art forms one could explore with simple algorithms.

Terminology. Talking about *code*, *algorithms*, or *algorists* can evoke wrinkled foreheads and blank stares. Let me offer some clarification. An algorithm can be viewed as a detailed step-by-step procedure for carrying out a task. A recipe for baking bread or directions for reaching a specific location can be viewed as algorithms.

Clearly most kinds of instructions are not written to create art. "But," we may ask, "Can we write instructions for creating art?" "Can an artist 'code' a work of art?" Emphatically, "Yes!"

In responding I will use terms like "mind-ear" and "mind-hand" to indicate that the process of writing instructions for making art engages the whole person and cannot be identified with one body part separated from another. Consider Chopin's score for his *Nocturne opus 27*. Chopin created the musical notation or *code* that instructs the performer on how to play the *Nocturne*. The *score*, a unique procedure, provides detailed instruction for playing a specific musical form. It literally embodies a musical idea that originated in Chopin's "mind-ear" in 1835. His mind "conceiving sound" embodies his sense of hearing. When hearing a performance today we assume that Chopin's musical idea, as he conceived hearing it, has reached us via the musical score—an instruction, *an algorithm*. Inasmuch as Chopin's musical idea is adequately represented in the score, and, inasmuch as the performer interprets the score as Chopin intended, then we enjoy an experience of Chopin's "mind-ear".



Coded art. Musical notation may be viewed as a "detailed recipe" or algorithm for a musical art form. Theme and first chords, F. Chopin, *Nocturne*, Op. 27, No 1. C# minor, 1835. © Schirmer, Musical Classics.

The contribution of the performer surely colors each performance in a unique way and is not taken lightly. Even so, the musical score transcends an individual performance and continues to have a meaningful existence over generations. Note that the “musical score” is written in a “code” consisting of symbols specifying time and qualities of sound. In general, when we use the term “code”, we are referring to an instruction or algorithm in its notational form, a specialized language for precisely representing the instruction. Well formed code displays splendid rationality with every detail clearly spelled out. Note, however, that the procedure for creating the code transcends our understanding. We can study Chopin’s nocturne and understand every detail of the score: but we cannot fathom the procedure by which Chopin created the score. *We must be careful not to confuse the procedure by which the artist creates the code with the procedure specified in the code.* The creative process lies primarily in the process of writing the code.

Coded visual form. With these considerations in mind let us turn our attention from coded “musical form” to coded “visual form”.

I am a founding member of an informal group of artists known as algorists.¹ For us, the term algorist applies broadly to any artist who employs original coded procedures for generating art forms. My interest, shared with pen plotter algorists, focuses on coded drawing procedures.

An algorist’s envisioned drawing procedure should not be thought of as a disembodied concept conceived apart from a feeling hand. Just as a composer composes with his “mind-ear” so an algorist composes with his “mind-hand”. As surely as the mind is present to the hand when it draws, so also the hand is present to the mind when it creates drawing procedures.

To achieve this, an algorist, in writing code, addresses aesthetic qualities and limits of media and machine, for example, how paper surface and sizing affect the receptivity of ink. An algorist translates sensibility to these factors into intelligent coded operations. Specifically how does one get from this “mind-hand” drawing as “code” to the actual drawing? In my case, the code, operating on a PC, instructs a drawing machine known as a multi-pen plotter. These machines, designed for engineers and architects, have “drawing arms” that can select from a bank of pens and draw lines precisely as instructed. In

Detail of code drawn lines.

Coded pen and ink drawing, 1987 (18 cm. by 26 cm).



1987 I adapted paintbrushes to fit the drawing arm and have written special procedures for executing brush strokes. For the most part my works employ thousands of pen strokes with only occasional use of brush strokes.

The drawing arm and recursion. One is tempted to view the plotter's drawing-arm as a simulator of the artist's physical hand. This is not the case. Rather it executes the "mind-hand" of the artist. The algorist's "mind-hand" embodies precision with extensive drawing procedures that exceed the reach of a physical drawing hand. We could say the artist's mind-hand draws via the machine's drawing-arm something like the power shovel operator's hand shovels with the power shovel's shovel.

These abilities can be original drawing manoeuvres that exceed the capacity and dexterity of the human hand because they employ functions capable of unlimited iteration while improvising on themselves. The mind-hand can conceive of a drawing loop that looks back at what has been drawn and sets a procedure for improvising on the next drawing movement. The artist can create procedures for improvising upon improvised moves for each step along the way. This kind of "drawing-mind" engages the deep well of recursive procedure. Through experience the algorist learns how to set improvisational rules to achieve aesthetic preferences.

Recursive procedures are not new. The expression for the Fibonacci number series would be a simple recursive function, $[n+(n+1)=n+2]$. What is new is the ability to execute code for extensive recursive iteration yielding dimensions of improvisation in drawing that exceed what our minds can contain or what our hands can draw. This capability would have been, to my mind, a heavenly delight in the hands of artists like Wassily Kandinsky, Piet Mondrian and the Pevsner brothers. Recursion lies at the very heart of algorist art. This is how self-similar qualities of the drawing lines and brush strokes in my Epigenesis mural become imbedded in drifting pen stroke clusters that mirror each other.² Herein lies the unique feature of computing power coupled with an artist's coded procedure.

Studio practice on the new frontier. For over a quarter century I have marveled at the power of coded procedure for experimenting with visual form. Artists who have integrated their art with original coded procedures are on the threshold of a new frontier. In tapping this frontier they are confronted with two great hurdles. One has been the task of translating form-generating ideas into a practical working code. The other has been the search for appropriate methods for producing the work in a tangible form. The code as a form generating idea is only half the struggle. To succeed as art, the code must be able to generate tangible form—something one can see, touch, feel or hear. Some years ago an artist friend, standing by an elm tree, asked, "What gives this tree such a



First coded brush strokes in 1987. Oriental brush mounted on drawing arm. Houston Instruments DMP 50 series pen plotter. Plotter driven with PC operating on DOS.

powerful presence?" And he answered, "You can touch it—feel its surface—its 'being here'!" Because aesthetic experience involves the senses, one cannot separate art from its material embodiment. In addressing human sensibilities artists embrace many diverse media including those emerging in cyberspace culture. My commitment has been clearly bent towards creating drawings and paintings on paper. My procedures are attentive to aesthetic qualities of papers, inks, pens, brushes and the manner of presentation. For me, the finished work should have an aura that invites the casual viewer to pause for a moment, sensing that the work, as a human endeavor, goes beyond material concerns. Even so, the process holds an unusual fascination in and of itself.

Studio Scribes. Many people who visit my studio are surprised when they see several engineering plotters cabled to a network of computers, one reserved for my experimental work and the others for generating art. When a plotter is working, visitors often stand with eyes glued on the drawing arm which, with a "seeming intelligence", draws precisely, surprisingly, and without hesitation—executing literally thousands of lines, and exuding an uncanny presence!

My first pen plotters, with names like Brunelleschi and Alberti, are now retired. Occasionally I bring Brunelleschi out of storage and have him make a small drawing. Although maintaining register is difficult and the movements are a bit coarse he can still execute a drawing to my approval. Over the years I have come to view my studio as a scriptorium with a network of electronic scribes. These scribes execute instructions tirelessly. Occasionally a specific work may require several days of drawing. Large pieces remain very difficult to achieve because of pen or hardware failure.

One can write code for unlimited sequences and a vast array of forms, but the drawing process must be coded in terms of specific drawing materials and tools. The code specifies procedures that operate with these tools in real time and space. The actual procedure must mesh with the limits and aesthetic qualities of paper and drawing instruments. At its best the code optimises the operation of tools and amplifies the aesthetic qualities of paper, ink and the character of the marking instrument.

Coping with these technological limits requires patience and experience with endless hours of trial and error. The jewel in this quest is the piece of code that works—like the acorn that can grow into a tree, the successful code literally "grows" a work of art!

This procedure for making art remains essentially the same today as in the 1980's when I wrote the original paper "Epigenetic Painting: Software as Genotype".³

A Personal System. The software, with a computer and a plotter, constitutes a complete personal system. Experimentation with paper, inks and plotting procedures has been extensive. Through many years of trial and error the system has evolved into a unique set of procedures with a language of its own. The software, under ongoing development, is an integrated program of algorithms written in elementary BASIC with plotter commands in DMPL. Routines have grown to thousands of lines.

```
RAD(SID)=(INT(RND*(RMAX-RMIN))+RMIN)
THETA(SID)=INT(RND*(DEGEND-DEGBEG))+DEGBEG
XQ=X1:YQ=Y1
JX(SID)=X1+INT((RAD(SID)*STRETCH)*COS(THETA(SID)*(PI/180)))
JY(SID)=Y1+INT(RAD(SID)*SIN(THETA(SID)*(PI/180)))
X1=JX(SID):Y1=JY(SID)
```

Six lines of code from a loop that specifies the radius, an angle and the coordinates for control points in a pen stroke

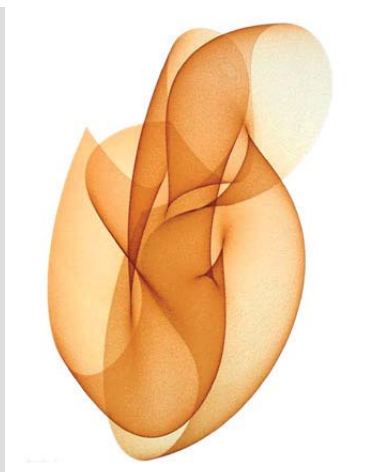
Hodos. I have named the software *Hodos* (ὁδός), a Greek word meaning *path* or *way*. Note also that *meta-hodos* (μεθόδος) is the classical root for “method”. *Hodos*, as a method of drawing, can be identified easily with my early work conceived as “pathways.” My studio, named Pathway Studio, received an oriental seal carved by the distinguished shufa master, Wang Dong Ling. Wang, awed by the brush strokes made with the “electric brain”, chose the classic expression *xiao jing zhai*, “little footpath studio”, as the seal for my studio. Many works executed in the studio receive this seal.⁴

Families of form. Form generating routines within *Hodos* create works with distinctive features related to those routines. Works created with the same or similar algorithmic routines bear a familial resemblance to each other. *Hodos* can be set for a specific paper size, palette and sequence of routines to create a radically new kind of limited edition. It can generate a family of forms with each one being simultaneously related and yet uniquely “one of a kind”. Works share strong familial features because they share the same algorithmic parents. Familial variety occurs because the code draws on random decisions within shared parameters. Parameters within which decisions are made can be: (a) established as *not more than* or *not less than*; (b) wide open with *no upper or lower limit*; or (c) weighted as a *percentage more or less than so much*. Parameter settings establish factors like angle, radius, coordinates, scale and color. Familial resemblances between the works vary more or less greatly depending on how many parameters are set to change and to what degree.

For example, in 1990, with *Hodos* and my assistance, my wife Alice Wagstaff plotted 125 original pen and brush drawings as frontispieces and 125 as end-pieces for a limited edition. The leather bound edition of 125 included other algorithmic drawings illustrating Chapter II of George Boole’s 1854 classic, *An investigation of the laws of thought*.⁵ One set of routines with a single set of parameters was employed for the frontispieces and another for the end-pieces. We believe this is the first time that a limited edition of a book engaged this kind of process for illustrations in a publication.



Two pen & brush drawings from a family of 125 originals. These are frontispieces from 2 books in the limited edition on chapter II of George Boole’s 1854 classic work on the laws of thought. (St. Sebastian Press, Minneapolis, 1990). See: www.verostko.com/boole.html



Cyberflower IV, pen & ink drawing on paper. Code generated with *Hodos*. Roman Verostko, 2000, 54 cm. by 74 cm

Epigenetic art: software as genotype. These procedures are of a different order than traditional procedures. What shall we label this process? This was a question Alice and I posed in 1986. In our search we found ourselves returning again and again to similarities between biological processes and coded procedures. We settled on the term *epigenetic* and this became the basis for my 1988 paper.

Epigenesis refers to the process whereby a mature plant or phenotype grows from a seed or genotype. For example an acorn embodies the genotype or code that contains all the information needed for growing a mature oak tree. Given the proper environment it can, in time, grow into a mature tree. Through the process of epigenesis, the acorn grows into a mature tree referred to as phenotype.

By analogy my software or code, likened to genotype, contains all the information needed for generating an art form. Given the proper studio environment, the software literally grows from a coded procedure into an art form. By analogy to biological epigenesis, this process may be viewed as *epigenetic*.

Clearly any coded procedure that has all the information necessary for generating an art form could be viewed as *epigenetic*. The more general term used recently is *generative art*.

Content and meaning. For over a quarter century epigenetic art has been creating the icons of our information age. One could think of these icons as diagrams or visual analogues to the coded procedures by which they were made. The essential character of each finished work is derived from the “form-generating-procedure” or “algorithm” acting as genotype. For this reason one could say that the finished work is an epiphany, or manifestation, of its generator, the *code*. For me each work celebrates its code, especially the recursive routines that shaped its character. It is noteworthy that such procedures hold much in common with processes associated with crystallization and genetics.

But these are generalities. How shall we approach specific work like an algorist pen and ink drawing, for example? Such drawings, at their best, will bear the imprint of the artist's coded procedure. Just as a painter's brush stroke may bear the unique mark of the painter's hand, so the lines in an algorithmic drawing reveal the distinctive qualities of the composer's algorithmic “mind-hand”. Through a unique convergence of conceptual innovation and knowledge of materials, the code created by each algorist engenders a personal style that is present with each drawing. The linear forms of an artist like Jean Pierre Hebert will celebrate themselves with self-similar meanderings coursing over the paper in a manner quite distinct from the character of the lines generated by my code or that of Manfred Mohr. Contemplating their forms reveals something, in each instance, of their creator's “mind-hand”. Through these works we are given a glimpse into the mysterious nature of the algorist's inner world.

In general these works provide a window on unseen processes shaping mind and matter. By doing so they become icons illuminating the mysterious nature of self, earth and cosmos.

-
- 1 www.verostko.com/algorist.html
 - 2 www.verostko.com/st/mural.html
 - 3 www.penplot.com/epigenet.html
 - 4 www.penplot.com/seal.html
 - 5 www.penplot.com/boole.html