Computing Drawing: Programming a Vintage Pen Plotter

by

Carl Lostritto

Master of Architecture
University of Maryland, 2008

B.S. Architecture
University of Maryland, 2006

SUBMITTED TO THE DEPARTMENT OF ARCHITECTURE IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE OF ARCHITECTURE STUDIES
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2012

Signature of Author: ______________________________

Department of Architecture
May 24, 2012

Certified by: ______________________________

George Stiny, PhD
Professor of Design and Computation
Thesis Supervisor

Accepted by: ______________________________

Takehiko Nagakura, MArch/PhD
Associate Professor of Design and Computation
Chairman, Committee for Graduate Students

Computing Drawing: Programming a Vintage Pen Plotter

by

Carl Lostritto

Advisor

George Stiny, PhD
Professor of Design and Computation

Reader

Nader Tehrani, MAUD
Professor of Architecture
Department Head

Computing Drawing: Programming a Vintage Pen Plotter

by

Carl Lostritto

SUBMITTED TO THE DEPARTMENT OF ARCHITECTURE ON MAY 24, 2012 IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE OF ARCHITECTURE STUDIES

ABSTRACT

The drawn artifact and the act of drawing are uniquely suited for design thinking. Specifically, drawings that were traditionally crafted “by hand” are prone to qualities that promote a productive multiplicity of interpretations. These qualities are often incompletely characterized using terms such as “fuzzy” or “loose.” Digital output, however, is biased toward the notion of re-production. Representation in design, as a result, has become image-centric.

This project explores a method for computing drawing (and the converse, drawing computing) by programing a vintage pen plotter. An apparatus that spans from the computational to the material allows for the incorporation of the desirable qualities of the “hand” drawing into a digital process. The same limitations that led to the obsolescence of the pen plotter lead to an integrated relationship between process and project. Pen plotters demand linear (rather than pixel) information. Imperfections resulting from ink-filled pens making contact with paper at various speeds mandate the consideration of time.

A range of computational methods for representing line and making drawing are documented and implemented. A set of 32 drawings are framed in terms related to their making, and then evaluated in terms of their implications for architectural representation.

Thesis Supervisor: George Stiny
Title: Professor of Design and Computation

Acknowledgements

In addition to my thesis committee, this project received support from the following individuals.

During the period in which I was his teaching assistant, William O'Brien Jr. and the students of his Spring 2012 studio hosted me for a presentation of this work-in-progress. Besides that discussion, my participation in that studio, which included an emphasis on drawing, was influential and motivating.

I received regular input from many members of the Design and Computation Group during this work. In particular, Duks Koschitz provided focused critique and frank advice.

1. Introduction

What is computing drawing? How can can a programmed pen plotter demonstrate that computing drawing is possible and meaningful? This introduction section defines drawing and identifies its contentious position in contemporary discourse. The motivation for using a pen plotter controlled computationally is presented in this context. The structure of the apparatus is then documented in detain and related work is discussed.

1.1 Defining Drawing as space for research

Drawing is so ubiquitous that the definition of the word, "drawing," often remains unconsidered, flexible, or inconsistent within architectural discourse. At a 2012 Yale University Symposium, "Is Drawing Dead?" the explicit, headlining question was addressed with some contention. Implicitly, however, conflicting definitions revolved around the debate. A sampling of the implied definitions of "drawing" offered at the two and a half day event included:

> Drawing is any representation that exists in two dimensions (drawing is image, and therefore very much alive and thriving)

> Drawing is any 2-D representation that is constructed (in process and order) rather than generated (in which case drawings are long dead)[1]
>
> Drawing is the process of making marks on paper with a hand-guided device (which may or may not be pens or pencils)
>
> Drawing is making lines in any manner that is more considered and restrained than the "sketch" Drawing is 2-D content in which the line has more geometric meaning than symbolic meaning
>
> Drawing is making lines in any manner that is more loose and open than "drafting"
>
> Drawing is any image made predominantly of lines

Cammy Brothers, a historian, noted saliently that before and even during the Renaissance, a period unusually associated with the codification of drawing conventions in architecture, drawing was a "space for research."[2] She calls architects to arms in an effort to renew the role of drawing as experimental territory, considered parallel to built architecture and emancipated from it's position as conventional representation. Her view could be interpreted as rejecting the urgency of the question "is Drawing dead?" in favor of a more nuanced, "What should drawing be, now?"

Still, a definition of drawing, even one that is acknowledged as contingent, is required for legitimate research to occur. Given that obtaining an accurate definition that would be robust across disciplines and histories would be a research project in itself, Marco Frascari comes close to a definition of drawing by arguing an emphasis on it's salient feature: "facture."[3] Important in his schema is that "drawing" must be relevant as process and product. The drawn object must in some way convey (or betray) the drawing action. This definition could be narrowly interpreted as excluding the digitally produced line, which tends to describe the geometry rather than the mark. Frascari does, however, heavily rely on an exploration of hypothetical apparatuses, which fold operation, indirection and translation into a broad consideration of making. This leaves "drawing" open to potentially include computed actions that are manifest as image. Deanna Petherbrige, Professor of Drawing at Cambridge University, also devotes extensive scholarship to the nature of drawing's resistance to definition.[4] She proposes that drawing cannot be strictly defined, but offers, instead, the framing of drawing in terms of that which it "approaches" but unambiguously

[1] Cohen, "Plan Vs. Drawing."
[2] Brothers, "Experience and Fantasy in Renaissance Drawing."
[3] Frascari, Eleven Exercises in the Art of Architectural Drawing. 10
[4] Petherbridge, The Primacy of Drawing. 15

cannot become.[5]

In Petherbridge's "Economy of Line," she proposes that drawing approaches, but can not become, painting. The distinction lies in terms of drawing's emphasis on line relative to painting's reliance on matter. The distinction is one of interpretation rather than of medium. A drawing can be made with paint, but it cedes that label as soon as the quality inherent to the paint itself (hue, value, materiality, character of light) dominates over the legibility of line. In it's default state, the drawn line is an outline, registering as the edge of shape. Petherbridge provides ample evidence that lines gain capital by entering into territory that allows the suggestion of painting or an experience similar to that of painting without fully betraying the topology of the line.

In the context of architecture, Petherbridge's "tends to approach" method for framing drawing can be applied to the notion of image. The image is clearly distinct from the drawing–a visible impression, often associated with optics, screens or photography. As with the case of painting, drawn lines can gain value as they approach that which is antithetical to their nature. Piranesi or Durer etchings[6] are classic examples of fine lines collectively crafted to register as tonal variation within an apparently optical field. Positioning drawing relative to image is significant divergence from the reactionary default to compare drawing to modeling. If drawing is "dead," the assumption is often that it has been replaced by a software-driven tendency to digitally model. A re-alignment from a focus on the model to a a focus on image is practical as much as it is philosophical. The image and the drawing are literally and metaphorically in the same plane. An increased use of digital modeling as design medium leads directly to a resurgence of the rendered image, even to the extent of it being termed "the new conventional representation"[7] Even the screen-oriented modeling interfaces are image-bound, which corresponds directly to the relatively recent coinage of the term "screenshot aesthetic"

Images, drawings and paintings all can betray or obscure the structures and methods of their making. The salient characteristic of the drawing, as opposed to image, is that of time. Images tend toward the instant. Drawings can operate with respect to duration. The world of each drawing can contain or operate within multiple temporal levels. The quality of line can vary from start to end, registering time. Serialized lines can register the time of making the drawing as early lines manifest differently from those which are marked later. The drawing itself can convey a broader temporal scope, conveying or suggesting what came before or what will come next.

[5] Ibid.
[6] Durer, Melencolia.
[7] Brothers, "Experience and Fantasy in Renaissance Drawing."

Sometimes this temporal axis aligns with the design process, which can manifest directly within or as distance between drawing. Multiple and iterative versions operating within the same drawing space or drawing series. Temporal value is another component of Petherbridge's linear economy.

A final schema for in this sequence of definition of "drawing" by relative positioning extends to time with respect to function. A drawing that functions predominantly in the moment of drawing, for the purpose of extending or augmenting thinking, may be considered a sketch. At the other end of the spectrum is the drawing that extends or communicates and idea over time. The codified drawing conventions of the traditional "construction set" function as an architectural language and are replete with symbols. This schema is important because it sets the stage for human evaluation and input to be displaced temporally from the act of marking lines on paper.

1.2 Between Computation and Architecture: The Inherent Value of Drawing

The line is inherently architectural. Line is the prerequisite for edge, boundary, shape and form. It can be seen, therefore, as the most primitive element of architecture. Line is also fraught with ambiguity. Sidedness, scale, enclosure, materiality, and figure must be constructed though the ordering and the ordered making of lines. Architects rely on and leverage this linear ambiguity to build meaning and experience, which extend beyond what is materially constructed. Conversely, linear worlds engender confidence that non-representational lines can step into a territory of representation or can define the control of form-making instruments. This reasoning can be extended to define drawing, and it's basic element, the line, as necessary intermediate mode between anything that isn't architecture (idea, emotion) and architecture.

Drawing is the most direct territory in which computational structures can be explored visually. Conversely, although lines can be automatically interpreted as elemental, they require some computational structure for their construction and operation. Computing for drawing is, as a result, territory amenable to controlled exploration of the architectural implications of computation.

This basic and fundamental connection between drawing and computing is obscured by the commercialization of digital design media. Software biases pre-established applications for users. The inevitable result is a relatively inflexible layer of interface and fixed topologies. Likewise, drawing, as defined by the quality and time-laden line, is marginalized by hardware. Digital output and reprographic technology emphasize re-producing images from the screen on to paper. The apparatus associated with a "command-p" culture is closed to manipulation by a designer. As a result, "digital drawing" is a practical, if not conceptual nor technical, contradiction of terms. Mitchell and McCullough, in the canonical Digital Design Media, implicitly foreshadow the conceptual confusion with respect to "drawing" and computation by avoiding the term altogether. The structure of the book identifies "Images," followed by "Drafted Lines" as topological categories.[8] This is not an omission, but a characteristic rigor with respect to accuracy of terminology. Digital lines are more accurately labeled "drafted" despite the colloquial usage of the term "drawing." The implicit conclusion taken from Mitchell and McCullough's structure is that precisely defining or creating shapes with a computer does not conform to any rigorous and historically consistent definition of drawing. Even pixel-level, 2-D manipulation (including, for example, the coloring of pixels with the use of a mouse or other input device), for Mitchell and McCullough, would fall under the category of image creation, not drawing. This project suggests that gap between image and drafted lines is an opening for projective research.

With "computing drawing" established as a non-trivial task, the question of meaning still remains. Computing drawing might be inherently difficult, but what's it's value for design? Andrew Witt discusses a definition of architecture as the combination of design knowledge and instrumental knowledge.[9] He argues that the skilled use of machines and tools (instrumental knowledge) played an essential role in encapsulating knowledge for architects prior to modernism. A relatively recent near- exclusive emphasis on design knowledge (which Witt defines as logic, order, and relationships) relegates technique to the margins. Witt's categorical distinction and advocacy for the fusion of design and instrumental knowledge is timely given the speed at which fabrication and construction tools are being re-considered by architects. Implicitly, drawing is essential for Witt's argument as the territory in which instrumental knowledge can become design knowledge (and vice versa).

1.3 Designing the Apparatus, Computing and Drawing

Given that the contemporary role of drawing in architectural design remains uncertain and that drawings offer a unique opportunity to frame computation and architecture on mutually

[8] Mitchell and McCullough, Digital Design Media.
[9] Witt, "A Machine Epistemology in Architecture. Encapsulated Knowledge and the Instrumentation of Design."

respective terms, the aim of this project is to document the design, use, and result of a computational drawing apparatus. This apparatus spans the full territory of drawing from abstract representation of lines to their material manifestation in ink on paper. The goal is to document, demonstrate and discuss the implications of computing drawing.

Lines are defined computationally in the Python programming language. A suite of object-defining scripts was written which describe structure and behavior. The Python programs also control salvaged Hewlett Packard Pen Plotters[10] via the generation of commands in the machine language, HPGL.[11,12]The Chiplotle Python Library facilitates serial communication with the plotters and automates the formatting of HPGL commands.[13]

The structure, meaning, and complexity of the Python programs are presented and discussed in section two of this document in greater detail. The functions deployed to construct drawing programs are basic to the extent of being primitive, directly commanding the plotter to, for example, grab a pen from the carousel, move the pen to a position on the paper, lower the pen so that it makes contact with paper, move the pen to a new position, (leaving a mark) lift the pen back off the paper, and then repeat this process with a slightly offset position, producing a set of parallel lines. The more elaborate drawing programs define line types embedded with hierarchies (lines defined by sets of lines or series of lines), parameters (speed, length, color, direction, etc.) typological descriptions, (intersection points or curve degree, for example) goals, and probabilities. The least direct drawing programs operate with non- linear algorithms, creating additional computational structures (particles, lattices, grids and fields) that contain or are associated with relational behaviors and goals, and are translated into marked lines indirectly based on values, qualities and conditions gleaned from these computational structures.

HPGL, the machine language understood by the plotters, is simple to the extent that it can be written and read directly by a human with relatively little instruction. All HLGL commands are two-character codes followed by parameters, which are separated by commas. The conclusion of a command is marked by a semicolon. HPGL ignores spaces, line breaks or any "whitespace." The most common HPGL commands (and the exhaustive list of those used in this project) are:

SP (select pen)

[10] An HP DraftPro DXL (model HP7575A) and an HP DraftPro EXL (Model HP7576A)

[11] "HP-GL Reference Guide."

[12] Hewlett Packard Graphics Language, HP's attempt to proliferate an industry standard, output language. It preceded the more advanced HPGL/2, which lost prominence to the completing Postscript language.

[13] Adán and Repetto, "Chiplotle API."

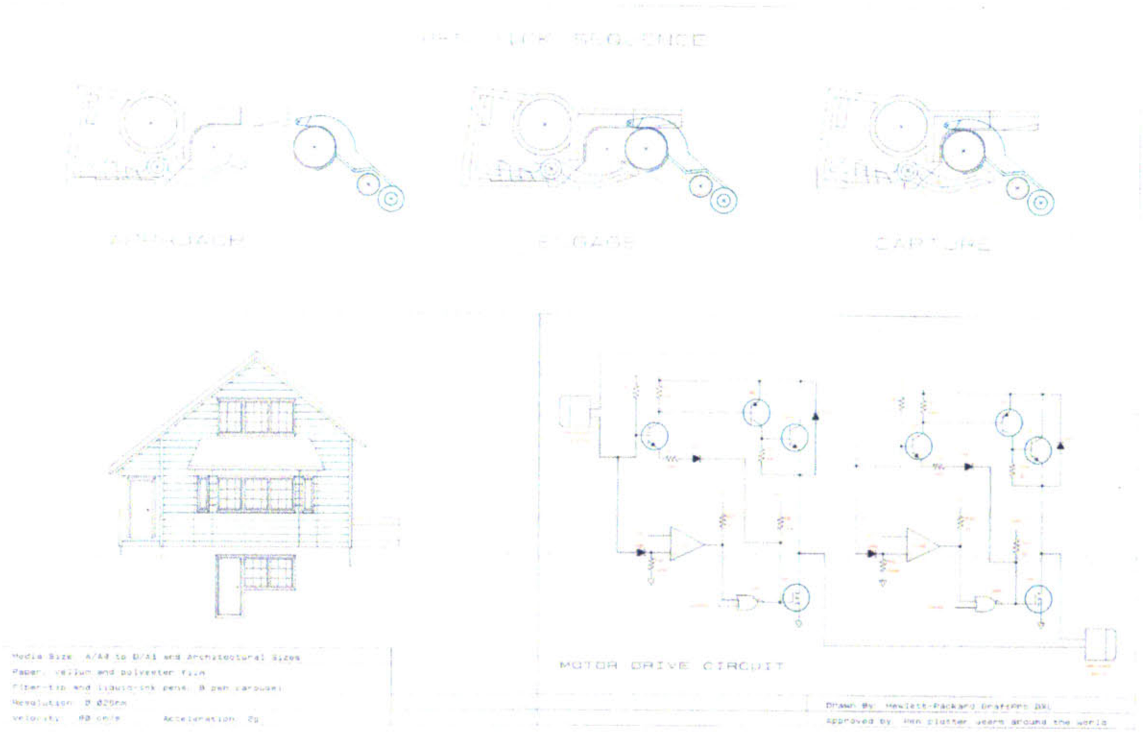
PU (move to coordinates with the pen in the "up" position, this making no contact with the paper)

PD (move to coordinates with the pen in the "down" position, leaving a mark on the paper)

The following code would instruct the pen to draw a horizontal line of length 1000 units from the coordinate position (200,200) on paper:

```
SP1;PU200,200;PD1200,200;
```

Two nearly-identical model HP plotters were procured for this project. Manufactured between 1988 and 1990, these devices were marketed for engineers and architectural draftsmen with the intent of automating the process of drafting precise technical information on paper. Each plotter contains 512 KB or RAM and a small amount of ROM to store a "demonstration plot." The content of this demonstration plot tellingly reveals the vocational bias of these devices. Although expensive, these plotters, which used up to eight disposable pens for each drawing, developed a


APPROACH
ENGAGE
CAPTURE
Media Size: A/A4 to D/A1 and Architectural Sizes
Paper: vellum and polyester film
Fiber-tip and liquid-ink pens: 8 pen carousel
Resolution: 0.025mm
Velocity: 60 cm/s
Acceleration: 2g
MOTOR DRIVE CIRCUIT
Drawn By: Hewlett-Packard DraftPro DXL
Approved by: Pen plotter users around the world

The scanned output of an HP7575A DraftPro DXL Demonstration Plot reputation for inconsistency and slowness.[14] The tip of the pen degraded markedly before running out of ink and the quality of line was highly dependent on the bleed of the ink, which varied by color, manufacturer, paper type and even humidity. The resolution of the drawing is technically limited by the x and y axis motors. On each axis the pen movement is controllable to about one thousandth of one inch. These plotters are sheet-loaded. Only one drawing can be processed or "sent" to the plotter at a time. As part of an initialization procedure, the sheet is measured and somewhat straightened by the plotter. A Serial-USB adapter bridges contemporary and vintage ports.

1.4 Advantageous Limitations of Vintage Plotters

The limitations of the pen plotters led to their commercial obsolescence. The same limitations, however, allow the kind of direct exploration between computation and drawing that motivate this project. Because the plotters operate without drivers of any kind; they must be controlled directly with the Python language. There is no software nor screen intermediary. The entire process is completely transparent and controllable from algorithm to ink. This is conceptually and methodologically important. The drawings and the drawing process are transparent, controllable and documentable.

The pen plotters can not move in curvilinear paths. The lines drawn can be curved, but are inevitably made of straight segments, often of lengths that are imperceptibly small. This is a result, in part, of the internal mechanics of the plotters–motors for each axis have multiple but discrete speeds. More importantly, the point is the only geometric unit in HPGL. In being technically accurate in the description of the marks made by the pen plotter, it becomes important to note that HPGL contains no explicit structure for defining line. Each HPGL command is an instruction to "move to" a point in the coordinate space of the paper. It is up to the computational structure to define lines (even if such a structure is as simple as a start and end point) and translate this information to plotter instructions for movement. This apparent limitation enforces a scalar component to any representation of line.

Pen plotters are equipped with a carousels that store multiple pens. The pen plotters used in this project can hold up to eight pens at a time. During the course of a drawing, the plotter can use any one of the pens at a time for drawing. The key limitation here is that pens are discrete.

[14] Mitchell and McCullough, Digital Design Media, 129.

Their color and thickness is unknown to the device. Furthermore, their ink is not designed for mixing (as is the case with cyan, magenta, yellow and black inks that are the standard for color printing). These plotters were designed to make eight kinds of lines with eight different pens. Any in-between or hybrid conditions must be constructed out of relationships between lines.

Another limitation, bleeding of ink into the paper, is the most significant asset. Because the effective area of the pen head is greater than the smallest increment of motor movement, material rather than geometric resolution is the restricting factor in the legibility of fine detail. Lines can be drawn closer than together than is visibly discernible given the flow of ink cross the paper. As with drawing "by hand," this condition invites a consideration of the paper space as effectively continuous. This treatment of paper is unlike a digital image–or digital print–which is organized into discrete picture elements (pixels).

Considered collectively, these limitations-turned-assets contribute to a tendency to construct drawings in an iterative manner. The computation is linked to the drawing and the drawing to the computation. It is impossible to consider any portion of the apparatus in isolation. In this project, computing and drawing are linked.

1.5 Related Work

A number of contemporary works explore the construction and use of apparatuses to create material drawings. In an effort to position this project relative to related work in a manner that advances the discourse, these related works have been organized into three categories: chaotic drawing machines, improper drawing apparatuses, and explorations into aesthetics of computation.

Chaotic drawing machines use at least three interconnected parameters to create an unpredictable system. Usually autonomous, these machines often produce a line quality similar to thatwhich can be achieved through drawing by hand. Lorenzo Bravi's "Drawbot,"[15] "Pendulum Drawing,"[16] Benjamin Grosser's "The Robot Artist"[17], Robert Howsare's "Drawing Apparatus"[18] are successful examples of this type. Because these machines operate entirely with physical mechanisms, pressure, speed, and position are not subject to the kind of discretization inevitable

[15] Bravi, "Drawbot Children Workshop."

[16] "Pendulum Drawing on Vimeo."

[17] "The Robot Artist | Www.portable.tv | Fashion, Film, Music, Art, Design, Online Video."

[18] "Drawing Apparatus by Robert Howsare : The Method Case."

with digitally-controlled devices. As a result, changes in line quality and character can be gradual. The chaotic system itself tends toward the production of geometry that is ordered, but free from any kind of apparent structures. This type of apparatus is conceptually relevant to this project in that the drawn line has no representation outside of the paper. The interconnected mechanisms resolve into drawing action rather than drawable information. This approach also emphasizes the act of drawing as interconnected with the drawing output. These apparatuses are often themselves exhibited because their operation is a spectacle and their physicality appreciated as kinetic sculpture. The limitations of these apparatuses becomes apparent when they are considered in the context of design. Process and product are essential to understanding the meaning of the drawings. To be used in a design process, however, the resultant drawings must be divorced from their making and interpreted out of context. This should not be taken as critique of these apparatuses, as none of them are framed with respect to design. Instead, they function as artistic commentary questioning our assumptions regarding authorship, cognition and creativity. They rely on the distance of human maker from the act of drawing itself. A human designer can interact with the drawing objects but not with the drawing process.

A significant body of work relating to drawing apparatuses falls into the category of "Improper Drawing Apparatuses," a term used extensively by Marco Frascari.[19] These apparatuses, including those by Tristan Perich,[20] the "Drawbot: robot designer"[21], and ArchiDNA[22] can be placed into this category. This research places an emphasis on a human user, who functions in various roles ranging from arbitrary inputer to alleged design partner. Particularly relevant work investigates the misuse of a pen plotter to make "wobbly" drawings[23]. The drawings that were improperly plotted were found to be accepted by digital "skeptics" despite the increased indirection and presence of the apparatus simply because they appeared similar to freehand sketches. Even in the cases in which the user is framed as a designer–the ArchiDNA project, for example–the operation of the apparatus itself lacks transparency or operability. If those operations are designed, they are designed by another who has no role in evaluating or interpreting the resulting drawings. The user will indirectly generate unexpected results, which may be serendipitous. However, it is difficult if not impossible for emergent the qualities that result from the imperfections to contribute to or offer design knowledge.

[19] Frascari, Eleven Exercises in the Art of Architectural Drawing.10

[20] "Tristan Perich - Machine Drawings."

[21] "Me Salió así: Drawbot: Robot Dibujante."

[22] Kwon, Gross, and Yi-Luen Do, "ArchiDNA: An Interactive System for Creating 2D and 3D Conceptual Drawings in Architectural Design."

[23] van Bakergen and Obata, "Free-hand Plotting."

An equally relevant research trajectory indexes and evaluates computation with respect to drawing aesthetics. John Maeda's pedagogical and conceptual DBN language and it's documentation[24] structures the most fundamental programming operations–loops and conditional statements, for example–in terms of the images that can be produced. Design By Numbers does address drawing metaphorically, but conveys the primacy of the pixel and therefore implicitly concedes that the result is image. His consistent use of the minimal 8x8 pixel grid for most of the documentation emphasizes the tendency for programmed computation to manifest in the appearance of line at the expense of maintained linear structures. Maeda's contribution goes far beyond Design By Numbers, and his cultural as well as technical influence in hybridizing design and computation can't be understated. While Maeda's work expands well beyond the domain of the drawing but aligns perfectly with the motive force of much of architectural drawing. In a statement that introduces some student work he writes, "The work that follows is a collection of projects by my students that attempts to illustrate the invisible realm of programmatic space. In some cases, they enter the temporal dimension to maximize the opportunity for expression; in other cases, they invade the third dimension of depth. The word I use to describe the invisible pockets of cyberspace that leak into our visible space is 'sound.' Not sounds as heard by the ear, of course, but sound within the confines of the mind."[25] One works that Maeda is introducing, Computer Graphics for Print Media[26], uses a large-format printer to overcome the resolution limitations of screens by directly creating a print output of a graphical representation of the human genome.

Kostas Terzidis also presents the direct implications of program structures[27], but avoids any discussion of the drawing, and in some cases, the image altogether, by relying on the arrangement of three-dimensional forms, even when demonstrating operations in one or two-dimensional space.

This project diverges from prior research in computational aesthetics with respect to the context of architectural design. For architects, the burdens placed on the drawing are different from the direct implications that computational aesthetics may have for graphic design. Still, the contributions of the Media Lab's Aesthetics and Computation Group, which was led by John Maeda, are significant and essential conceptual buildings blocks for this work.

[24] Maeda, Design By Numbers.
[25] Maeda, Creative Code, 20.
[26] Benjamin Fry, Computer Graphics for Print Media.
[27] Terzidis, Algorithmic Architecture. 76

This project extents upon the drawing apparatuses mentioned above as it spans from computation to resulting drawing. While some of the characteristics and qualities of the resulting drawings align with those found in the chaotic and improper apparatuses, this work aims to control, diagram, structure and evaluate the forces that lead to desired drawing qualities.

References

Adán, Víctor, and Douglas Repetto. "Chiplotle API", n.d. http://music.columbia.edu/cmc/chiplotle/.

van Bakergen, W. D., and G. Obata. "Free-hand Plotting: Is It Live or Is It Digital?" CAD Futures 91 (1991): 567–582.

Benjamin Fry. Computer Graphics for Print Media, 2002. Creative Code, Page 21.

Bravi, Lorenzo. "Drawbot Children Workshop", n.d.

Brothers, Cammy. "Experience and Fantasy in Renaissance Drawing" presented at the J. Irwin Miller Symposium, Yale School of Architecture, February 10, 2012.

Cohen, Preston Scott. "Plan Vs. Drawing" presented at the J. Irwin Miller Symposium, Yale School of Architecture, February 11, 2012.

"Drawing Apparatus by Robert Howsare : The Method Case", n.d. http://www.themethodcase.com/drawing-apparatus-by-robert-howsare/.

Durer, Albrecht. Melencolia, n.d.

Frascari, Marco. Eleven Exercises in the Art of Architectural Drawing: Slow Food for the Architect's Imagination. 1st ed. Routledge, 2011.

"HP-GL Reference Guide", n.d. http://www.isoplotec.co.jp/HPGL/eHPGL.htm.

Kwon, Doo Young, Mark D. Gross, and Ellen Yi-Luen Do. "ArchiDNA: An Interactive System for Creating 2D and 3D Conceptual Drawings in Architectural Design." Computer-Aided Design 41, no. 3 (March 2009): 159–172. http://www.sciencedirect.com/science/article/B6TYR-4T5JJ03-2/2/8ab193dce1229100108f7d4ae7c2db55.

Maeda, John. Creative Code: Aesthetics + Computation. Thames & Hudson, 2004.

———. Design By Numbers. The MIT Press, 2001.

"Me Salió así: Drawbot: Robot Dibujante", n.d. http://mesalioasi.blogspot.com/2012/01/drawbot-robot-dibujante.html.

Mitchell, William J., and Malcolm McCullough. Digital Design Media. 2nd ed. Van Nostrand Reinhold, 1995.

"Pendulum Drawing on Vimeo", n.d. http://vimeo.com/34720820.

Petherbridge, Deanna. The Primacy of Drawing: Histories and Theories of Practice. Yale University Press, 2010.

Terzidis, Kostas. Algorithmic Architecture. 1st ed. Architectural Press, 2006.

"The Robot Artist | Www.portable.tv | Fashion, Film, Music, Art, Design, Online Video", n.d. http://portable.tv/tech/post/the-robot-artist/.

"Tristan Perich - Machine Drawings", n.d. http://www.tristanperich.com/Art/Machine_Drawings/.

Witt, Andrew. "A Machine Epistemology in Architecture. Encapsulated Knowledge and the Instrumentation of Design." Candide, no. 3 (December 2010): 37–87.

2. Drawing

This section catalogs drawing process and product. Drawings are organized by method, series, and run using the following syntax: Method-Series-Run. A "method" is an algorithmic approach to controlling the pen plotter and is the most general way to organize these drawings. Within each method, a "series" refers to a specific python code and/or plotter configuration. A run refers to one drawing within the series. Whether the drawing is re-plotted or generates a series of drawings, the run identifier keeps track of their production order over time. For example, drawing A-002-001 is the first run of series 2,in the method 'A' set.

Methods are organized in sequence. Their order is associated with the extent to which they are layered with representation and operation. Early methods are made up of fewer layers, making the process of crafting drawing legible without explanatory diagrams. Series in later methods use more stages of representation and algorithmic steps, and require diagrams to elucidate the working apparatus. At the beginning of each series, when necessary, diagrams explain the computational representation of line that have been developed and deployed. Diagrams take two distinct forms: vector diagrams and pixel diagrams. Vector diagrams demonstrate the geometric relationships between lines. Pixel diagrams illustrate linear information as image. Vector diagrams make every layer of representation and action explicit. Pixel diagrams illustrate (often in at an exaggerated scale) the visual implications of structures that might not manifest directly in drawn lines. Vector drawings use the following consistent notations for lines and points.

□	an animate or temporary point
■	a fixed point
⋱	a computed (but not drawn) line
⇘	a computed (but not drawn) line with direction
╲	a drawn line

Method Page Organization

Method X

This text area contains a nattative description of this method.

Series X-001

Scaled versions of each drawing in this mehtod are organized by series. Here, an abridged verion of the series description is written.

1

2

Method-Series-Run

Method-Series-Run

3

4

Method-Series-Run

Method-Series-Run

Series X-002

At the start of the next series in this method, a label followed by this description mark the transition.

Method Page Organization

Series X-003

At the start of the next series in this method, a label followed by this description mark the transition.

5

6

7

Method-Series-Run

Method-Series-Run

Method-Series-Run

8

9

10

Method-Series-Run

Method-Series-Run

Method-Series-Run

Series Page Organization

Series X-001

This text is a description of operations and actions that make up this series. It makes reference to vector drawings.

Vector Diagram 1

X-001-VectorDiagram-01

Vector Diagram 2

X-001-VectorDiagram-02

Vector Diagram 3

X-001-VectorDiagram-03

Vector Diagram 4

X-001-VectorDiagram-04

Series Page Organization

Pixel Diagram 1

X-001-PixelDiagram-01

Pixel Diagram 2

X-001-PixelDiagram-02

Pixel Diagram 3

X-001-PixelDiagram-03

Pixel Diagram 4

X-001-PixelDiagram-04

The operations and actions of this series are further explored with pixel diagrams at left.

Cropped Full Scale Drawing

Run Page Organization

Method-Series-Run: Title of Drawing

This text is a narative description of this drawing

Run Page Organization

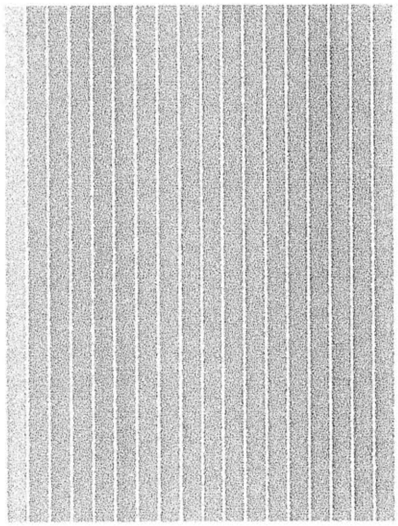
Drawing Scaled to Fit Page

Method A

The space of the paper is a catalog of variables.

Series A-001

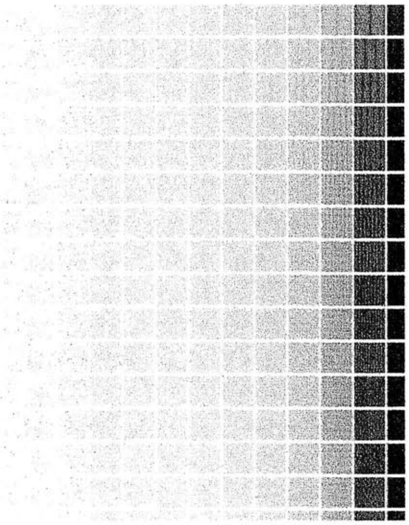
The same line is repeatedly drawn in columns beginning at the top left.



A-001-001

Series A-002

The drawing is a matrix. Vertically, each square swatch contains lines drawn with increasing numbers of stops along the paths.

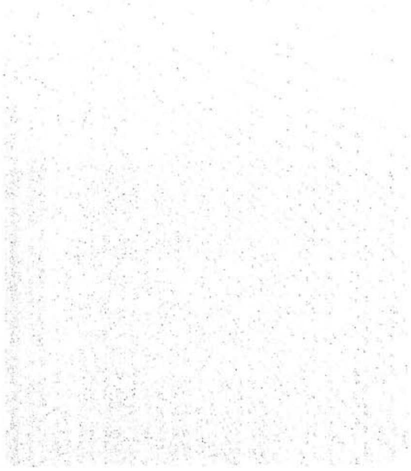


A-002-001

A-003-001

Series A-003

Each drawing is a matrix. Vertically, each of the lines that make up a swatch are increasingly splayed around a middle pivot. Horizontally, the number of lines in each swatch increases.



A-004-001

Series A-004

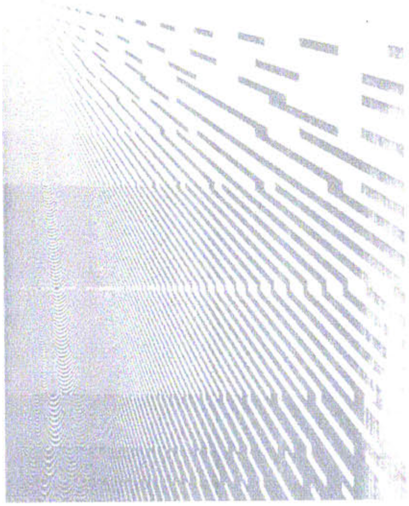
Each vertical line is divided into equal length portions. A dashed line is drawn over that portion. Vertical position on the page affects length of the portion. Horizontal position affects the dash-gap ratio.

Series A-005

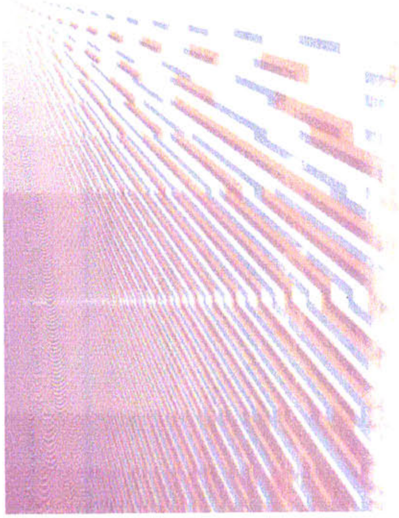
Two length/dash-gap matrices are drawn

Series A-006

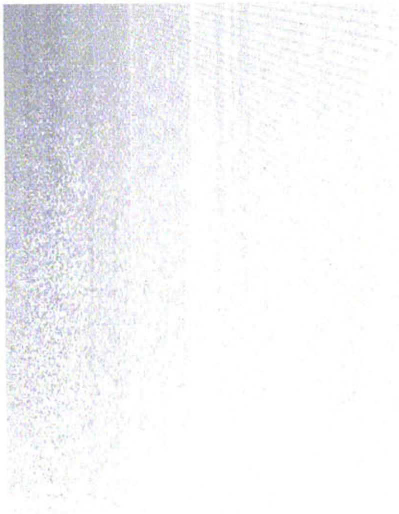
Vertical position affects the probability of each line segments existence. Horizontal position effects the density of segment divisions in each line.



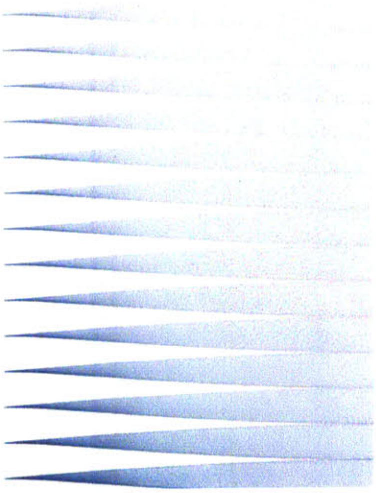
A-004-002



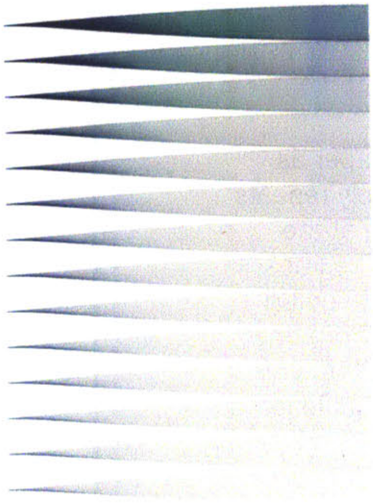
A-005-001



A-006-001



A-007-001



A-007-002

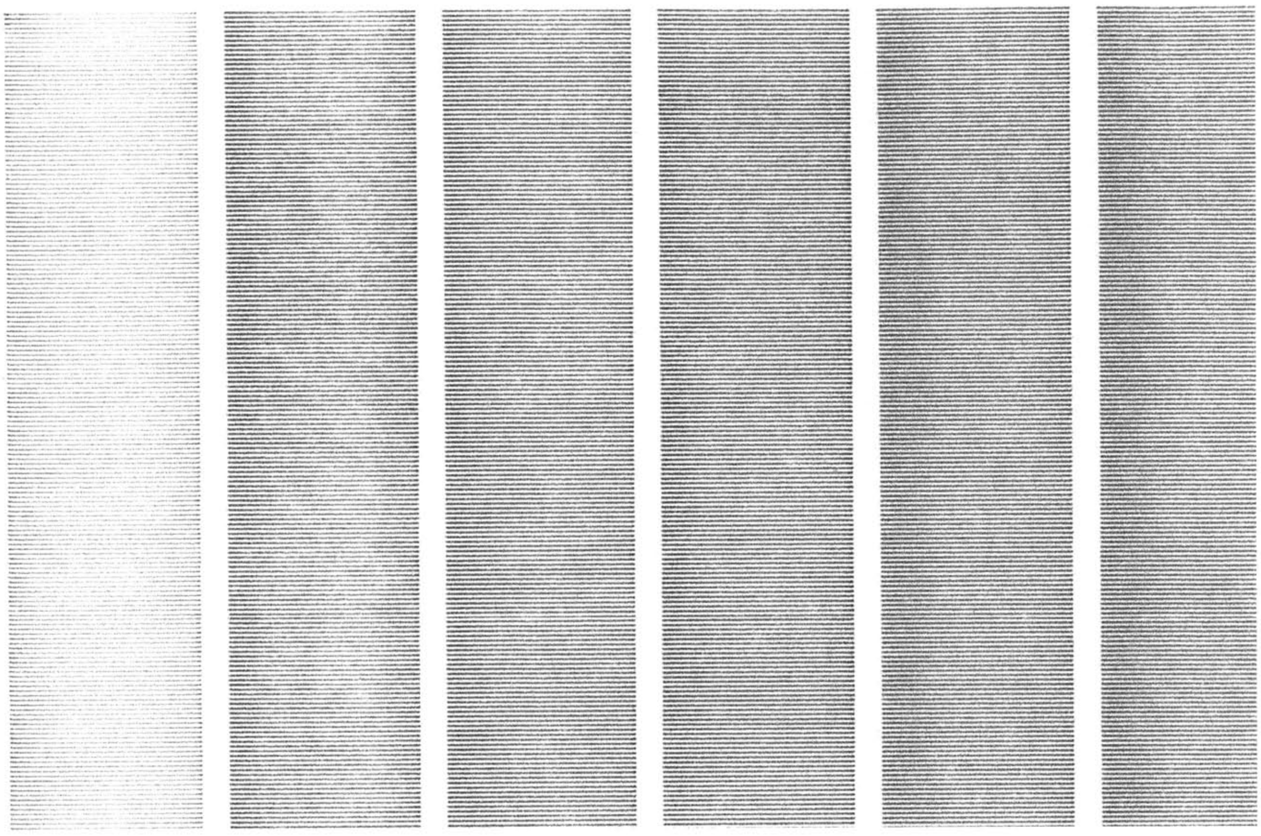
A-008-001

Series A-007

Each drawing is a matrix. Line direction is affected by horizontal position. Line density is affected by vertical position

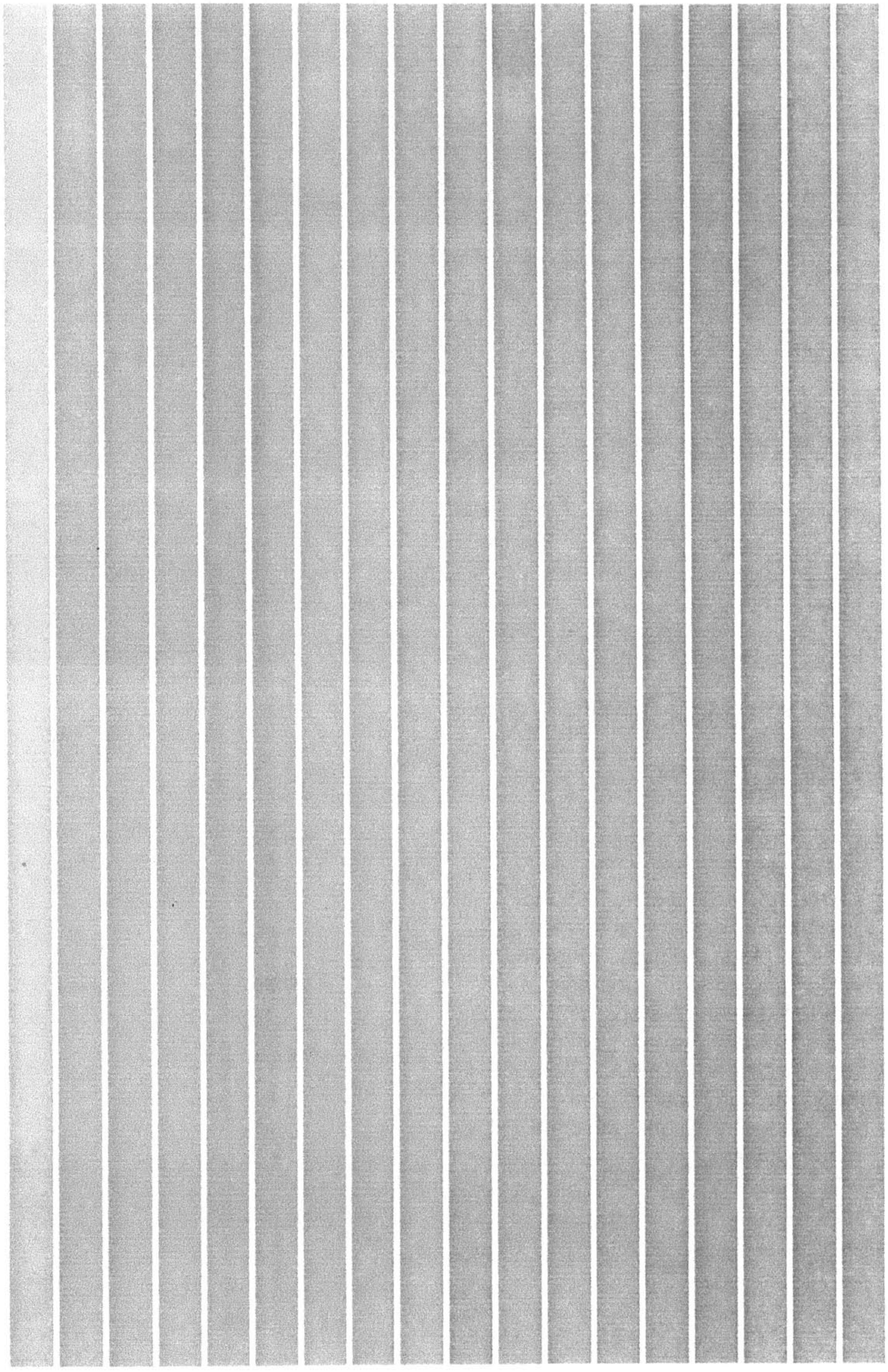
Series A-008

Horizontal lines are drawn in segments. Lower lines are drawn with more segments. Lower segment positions are subject to increased random variation in the vertical axis.

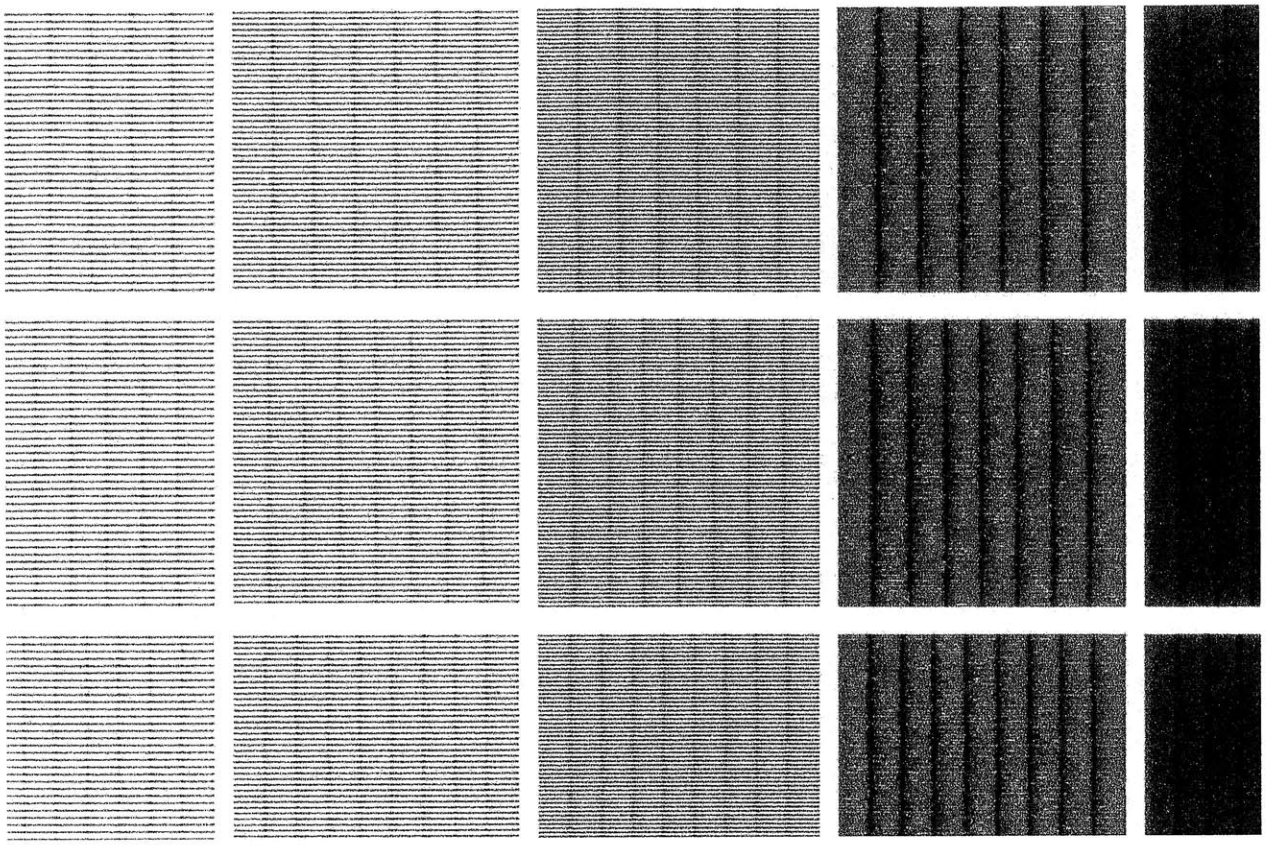


A-001-001: Lines Over Time

Method 'A' treats the space of the paper as catalog of variables. Series 001 simply repeats the same line in columns beginning at the top left. The spacing is consistent. The lines are as close as possible before the bleeding ink renders them indistinguishable from one another. Time is the only variable as the wear of the felt tip on the plotter and the depletion of ink affect the lines.

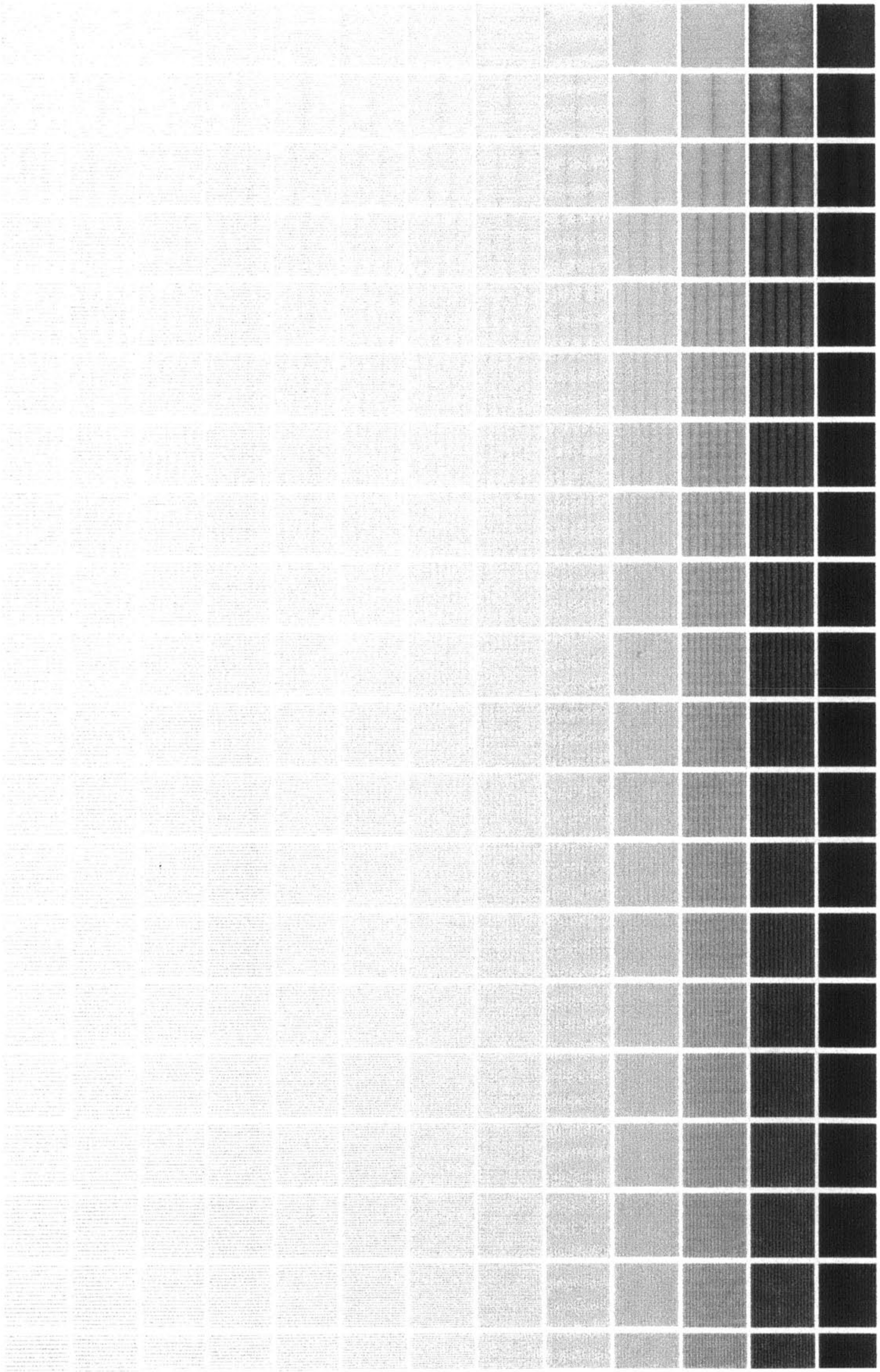


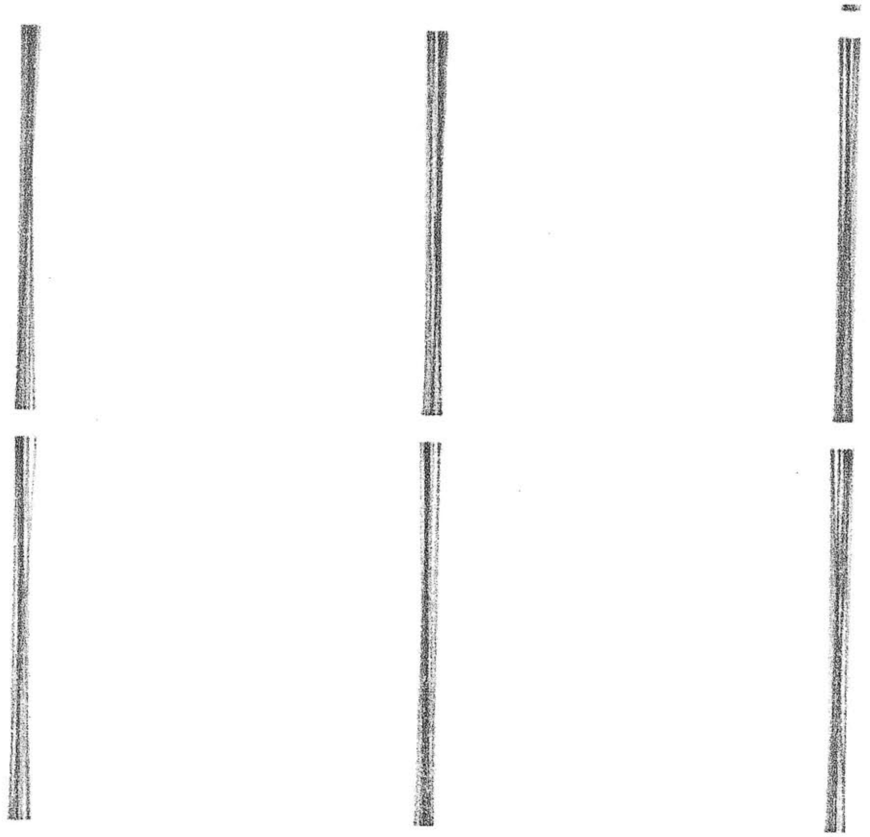
A-001-001



A-002-001: Density and Stoppage

Method 'A' treats the space of the paper as catalog of variables. In series 002, each drawing is a matrix. Vertically, each square swatch contains lines drawn with increasing numbers of stops along the paths. This leads to pools of ink building up at the stop point. Lines at the top are drawn with no stops. Lines at the bottom are drawn with 20 stops. Horizontally, each swatch contains an increasing number of lines, always regularly spaced. Both the variables in the vertical and horizontal dimensions vary linearly.

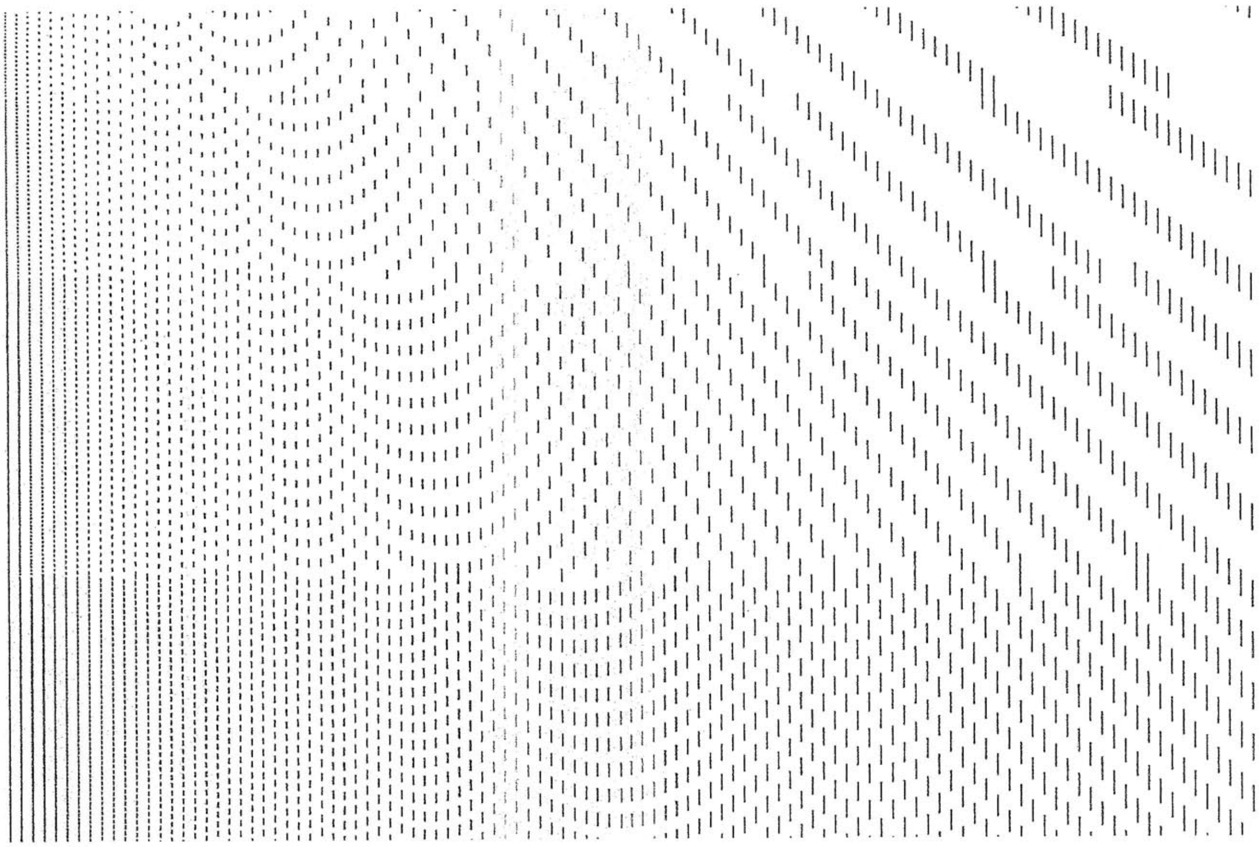




A-003-001: Lines of Lines (of Lines)

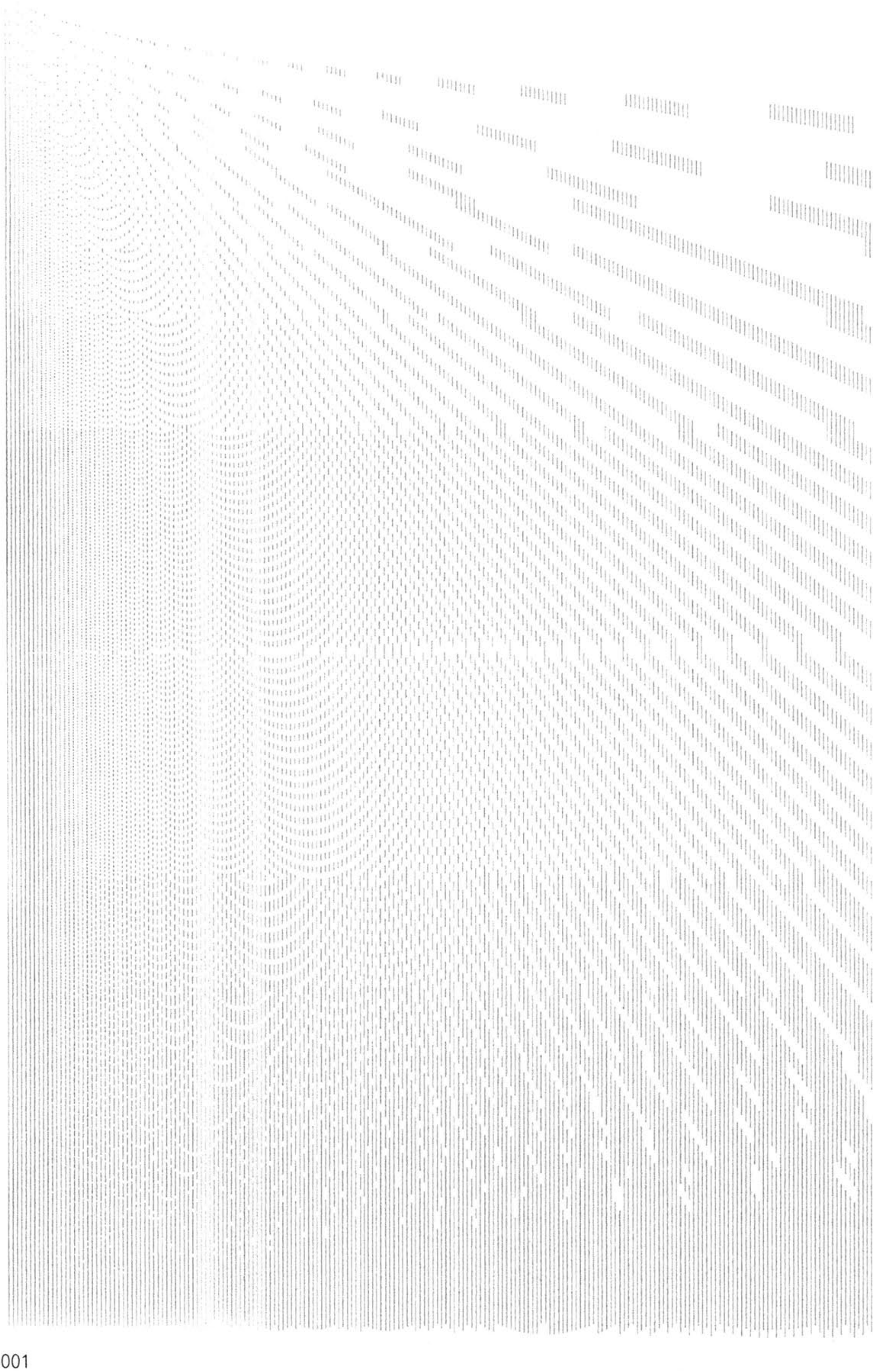
Method 'A' treats the space of the paper as catalog of variables. In series 003, each drawing is a matrix. Vertically, each of the lines that make up a swatch are increasingly splayed around a middle pivot. Horizontally, the number of lines in each swatch increases. Both the horizontal and vertical variation is linear.

A-003-001

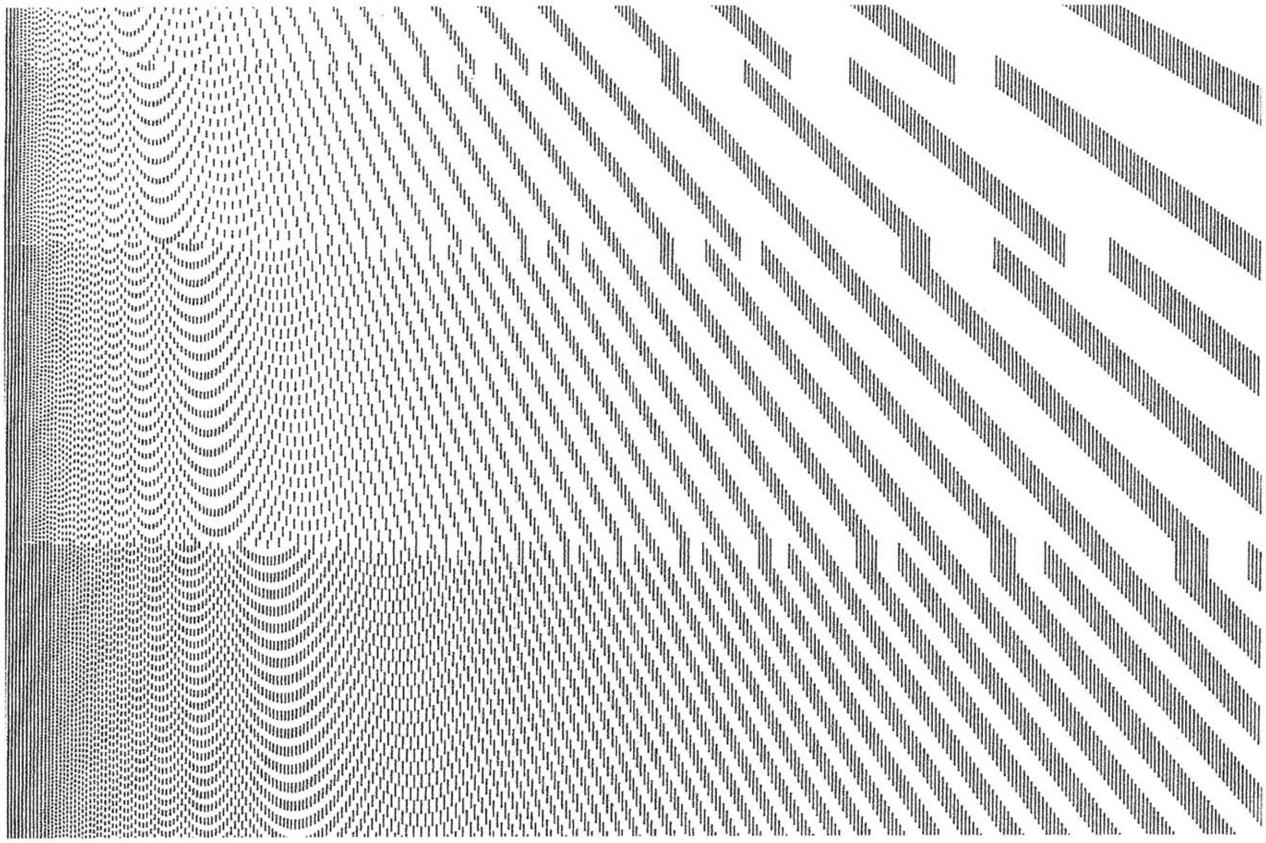


A-004-001: Dash Size and Ratio

Method 'A' treats the space of the paper as catalog of variables. In series 004, each vertical line is divided into equal length portions. A dashed line is drawn over that portion. Depending on the vertical position, each portion is assigned a ratio of dash length to gap length. Portions near the bottom of the page have higher ratios than those nearer to the top. The relative size of each dash is related to the portions position on the page from left to right. Lines on the left are made with short segments, while the lines towards the right are longer.

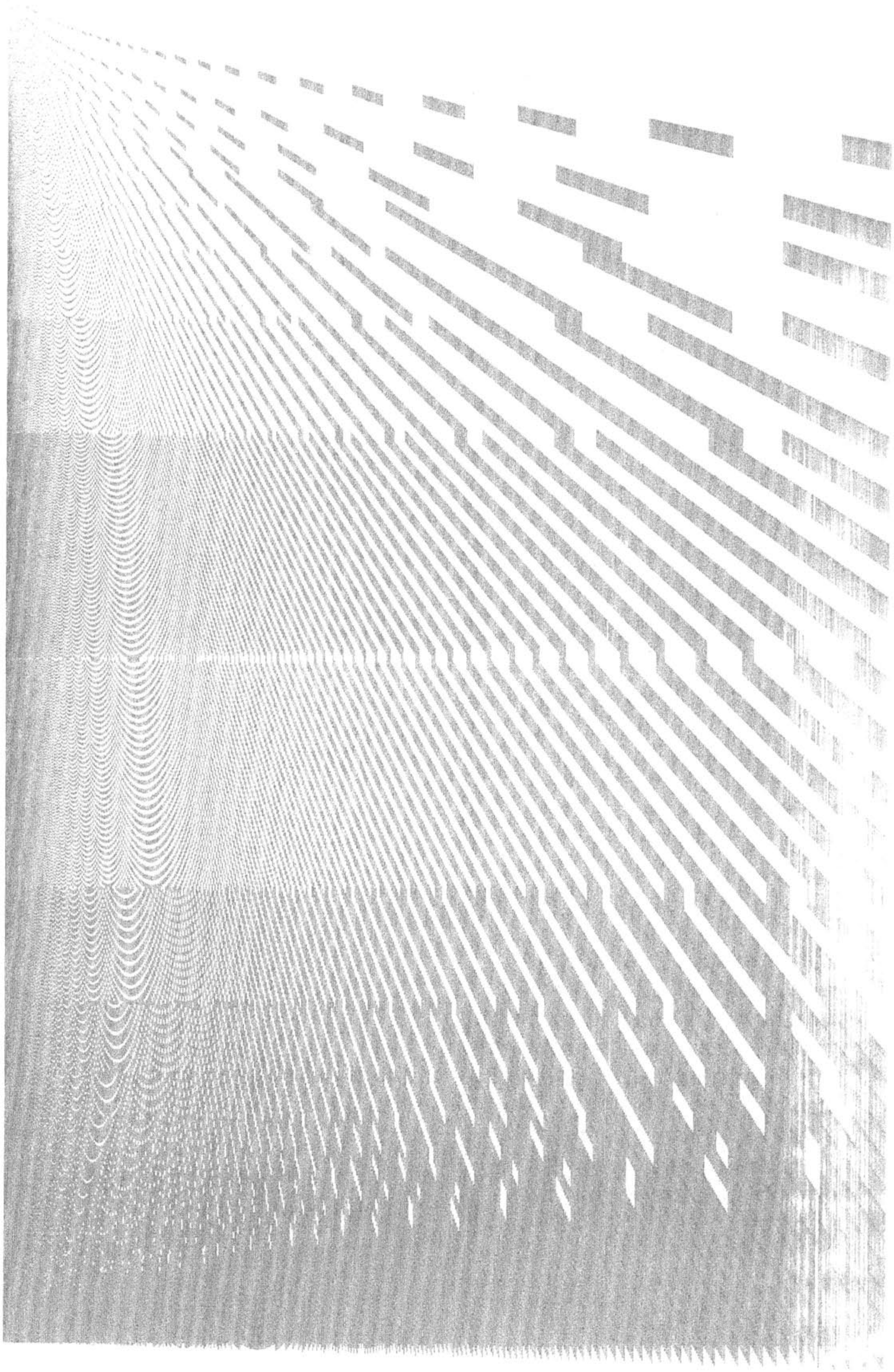


A-004-001

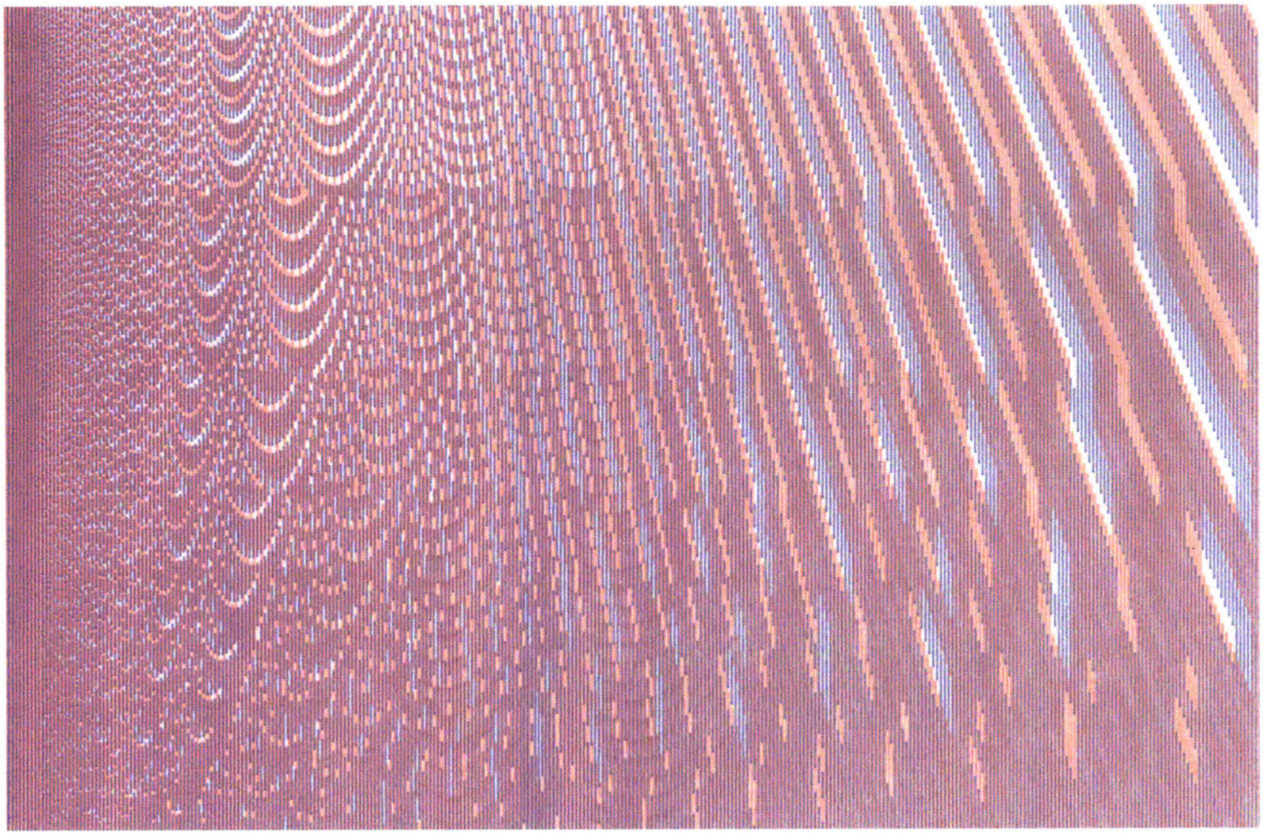


A-004-002: Dash Size and Ratio– Closely Aggregated

Method 'A' treats the space of the paper as catalog of variables. In series 004, each vertical line is divided into equal length portions. A dashed line is drawn over that portion. Depending on the vertical position, each portion is assigned a ratio of dash length to gap length. Portions near the bottom of the page have higher ratios than those nearer to the top. The relative size of each dash is related to the portion's position on the page from left to right. Lines on the left are made with short segments, while the lines towards the right are longer. In this drawing, the second run, more lines in total are drawn. This increases the portion of the drawing in which the line segments are significantly closer to their left-right neighbor than the next or previous vertical segment.

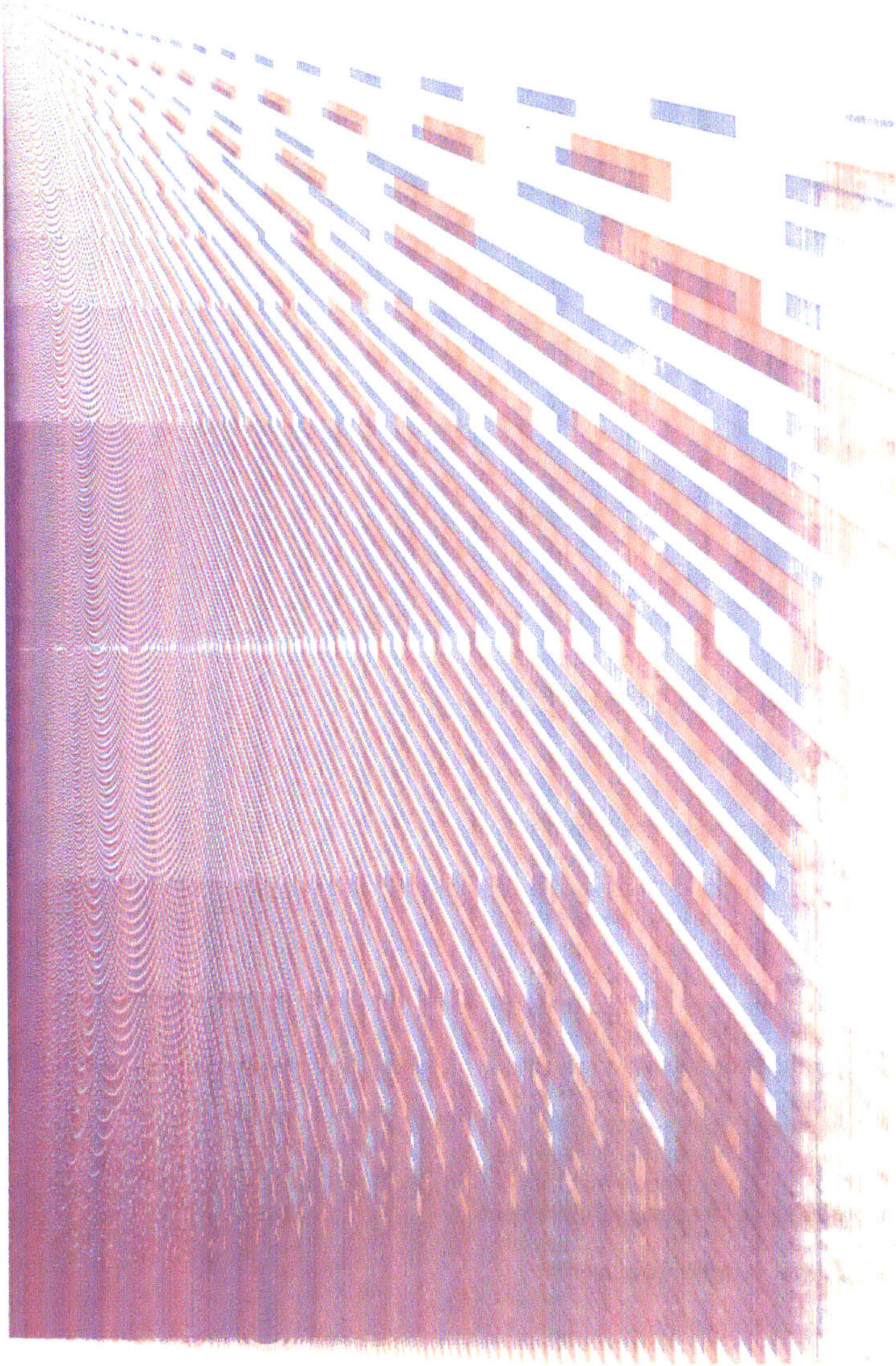


A-004-002

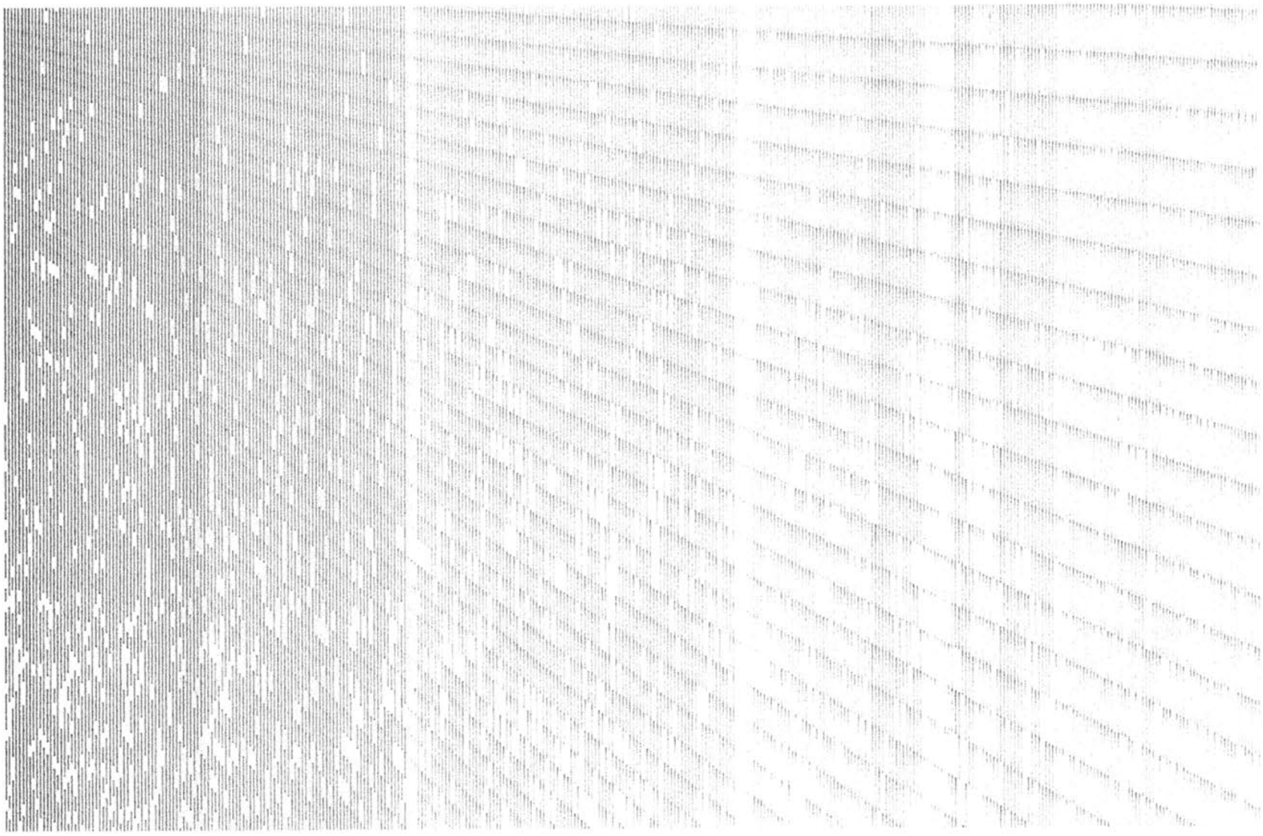


A-005-001: Two Overlapping Dash Size and Ratio Matrices

Method 'A' treats the space of the paper as catalog of variables. Series 005 is a spin-off of series 004, where the vertical dash to gap ratios are related to the vertical axis and the segment size is related to the horizontal axis. Series 005 overlays two such matrices. In this run the two sets are drawn with red and blue pens with varied densities. Furthermore, the red set's vertical and horizontal axes are scaled relative to those of the blue set.

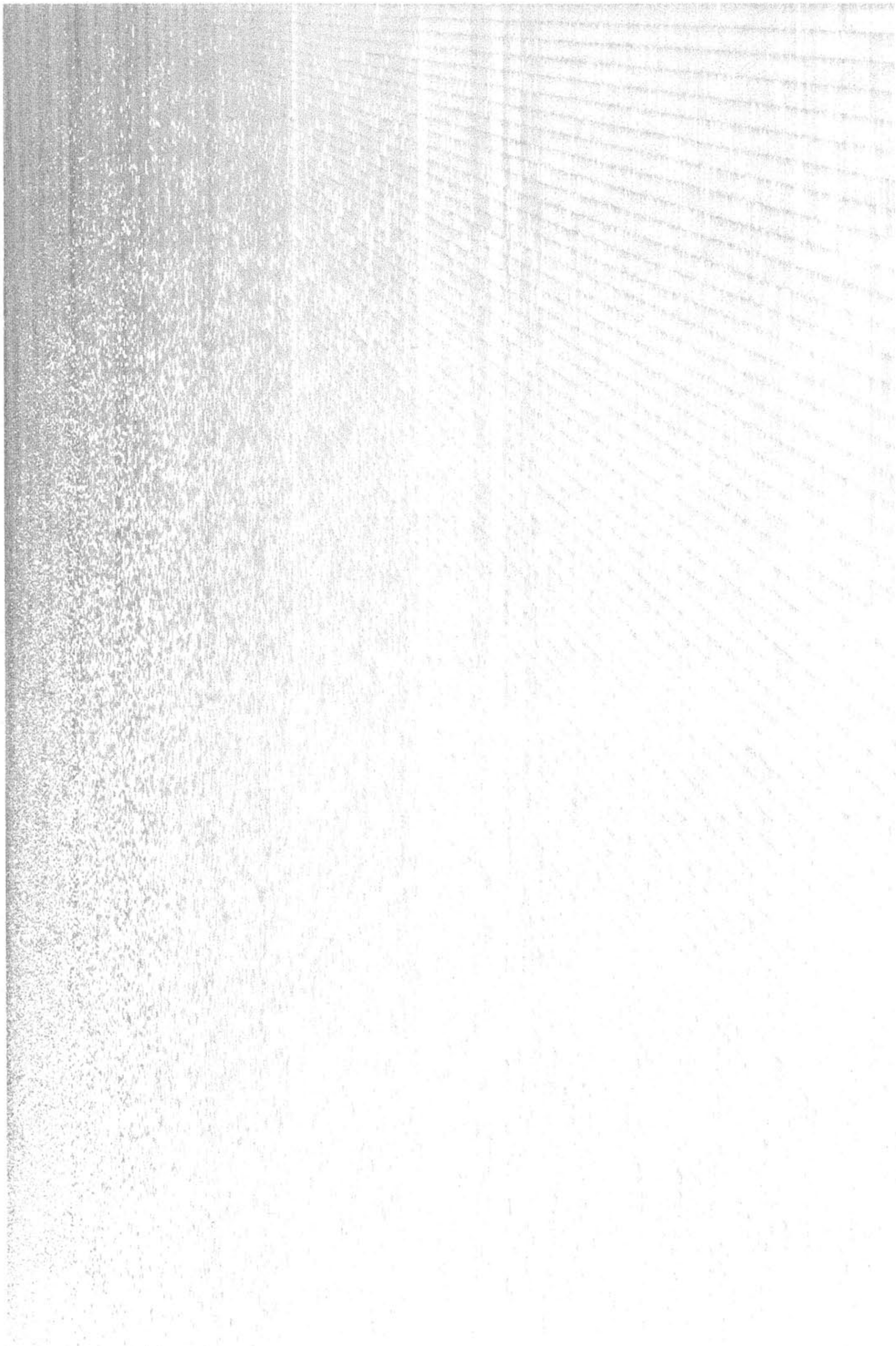


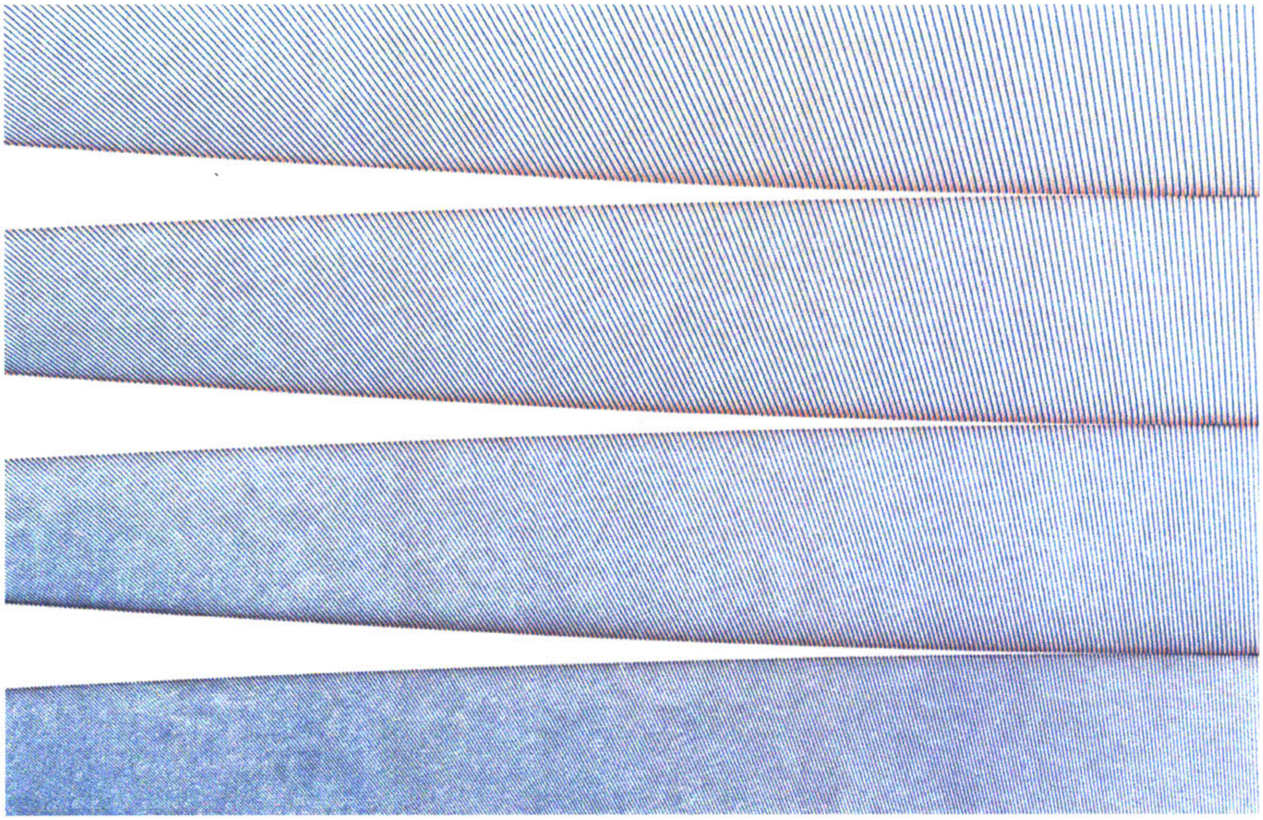
A-005-001



A-006-001: Gradient Dashes

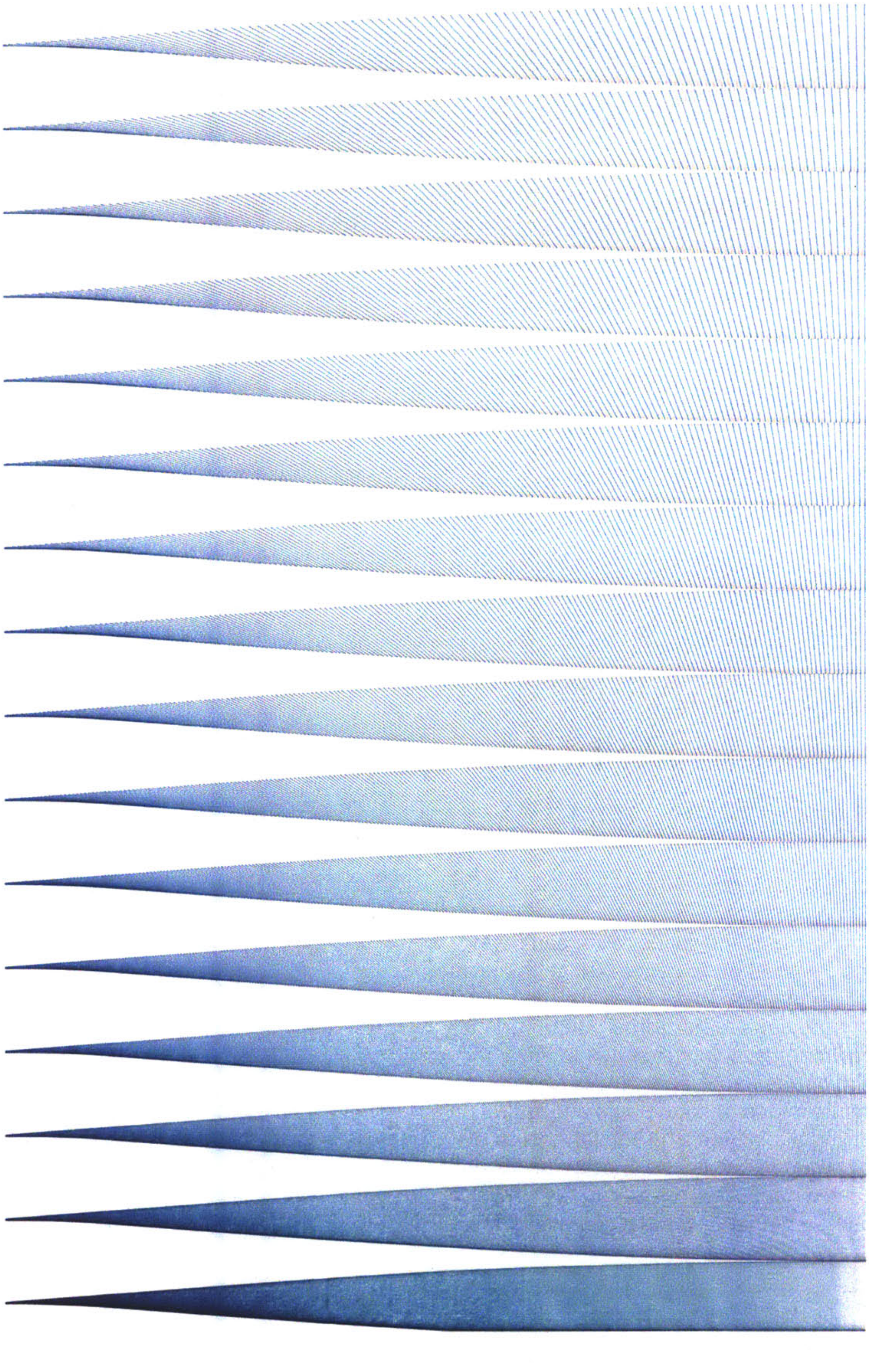
Method 'A' treats the space of the paper as catalog of variables. In series 006, each vertical line is drawn in segments. Unlike in series 004 and 005, these whole-page verticals are not sub-divided into portions. Instead, beginning at the top, a vertical segment is measured. Whether this computational line becomes a drawn line depends on the result of a random value. The algorithm works its way down the page in even increments. Unlike in series 004 and 005, there are no dash-gap patterns. Instead, as the calculated segments near the bottom of the page, the likelihood that they are drawn gradually decreases. The increment for each line varies with the line's horizontal position. This run was drawn with a heavily worn pen.



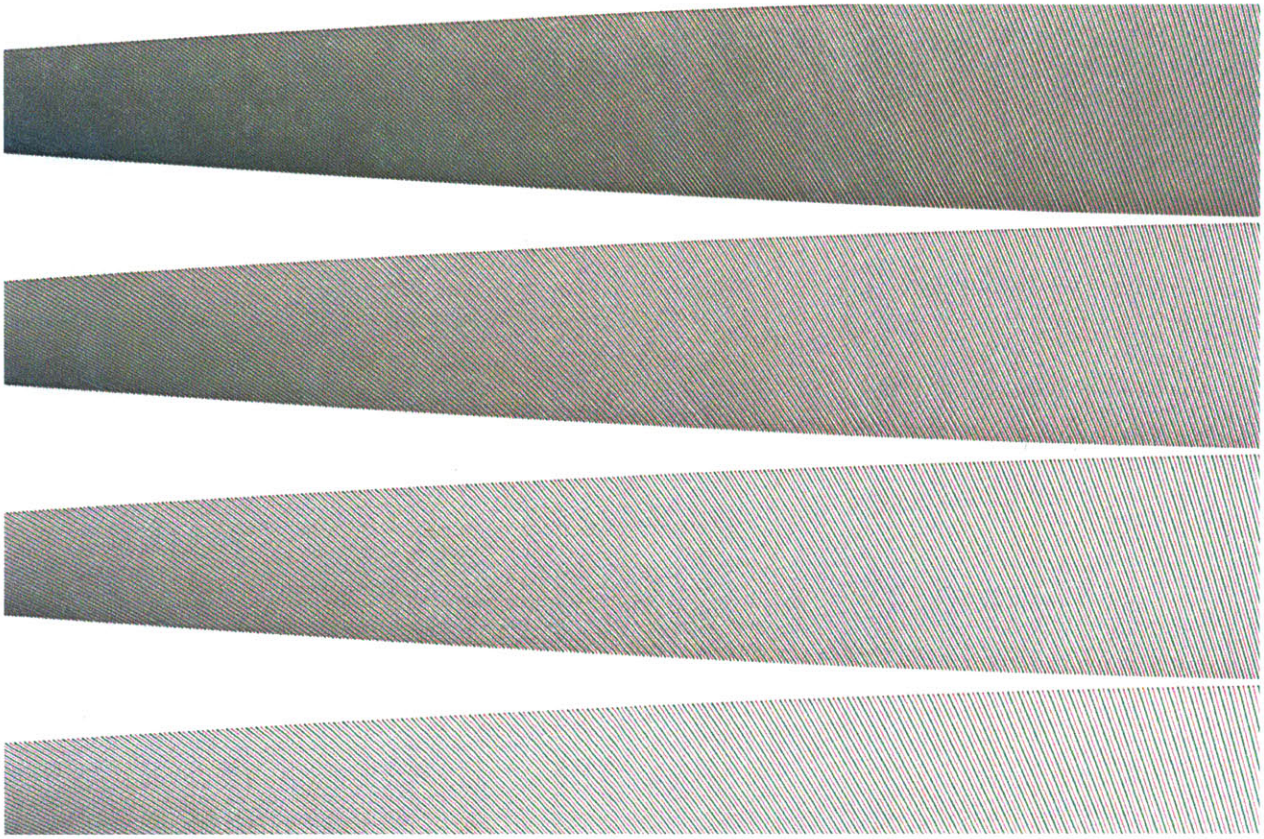


A-007-001: From Parallel to Pointed with Old Red and New Blue

Method 'A' treats the space of the paper as catalog of variables. In series 007, each drawing is a matrix of lines. These lines are spaced much more densely horizontally than vertically. The horizontal spacing is related to the vertical position on the paper. Each line is rotated around its midpoint at an angle depending on its horizontal position. The line farthest to the left is rotated 90° while the lines at the far right are rotated none. This run alternates between two pens, a moderately worn blue and a heavily worn red.

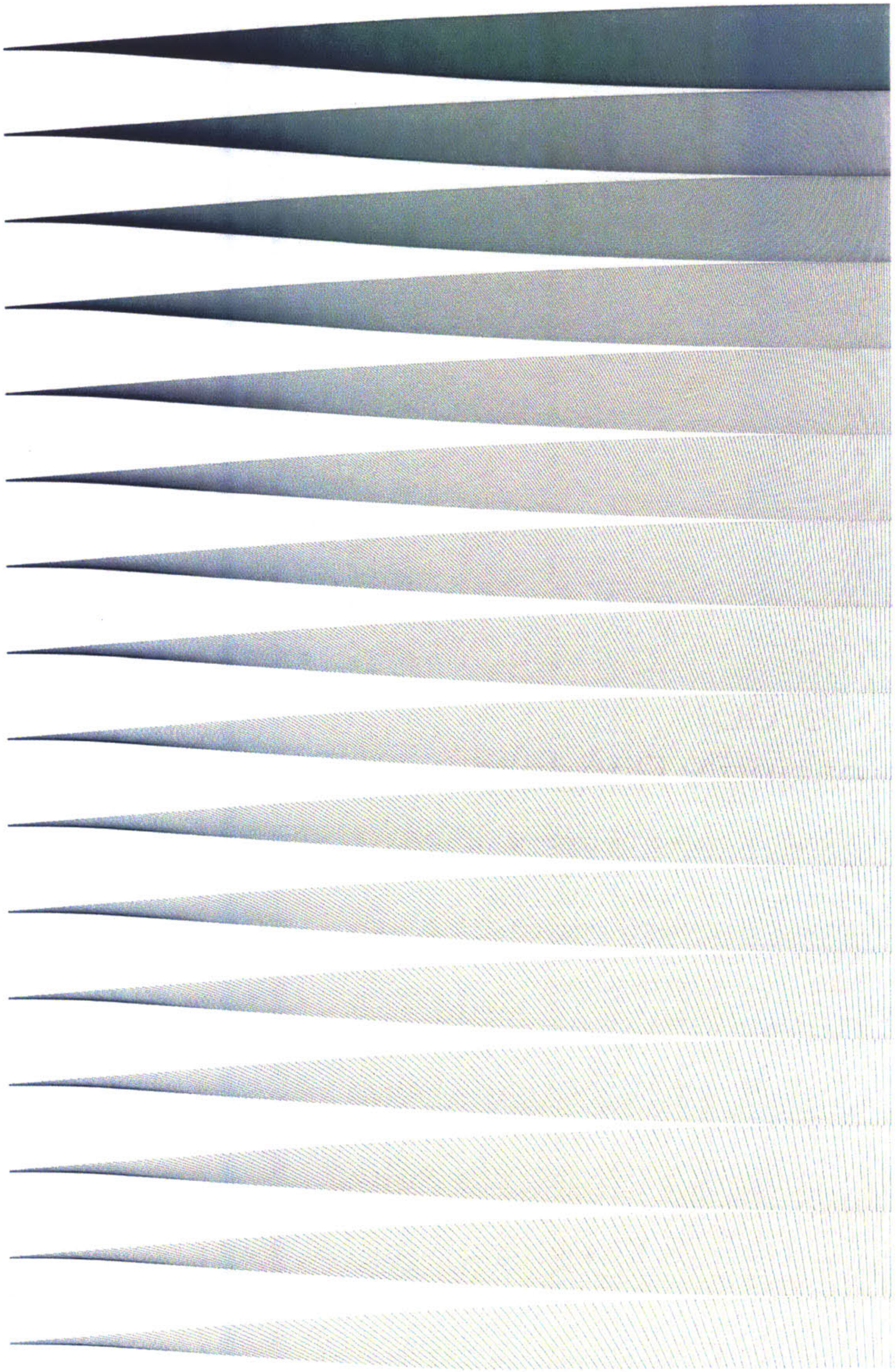


A-007-001

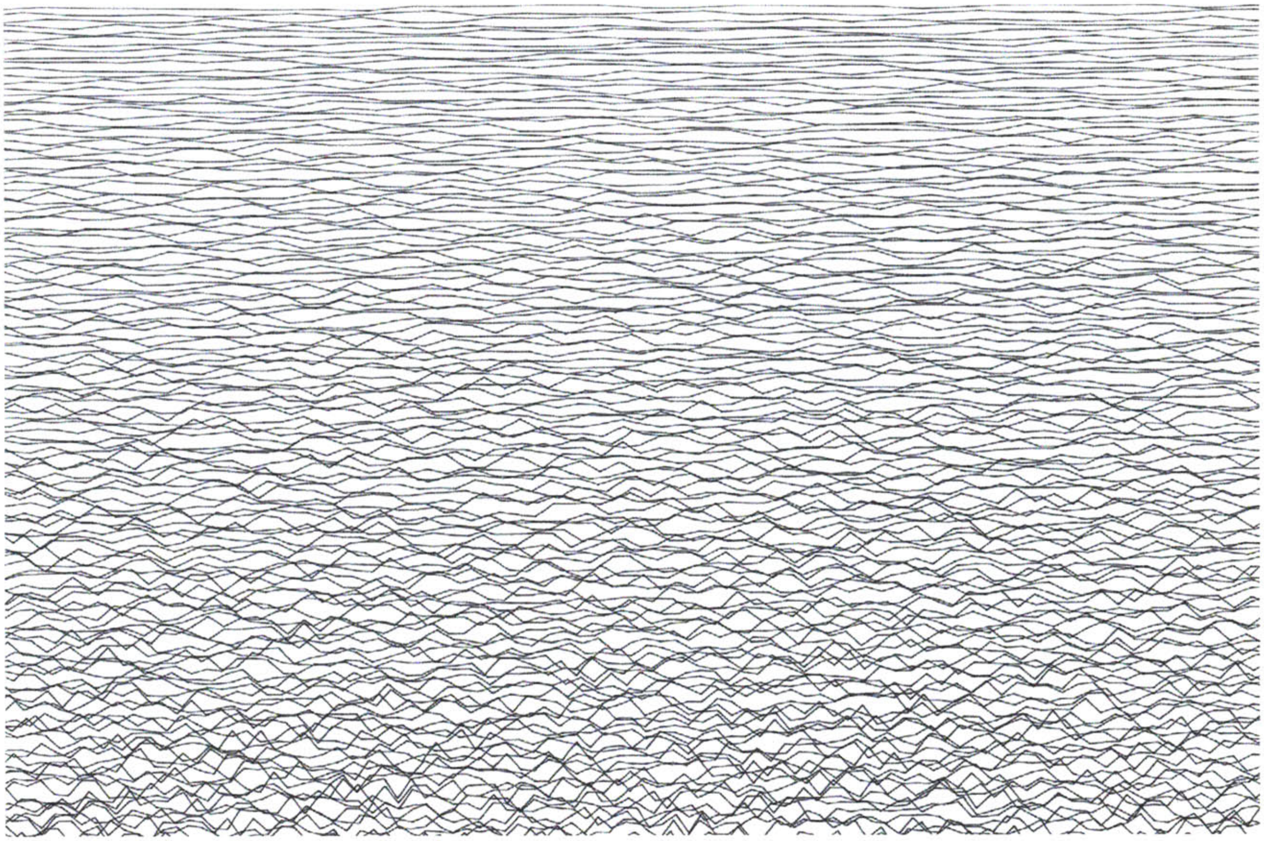


A-007-002: From Parallel to Pointed with Green and Magenta

Method 'A' treats the space of the paper as catalog of variables. In series 007, each drawing is a matrix of lines. These lines are spaced much more densely horizontally than vertically. The horizontal spacing is related to the vertical position on the paper. Each line is rotated around its midpoint at an angle depending on its horizontal position. The line farthest to the left is rotated 90° while the lines at the far right are rotated none. This run alternates between two pens, a moderately worn green and a moderately worn magenta.



A-007-002



A-008-001: Increasingly Random

Method 'A' treats the space of the paper as catalog of variables. In series 008, the variation occurs with respect to the vertical axis. Lower lines are made up of increasing number of segments and the likelihood of each vertex deviating from the local horizontal datum increases.

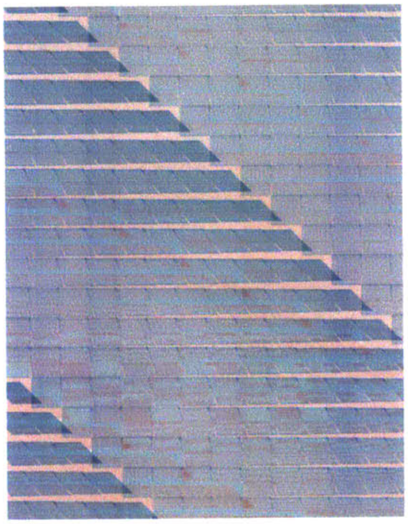
A-008-001

Method B

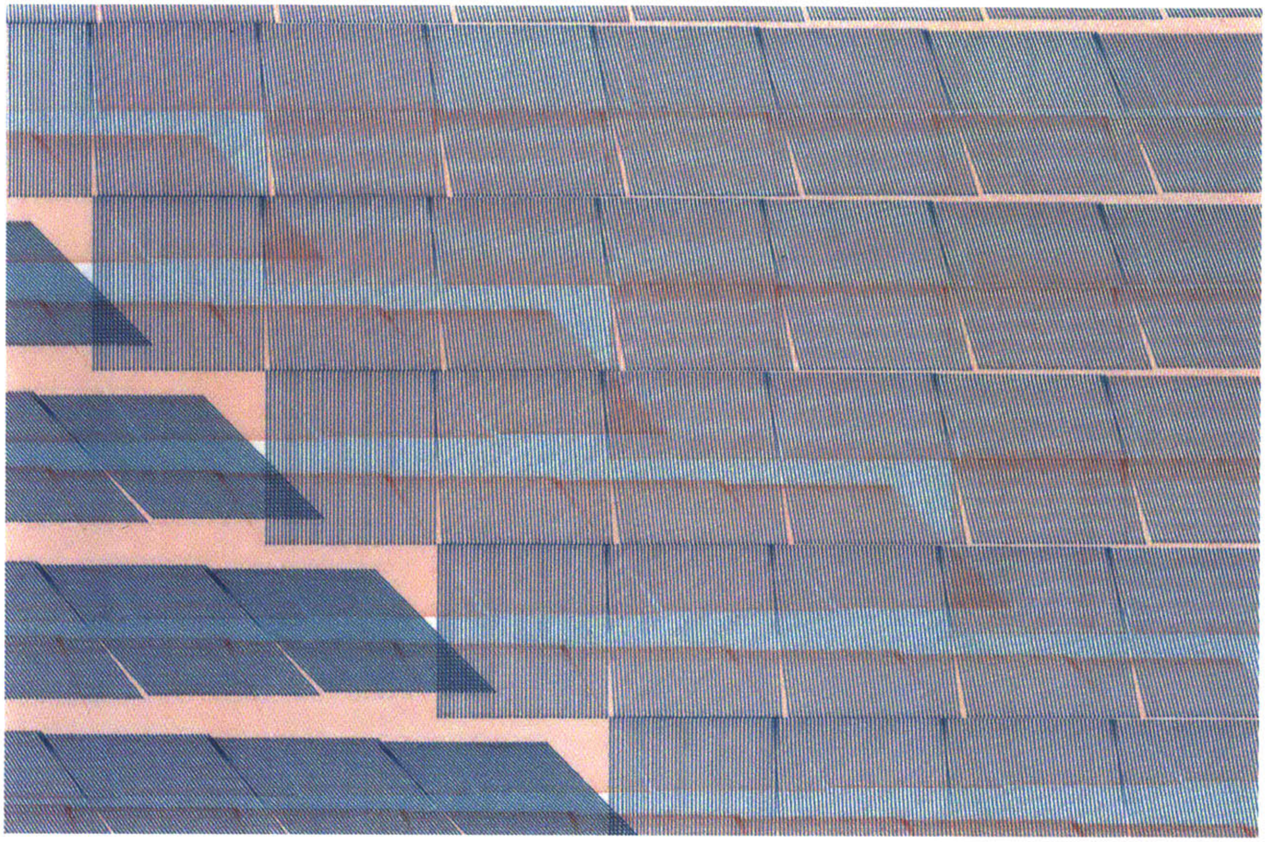
Method 'B' treats the space of the paper as catalog of variables but shifts the rows so that all possible adjacencies of swatch types are explored.

Series B-003

Swatches are made up of parallel lines. Before shifting, the angle of these lines are related to their position horizontally.

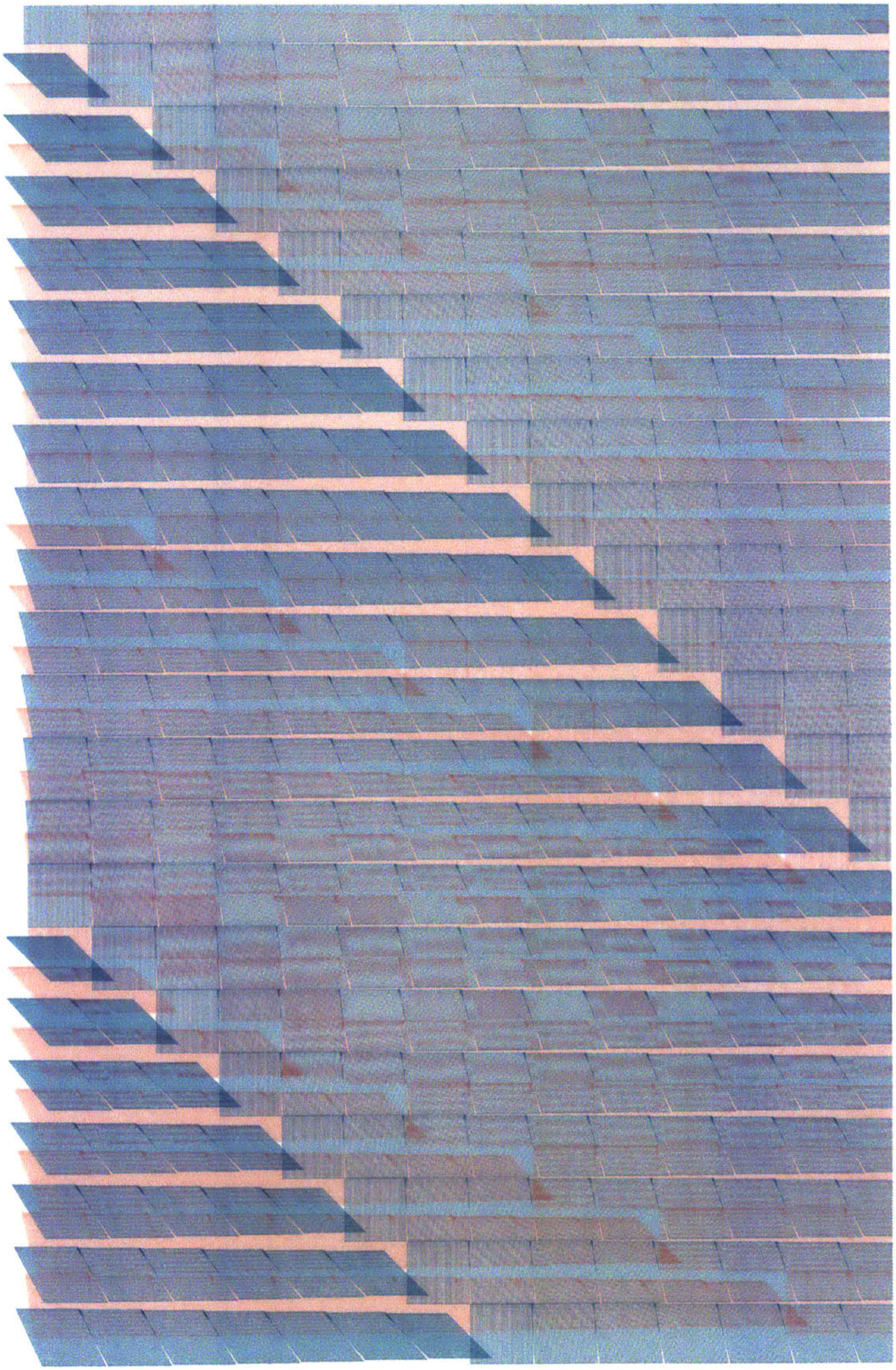


B-003-001



B-003-001: Swatches of Parallel Lines in a Matrix with Shifted Rows

Method 'B' treats the space of the paper as catalog of variables but shifts the rows so that all possible adjacencies of swatch types are explored. Swatches are vertically spaced so that they overlap with those above and below. In series 003, the swatches are made up of parallel lines. Before shifting, the angle of these lines are related to their position horizontally. Then, before drawing, every other row is shifted laterally by a number of modules that increases down the page. In this run, the rows are alternately drawn with red and cyan colored pens.



B-003-001

Method C

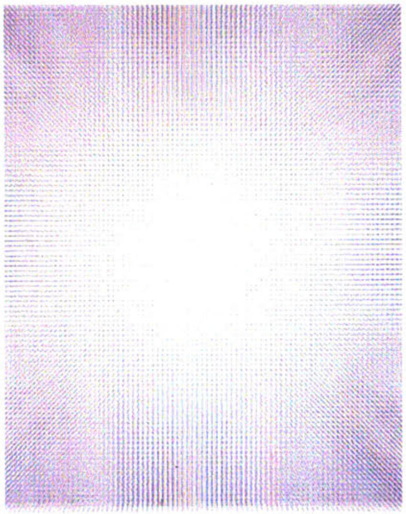
Method 'C' draws lines in grid sets with respect to a focal point.

Series C-001

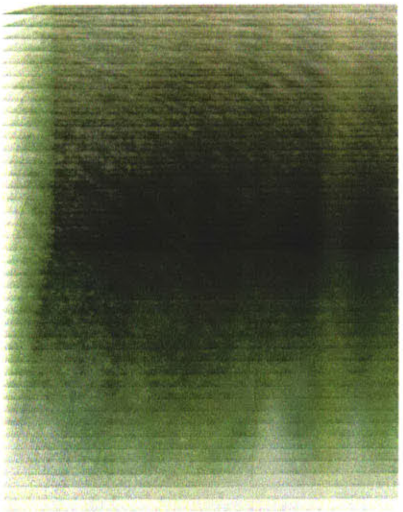
Two sets are layered. The spacing within each set is varied although the central focal point is shared.

Series C-003

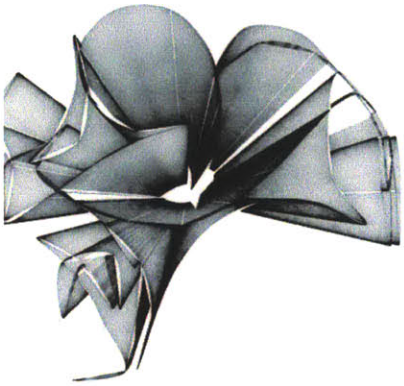
Three sets are layered. In two sets the lines are angled towards the focal point. The third set is made of vertical lines.



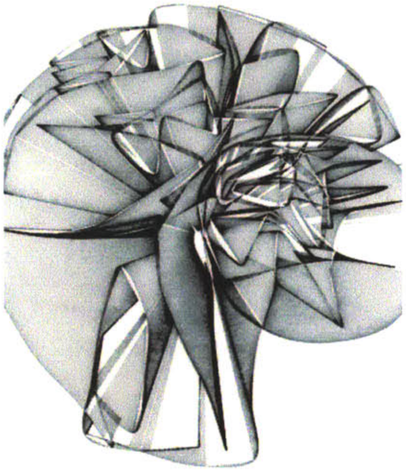
C-001-001



A-003-001



C-004-001



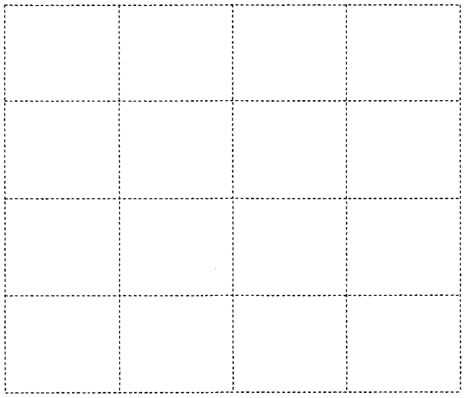
C-004-002

Series C-004

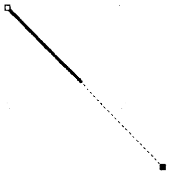
A coarse path is calculated first, within a circular boundary to avoid intersection. Lines drawn from interpolated spine curve points to the focus continue until intersection with the coarse path.

Series C-001

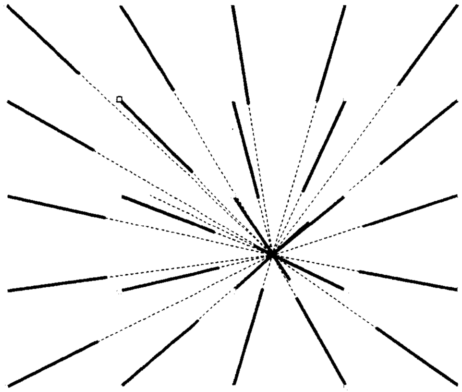
Two sets are layered. Each set is organized by a grid. The spacing within each set is varied although the central focal point is shared. Starting with the first grid point, a line is calculated from the grid point to the common focal point. (C-001-VectorDiagram-01) A line is drawn starting at the same grid point with a predefined length. (C-001-VectorDiagram-02) The process for creating these two related lines, the calculated and the drawn, is repeated for each grid point. (C-001-VectorDiagram-03)



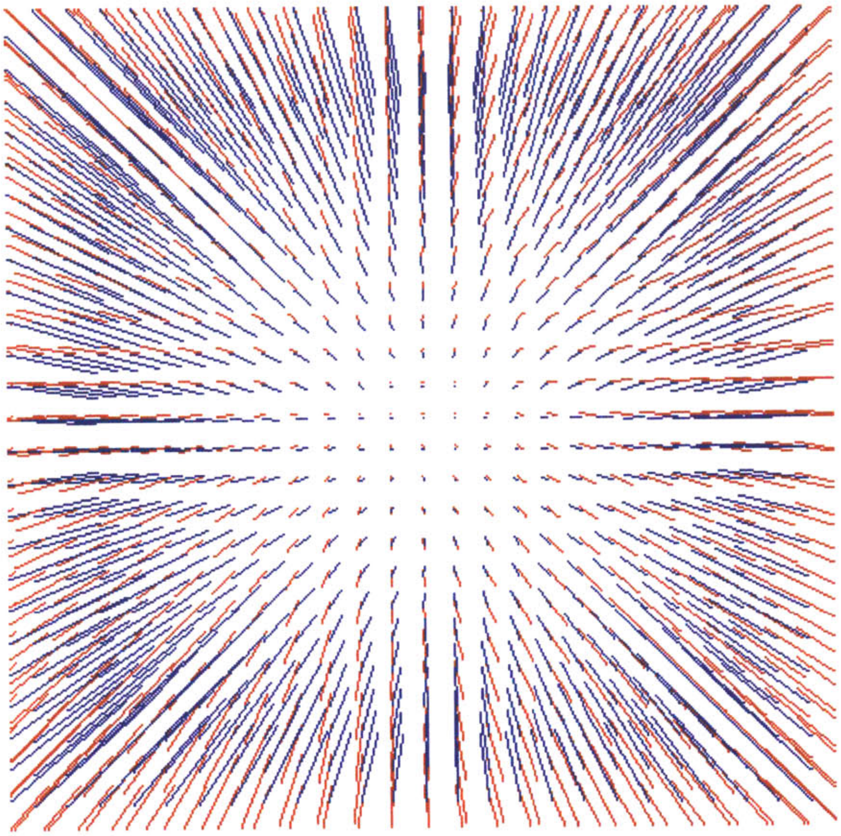
C-001-VectorDiagram-01



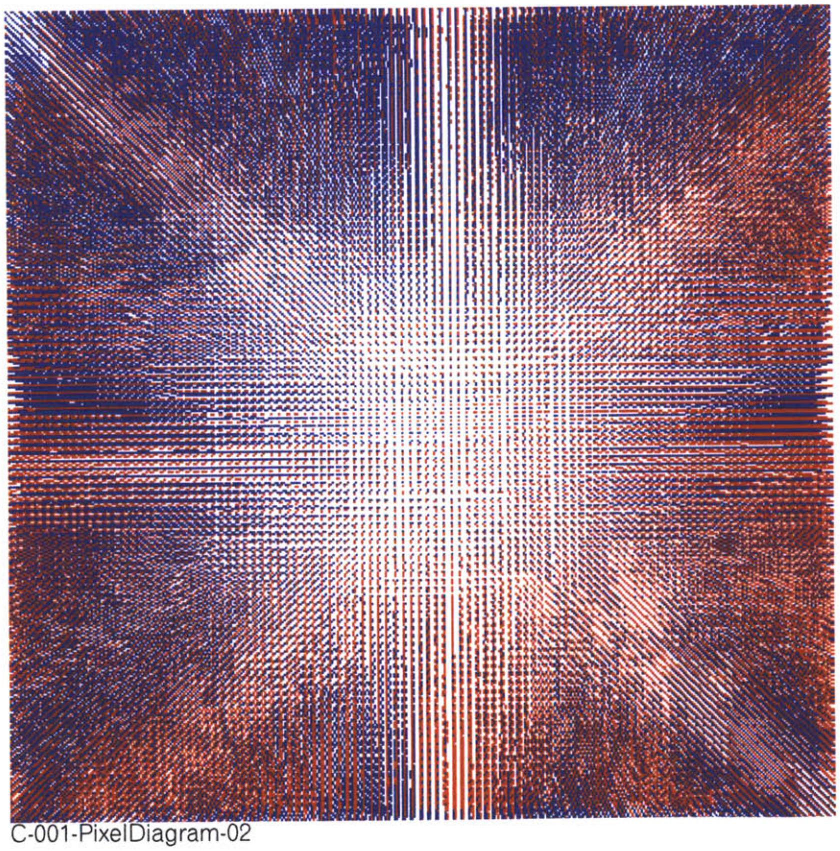
C-001-VectorDiagram-02



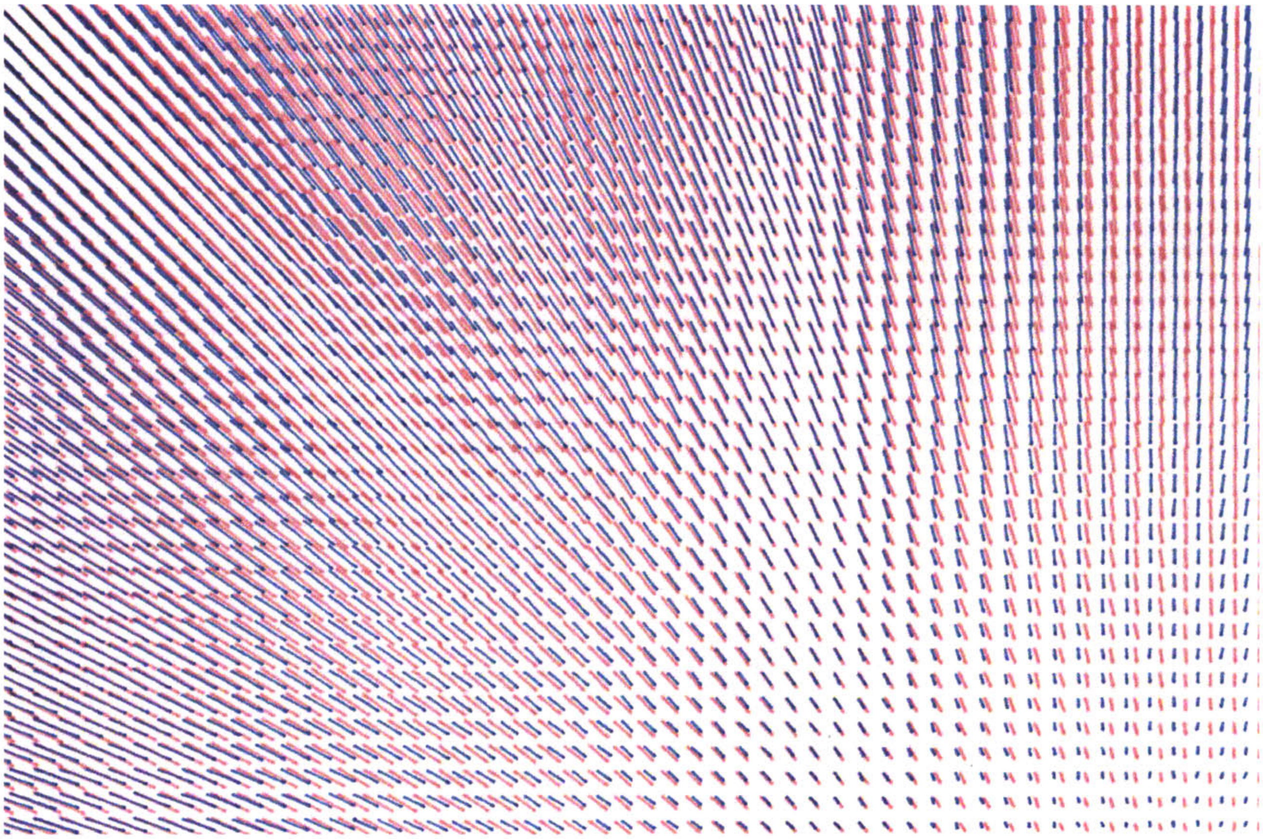
C-001-VectorDiagram-03



C-001-PixelDiagram-01

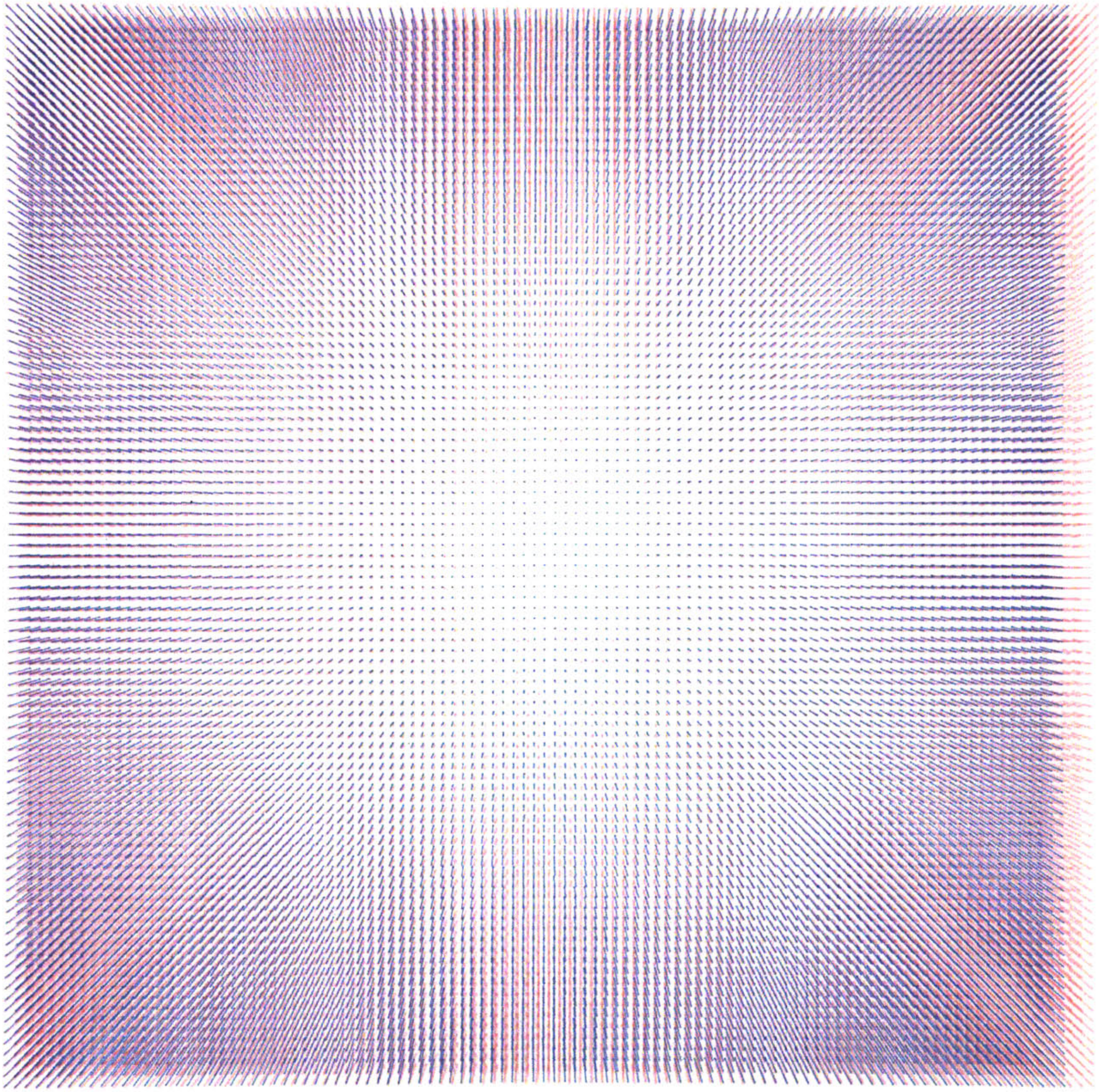


C-001-PixelDiagram-02



C-001-001: Multiple Centrally-Oriented Grids

Method 'C' draws lines in grid sets with respect to a focal point. In Series 003, Two sets are layered. The spacing within each set is varied although the central focal point is shared. In run 001, a set is drawn with a blue pen and another with a magenta pen.



C-001-001

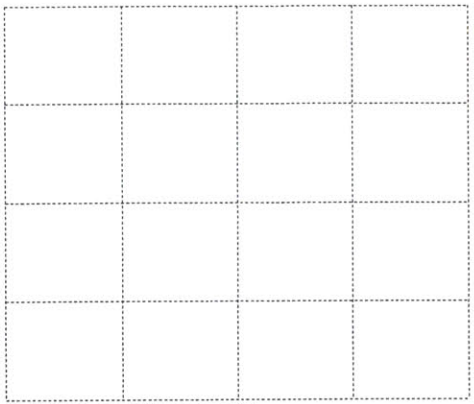
Series C-003

Three sets are layered. Each set is organized by a grid. The spacing within each set is varied although the central focal point is shared.

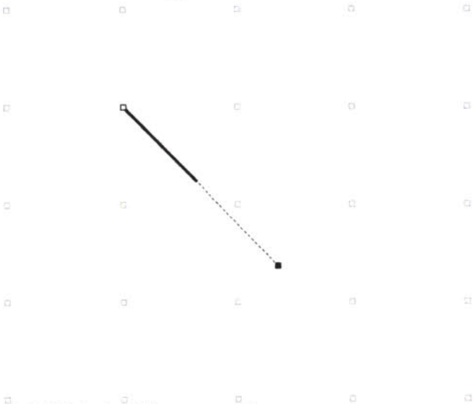
In the first two sets, lines are calculated from the grid points (C-003-VectorDiagram-01) to the common focal point. A line is drawn starting at the same grid point with a predefined length. (C-003-VectorDiagram-02)

In the third set, the focal point is ignored and lines are drawn at a constant length from each grid point.

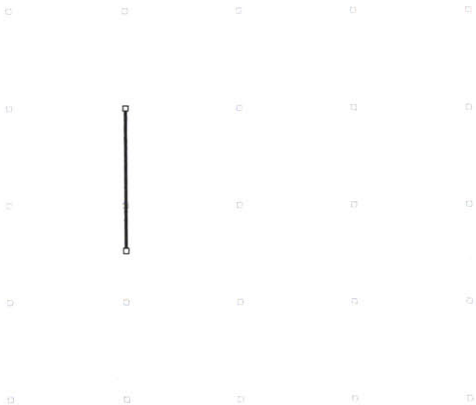
These three sets are drawn on the same sheet of paper (C-003-VectorDiagram-03)



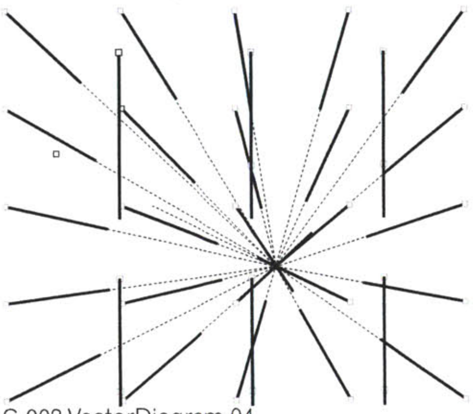
C-003-VectorDiagram-01



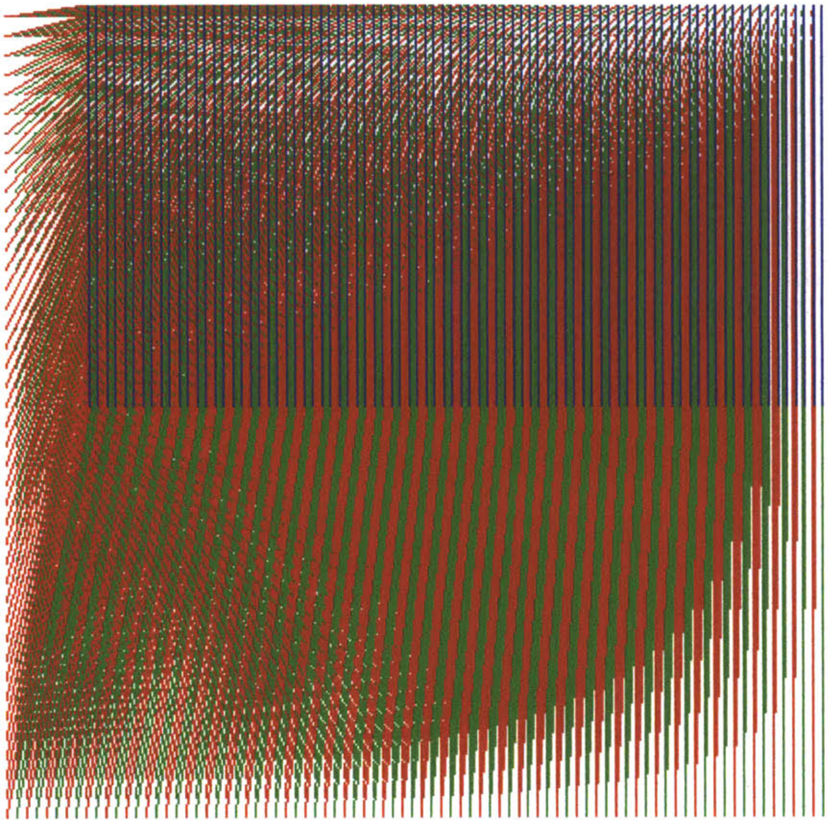
C-003-VectorDiagram-02



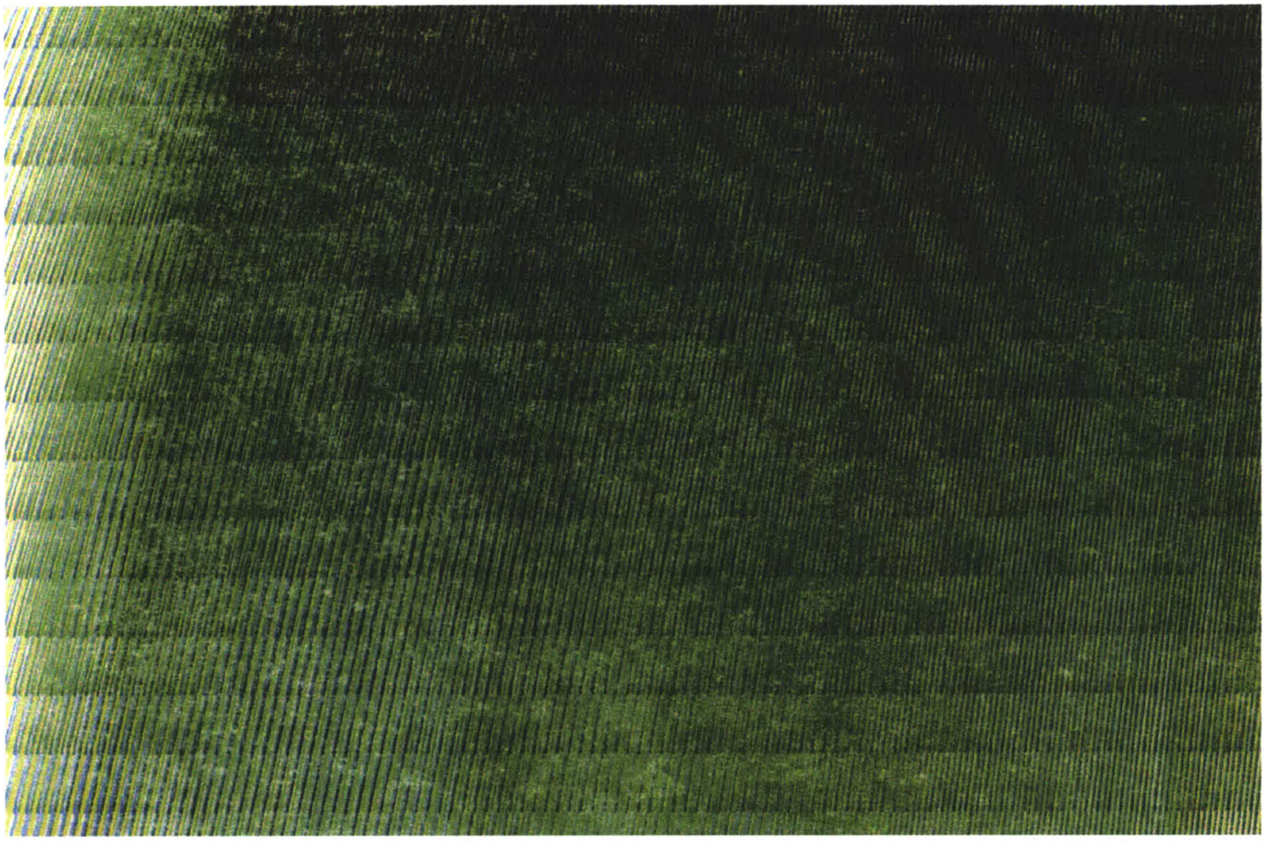
C-003-VectorDiagram-03



C-003-VectorDiagram-04

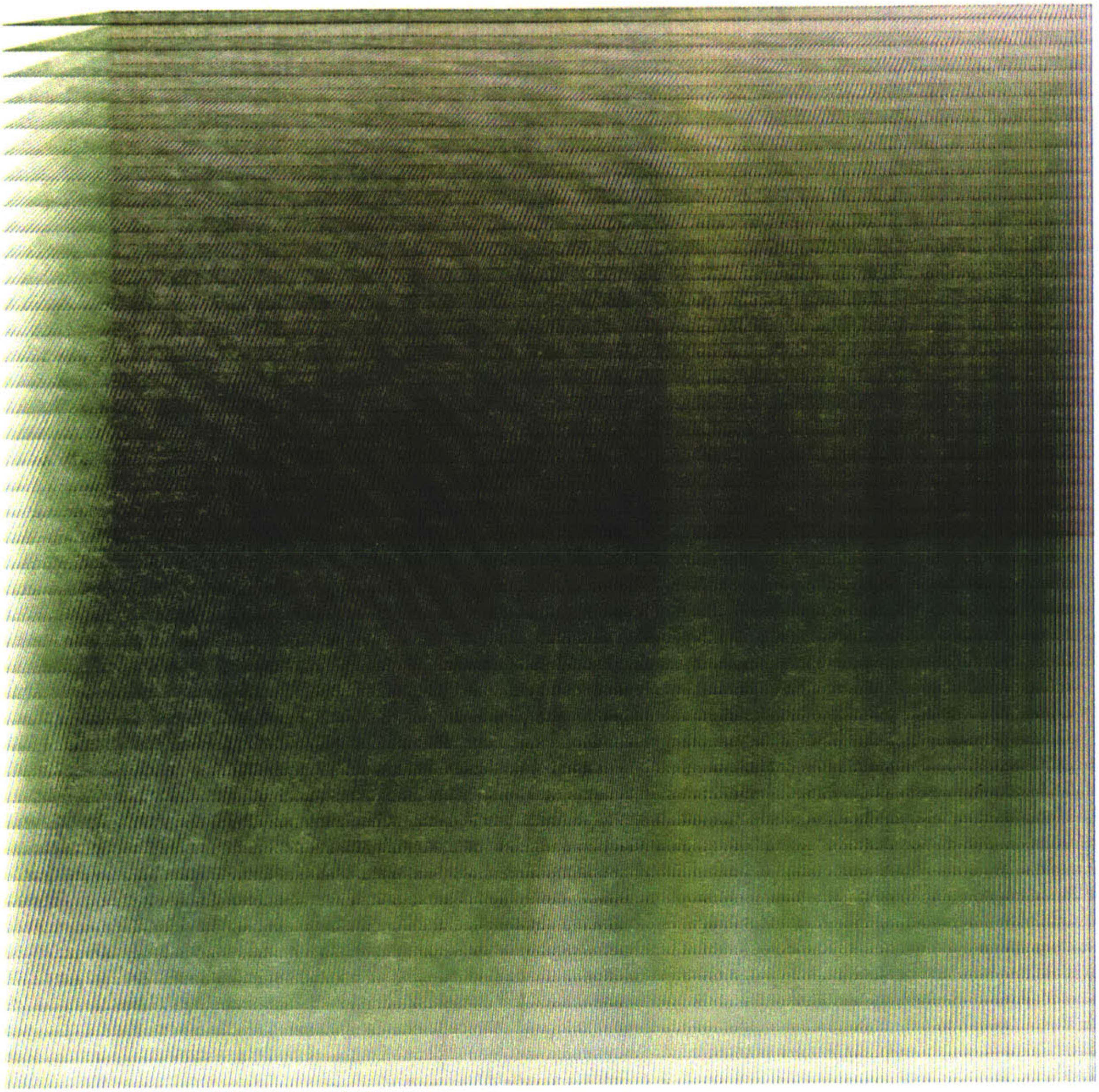


C-003-PixelDiagram-01



C-003-001: Towards Perspective

Method 'C' draws lines in grid sets with respect to a focal point. Series 003 layers three sets. In two sets the lines are angled towards the focal point. The third set is made of vertical lines.
In run 001, a set is drawn with a blue pen. The second set, closely aligned with the first set is drawn with a yellow pen. The third (vertical) set is drawn with a red pen.



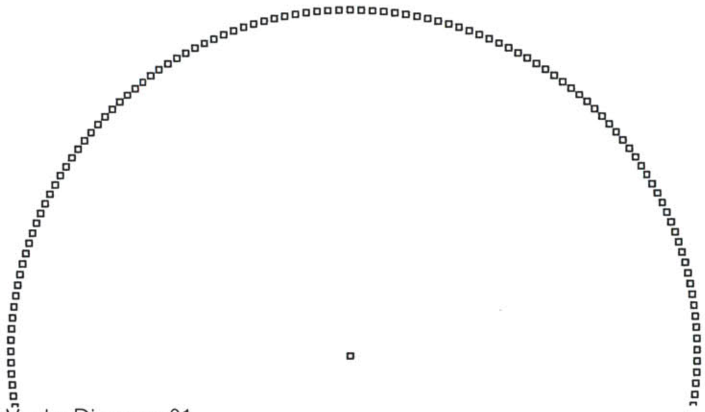
C-003-001

Series C-004

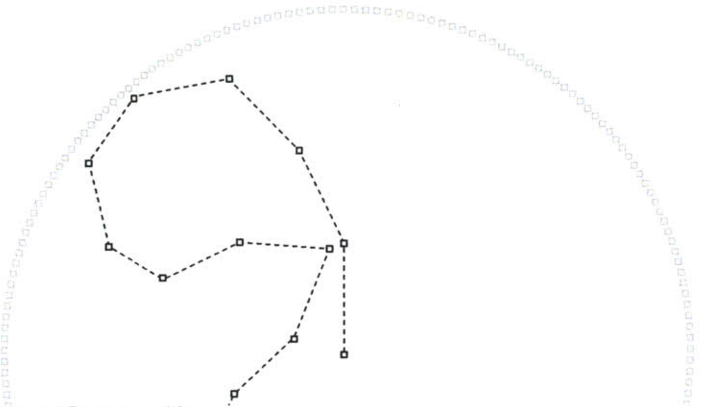
38 lines are constructed to make a circular boundary (C-004-VectorDiagram-01) A coarse path is then calculated by sequentially constructing segment lines. Each segment line is computed with respect to the previous segment vector and deviates by an angular acceleration value. The angular acceleration is subject to a random variation at each segment. If a computed line segment intersects any previous segment, it is deleted, and the vector reset. At this point the computed line may sharply change direction. (C-004-VectorDiagram-02) If vector could not possibly avoid intersection, the line is allowed to cross a prior segment.

20 points along a Catmull Rom spline are then calculated along each segment. (C-004-VectorDiagram-03)

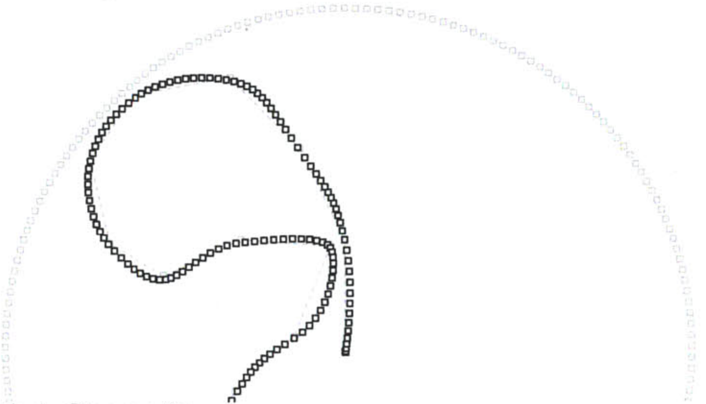
Lines drawn, beginning from interpolated spine curve points to the focus, continue until intersection with the coarse path.



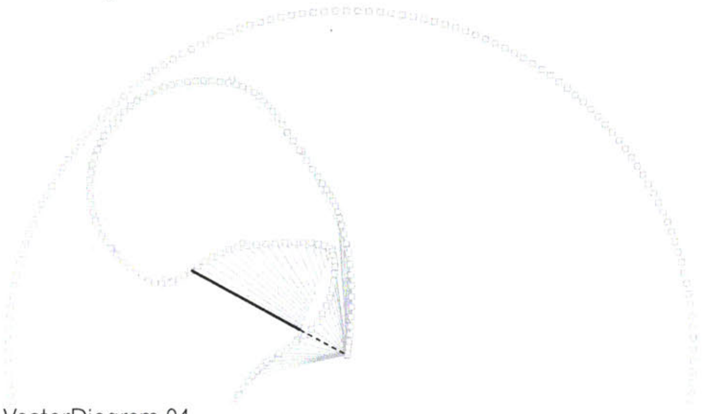
C-004-VectorDiagram-01



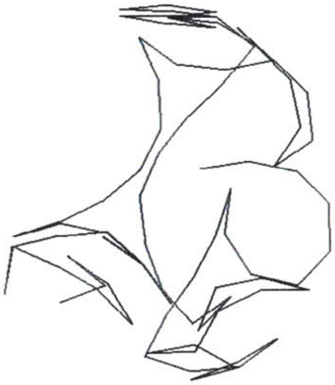
C-004-VectorDiagram-02

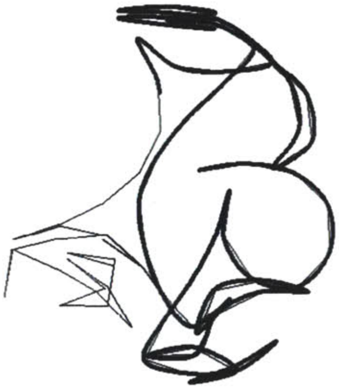


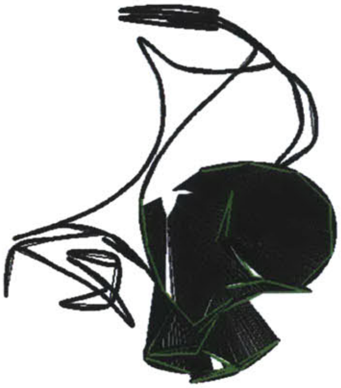
C-004-VectorDiagram-03

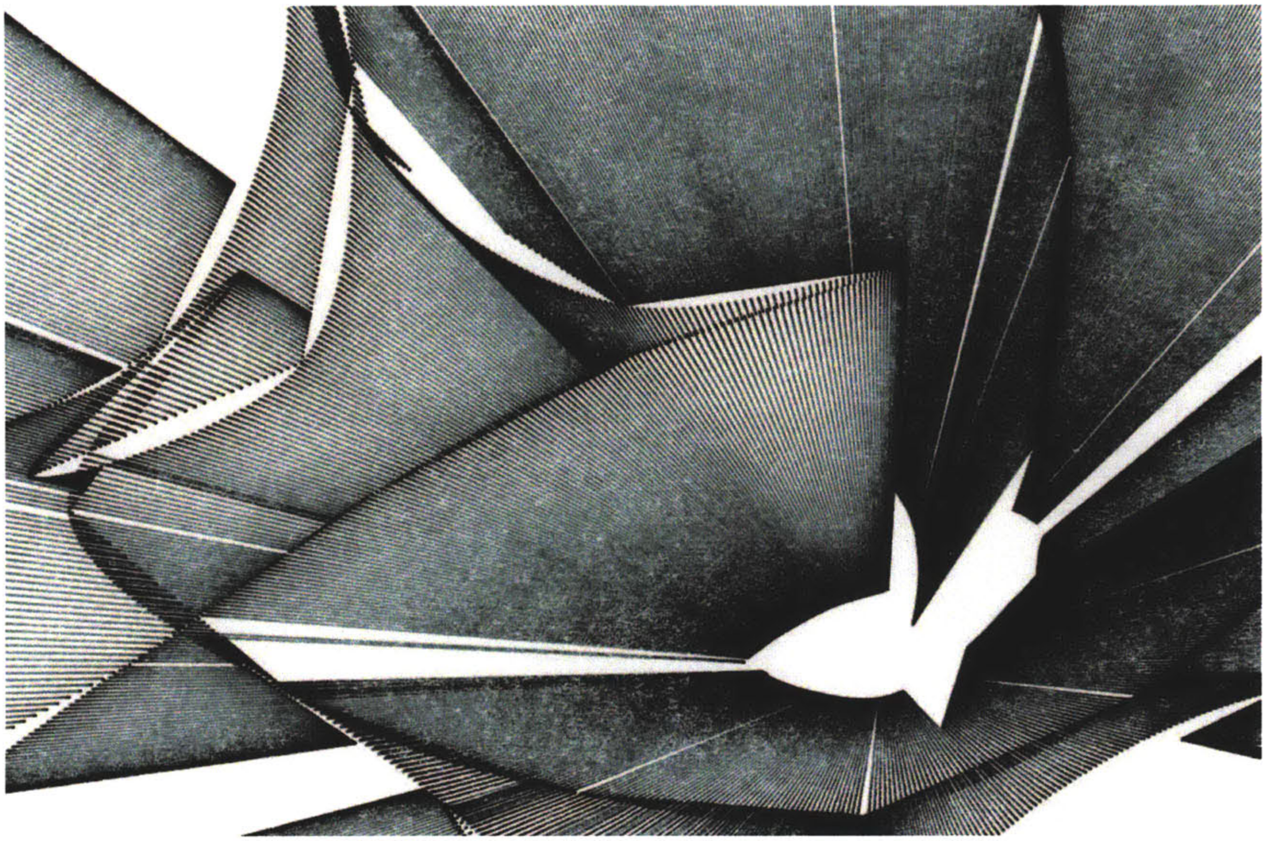


C-004-VectorDiagram-04


C-004-PixelDiagram-01

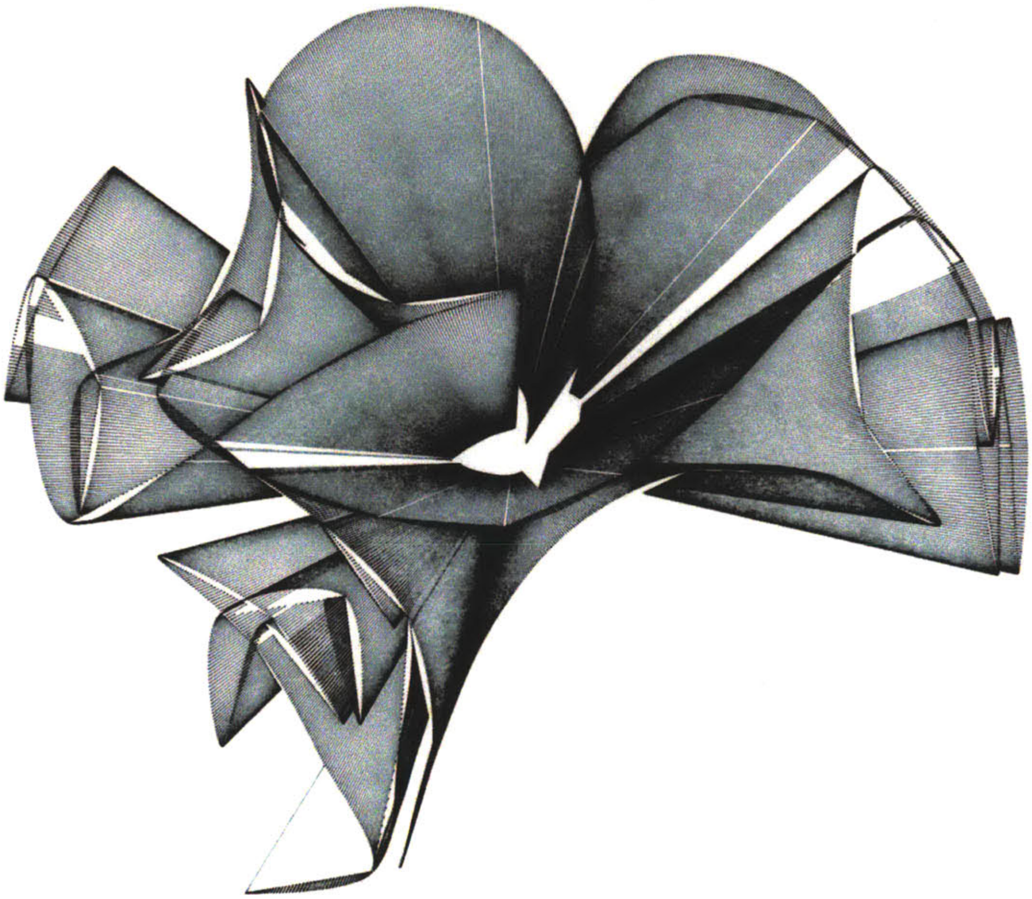

C-004-PixelDiagram-02


C-004-PixelDiagram-03

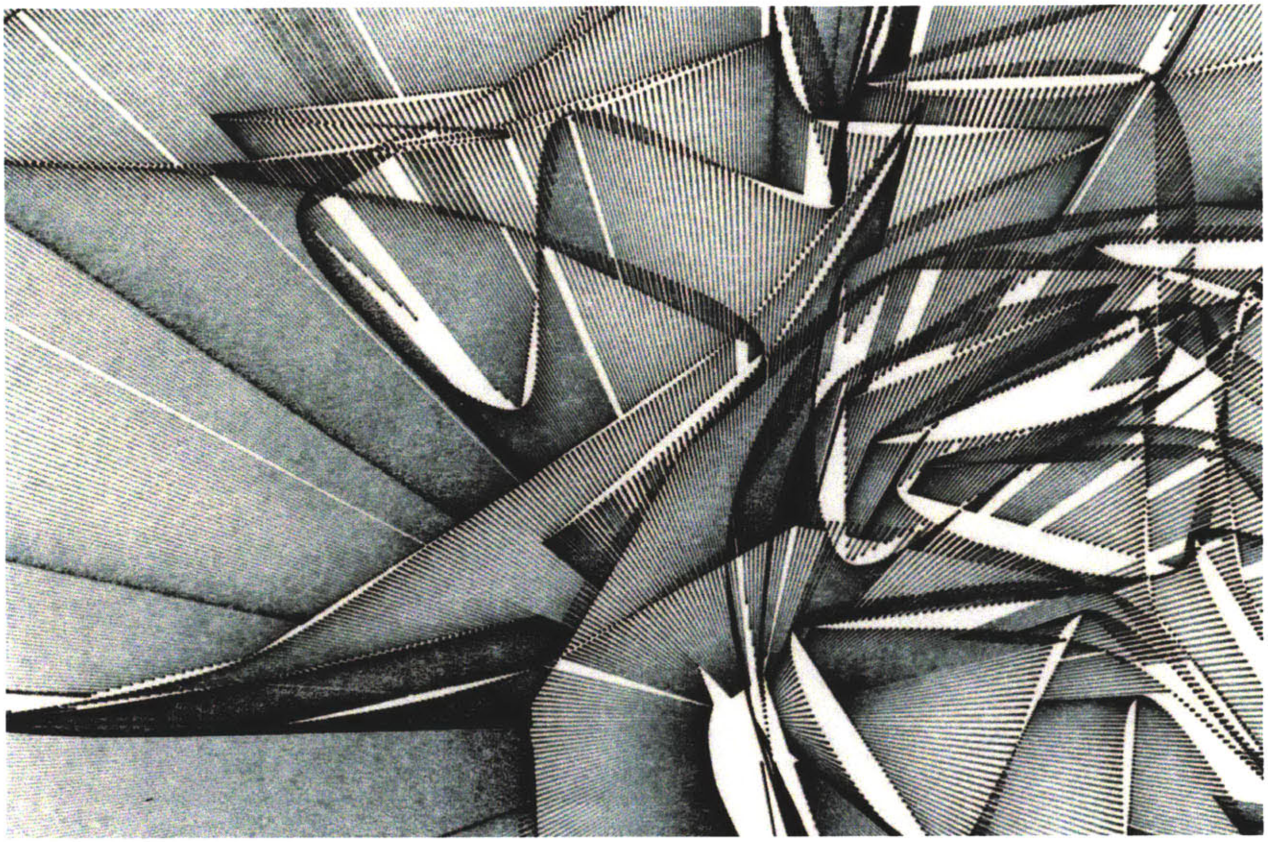


C-004-001: Towards Spherical Figure

Method 'C' draws lines in sets with respect to a focal point. In Series 004, a single coarse path is calculated first, within a circular boundary. The path tends to avoid intersections. A finer path with 50 points is generated per each initial point, and forms an interpolated spine curve. Each new point marks the beginning of a drawn line towards, but not ending at, the focal point. Lines stop at any intersection with the path. Run 001 is the result of a 70-segment coarse curve. The plotter is loaded with a highly worn pen.

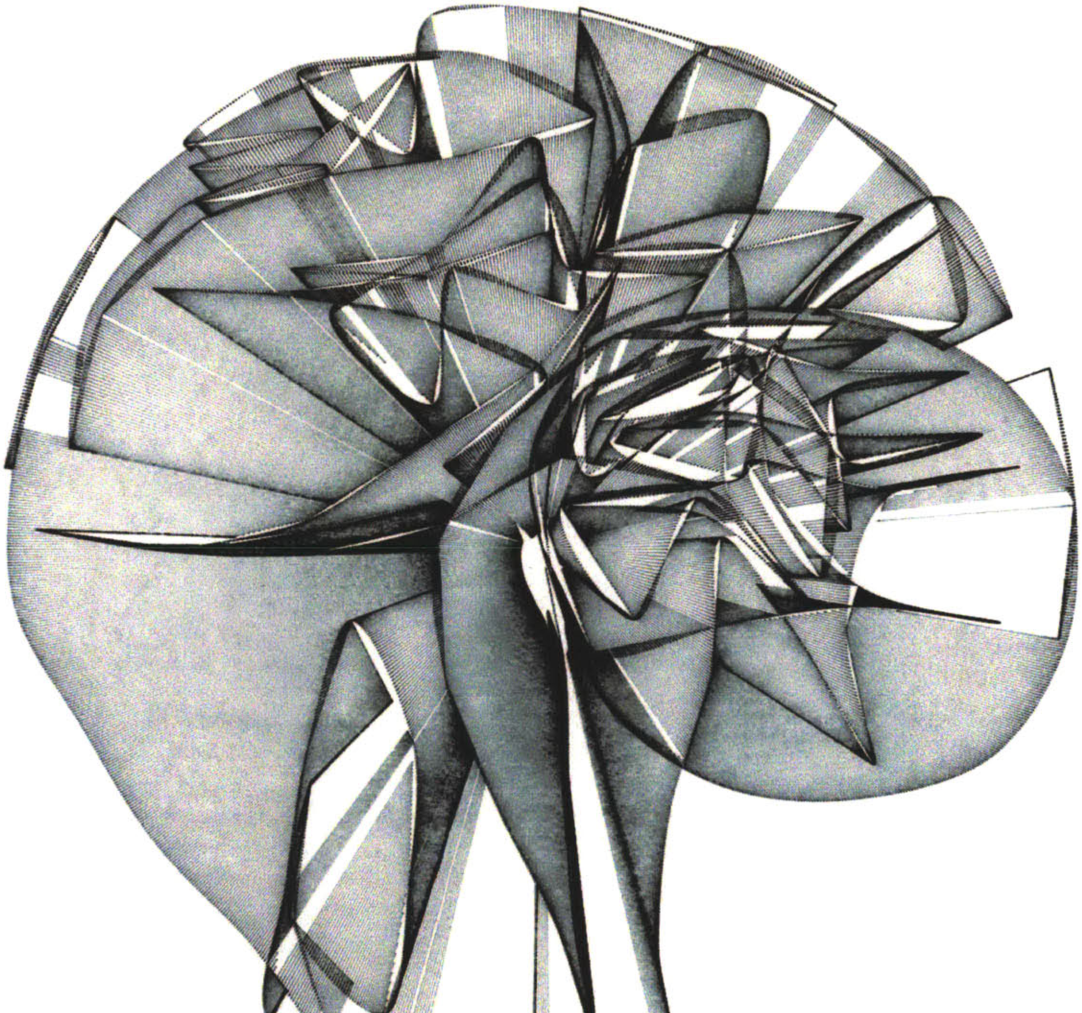


C-004-001



C-004-002: Densely Towards Spherical Figure

Method 'C' draws lines in sets with respect to a focal point. In Series 004, a single coarse path is calculated first, within a circular boundary. The path tends to avoid intersections. A finer path with 50 points is generated per each initial point, and forms an interpolated spine curve. Each new point marks the beginning of a drawn line towards, but not ending at, the focal point. Lines stop at any intersection with the path. Run 001 is the result of a 120-segment coarse curve. The plotter is loaded with a highly worn pen.



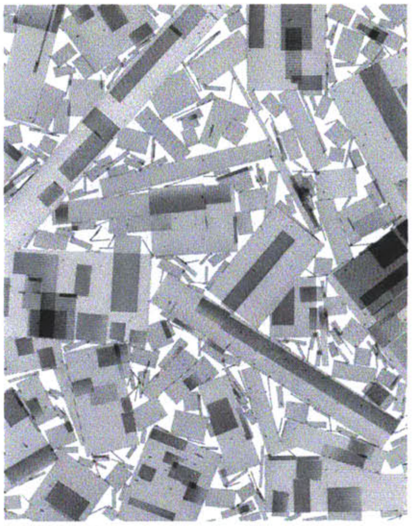
C-004-002

Method D

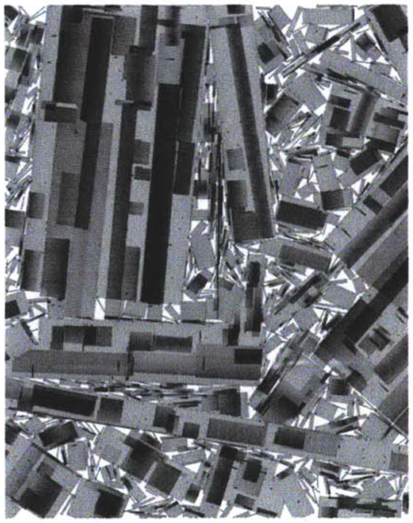
Method 'D' involves algorithms that treat the paper as space and ink as object.

Series D-001

A line is drawn at a random angle that will not intersect any previously drawn line. Lines of the same length are drawn slightly offset from the previous line until intersection.



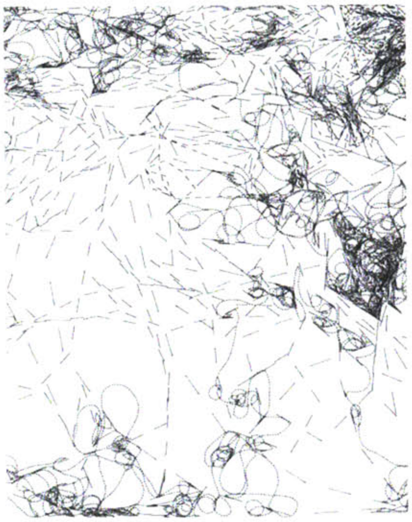
D-001-001



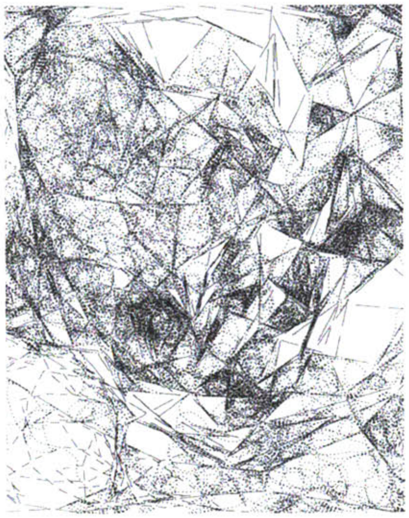
D-001-002

Series D-002

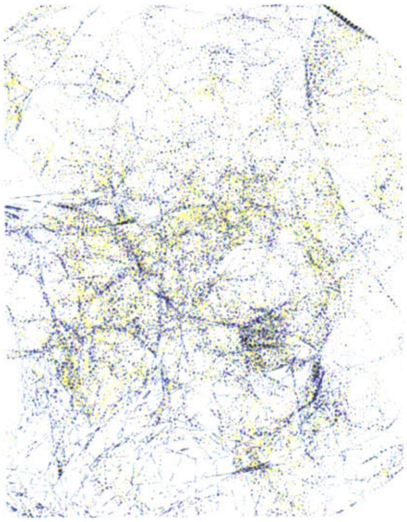
Marks are made by a particle "walking" in a random but generally curving path within a boundary on the page. Marks gradually decrease in length and avoid intersection.



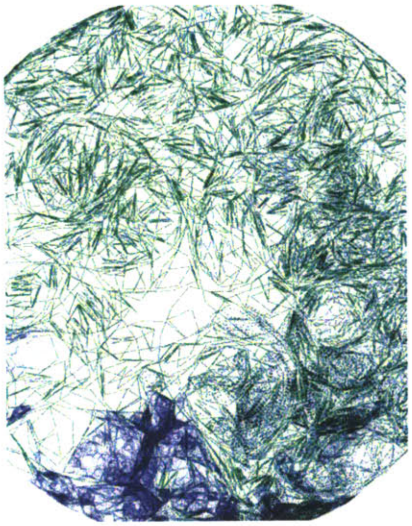
D-002-001



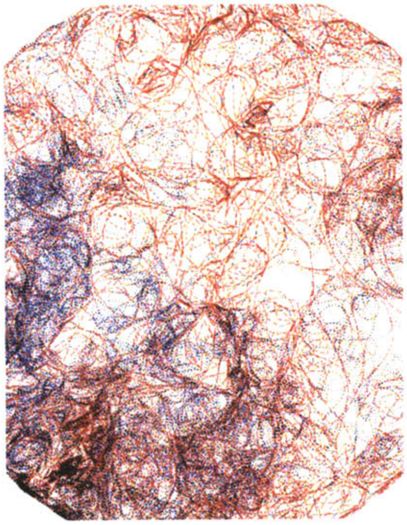
D-002-003



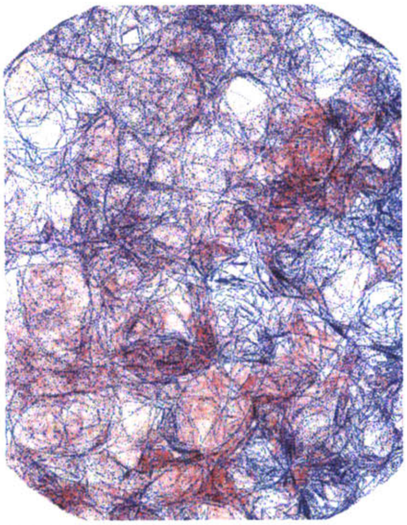
D-002-006



D-002-007



D-002-008

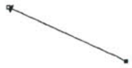


D-002-009

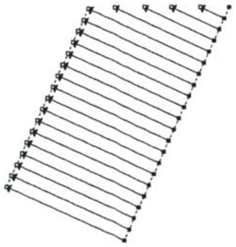
Series D-001

A line is drawn at a random angle that will not intersect any previously drawn line. (D-001-VectorDiagram-01) This line is defined in terms of a start point and an end point.

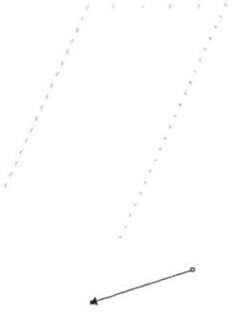
Lines of the same length are drawn slightly offset from the previous line. This process is repeated until the new line intersects with any other line, including the boundary lines. (D-001-VectorDiagram-02). If the line slopes upward, its end point is adjusted to the intersect point. Otherwise, the cycle of repetition is broken and a new line is drawn at a random position and angle. (D-001-002-VectorDiagram-03)



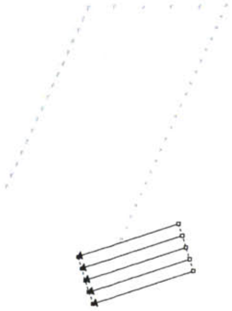
D-001-VectorDiagram-01



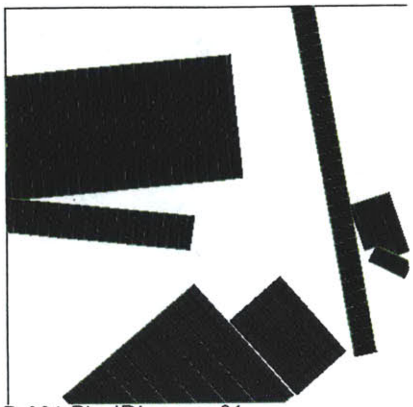
D-001-VectorDiagram-02



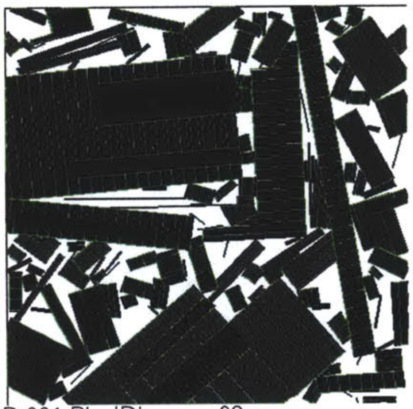
D-001-VectorDiagram-03



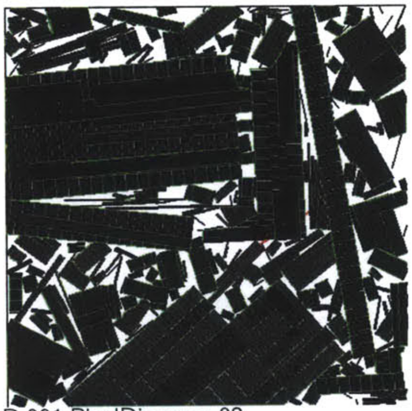
D-001-VectorDiagram-04



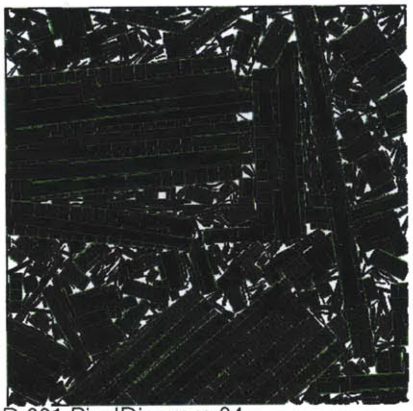
D-001-PixelDiagram-01



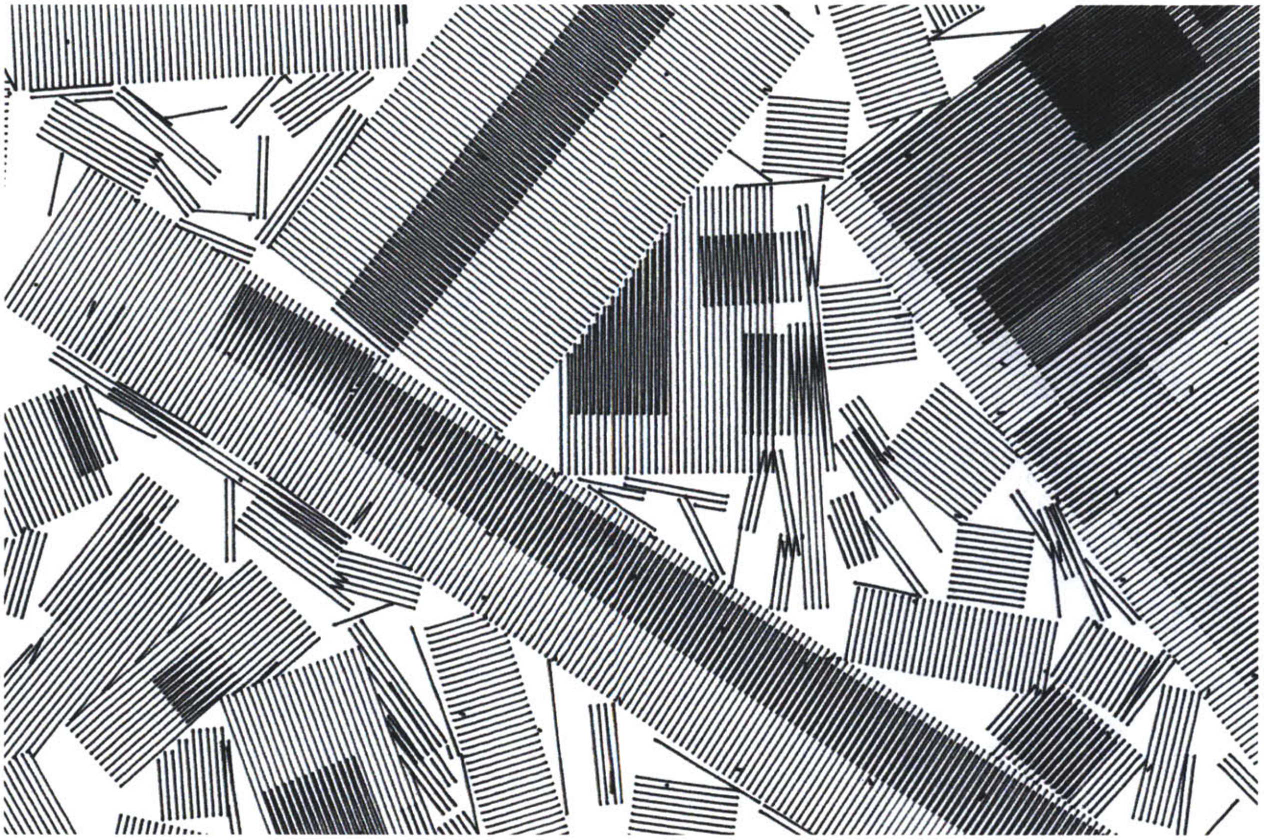
D-001-PixelDiagram-02



D-001-PixelDiagram-03

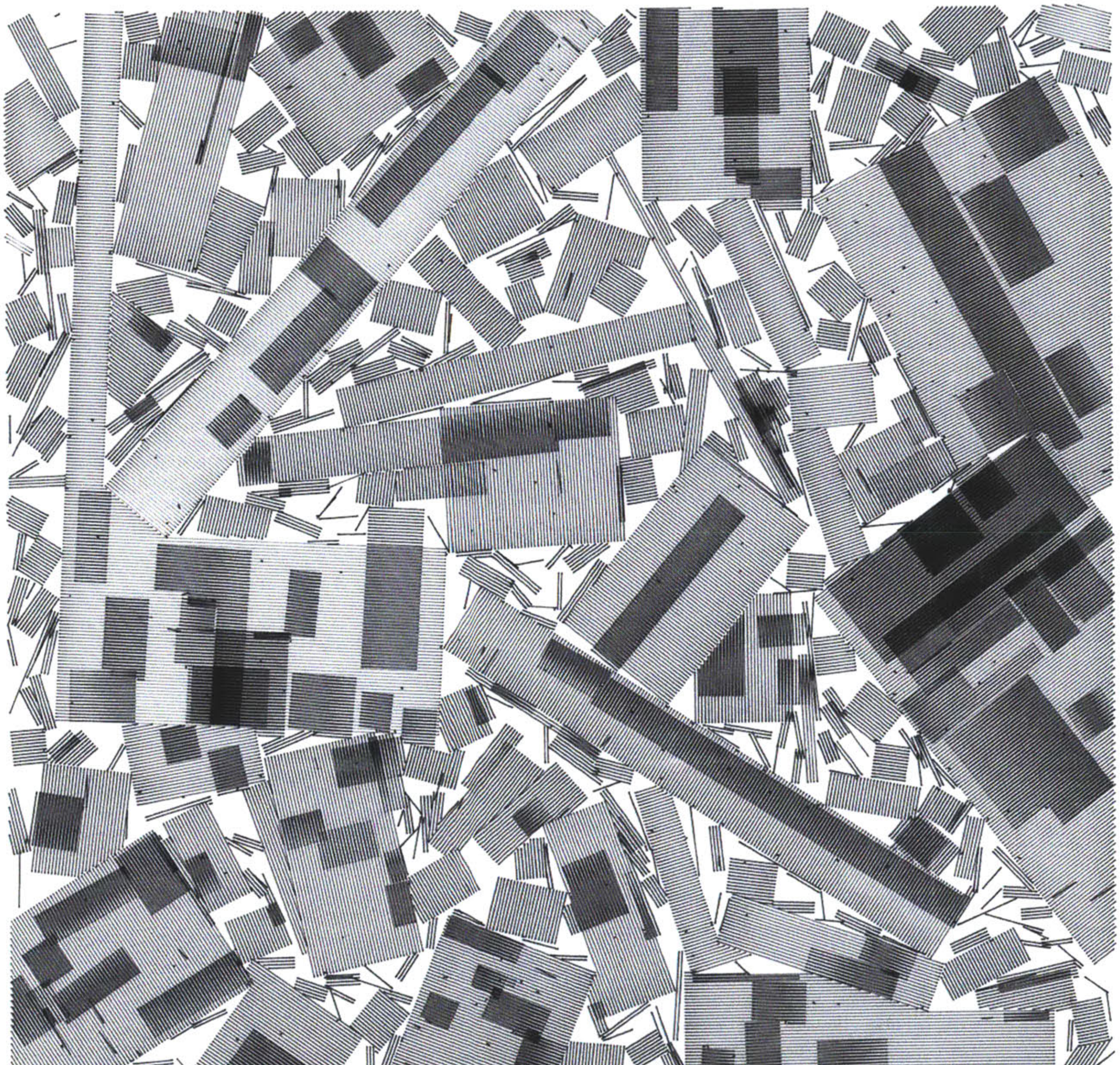


D-001-PixelDiagram-04

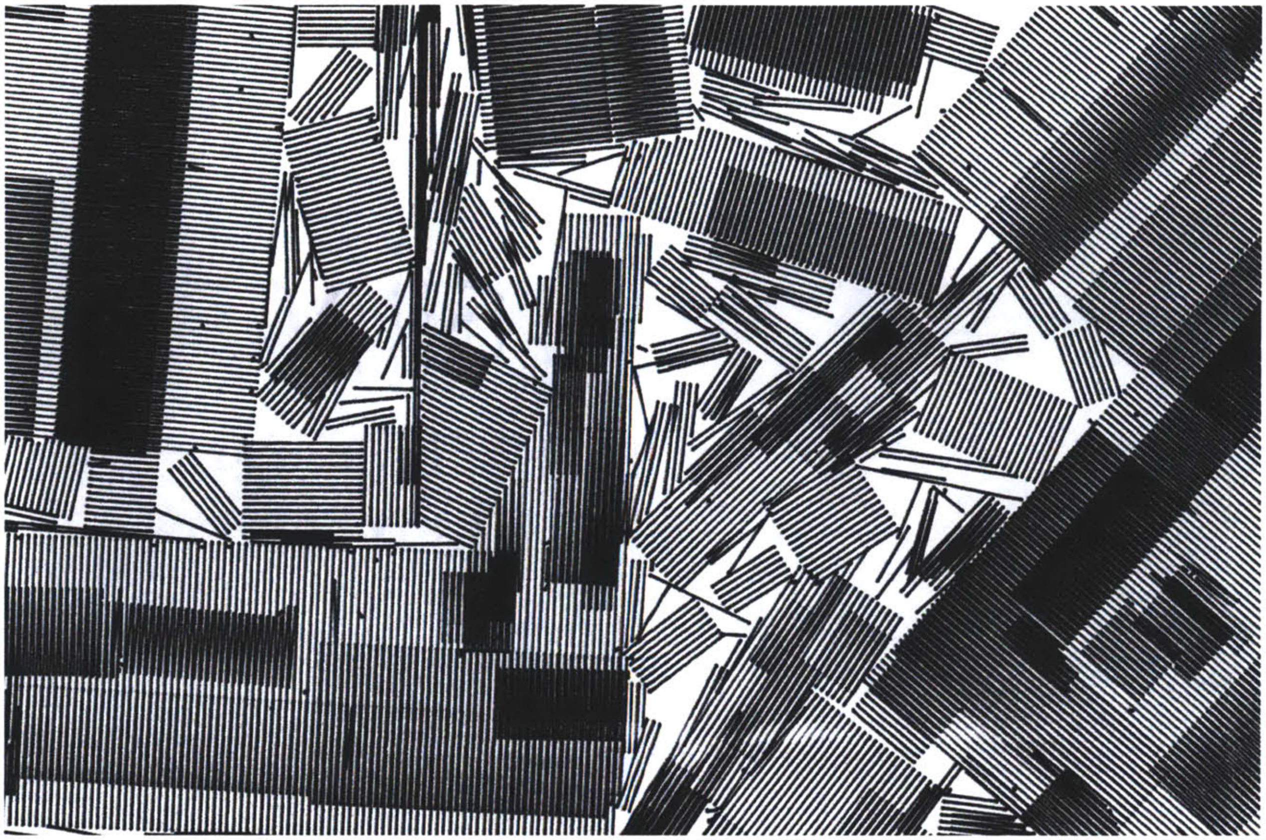


D-001-001: Intersection-free hatching, for Some Time

Method 'D' involves algorithms that treat the paper as space and ink as object. Series 001 draws a line at a random angle that will not intersect any previously drawn line. Then lines of the same length are drawn slightly offset from the previous line. When the next line will intersect any previously drawn line, repeat the process. Run 002 is drawn with around 12,000 lines.

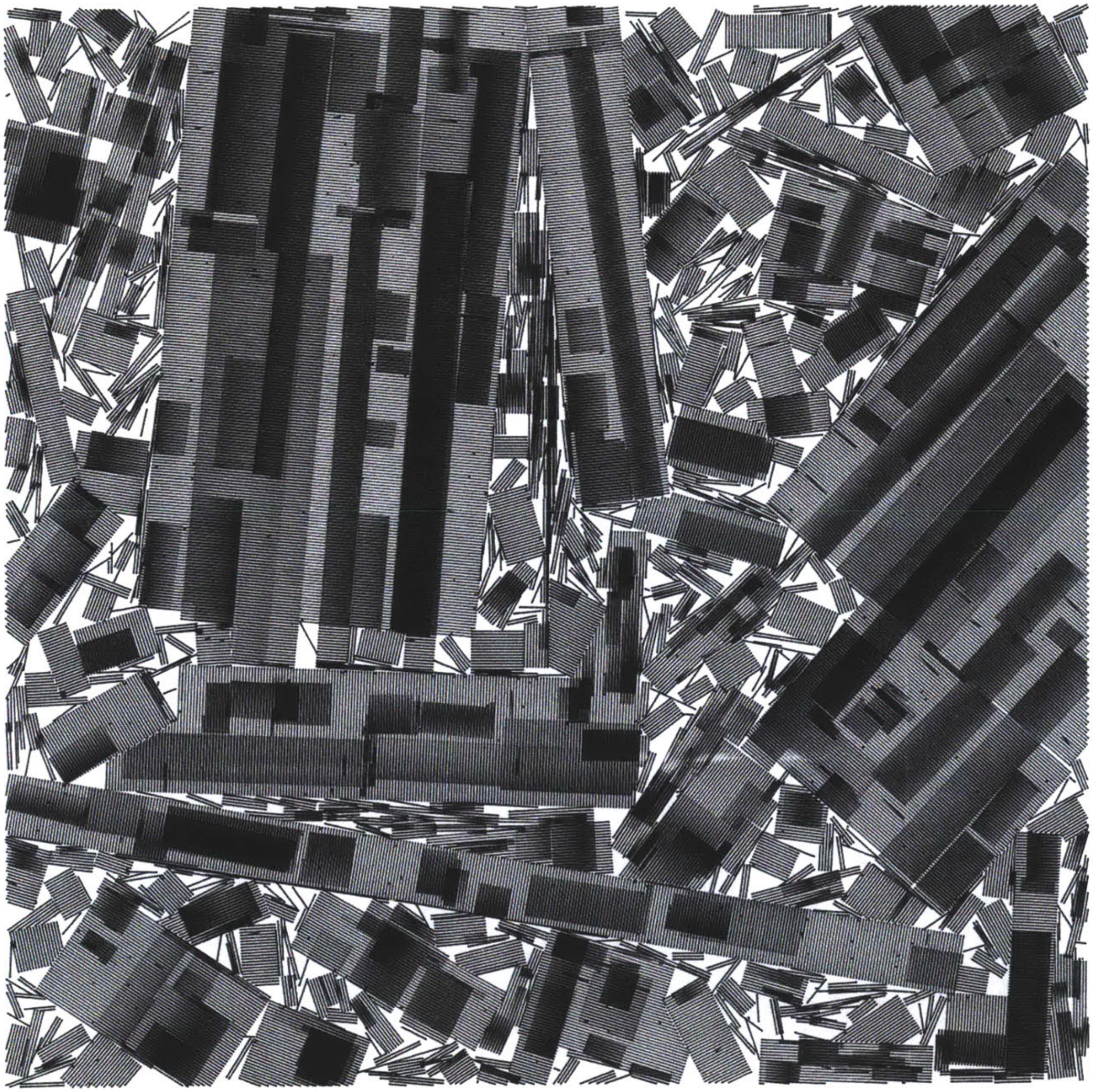


D-001-001



D-001-002: Intersection free hatching, for a Long Time

Method 'D' involves algorithms that treat the paper as space and ink as object. Series 001 draws a line at a random angle that will not intersect any previously drawn line. Then lines of the same length are drawn slightly offset from the previous line. When the next line will intersect any previously drawn line, repeat the process. Run 002 is drawn with around 12,000 lines.



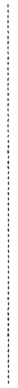
D-001-002

Series D-002

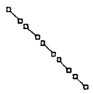
Marks are made by a particle "walking" in a random but generally curving path within a boundary on the page. Marks gradually decrease in length and avoid intersection.

The broadest structure of line in this series is the "walking" path. The secondary structure consists of each "step," (D-002-VectorDiagram-02) which is in turn structured by a series of marks separated by a proportional spacing. The path maintains a reference to the current trajectory and angular acceleration.

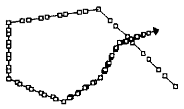
Each step line is created based on the current trajectory and angular acceleration. The angular acceleration becomes more extreme as necessary to avoid intersections of the mark lines. (D-002-VectorDiagram-03) As a result, the "path" appears to cross itself only at the gaps in its own consistency.



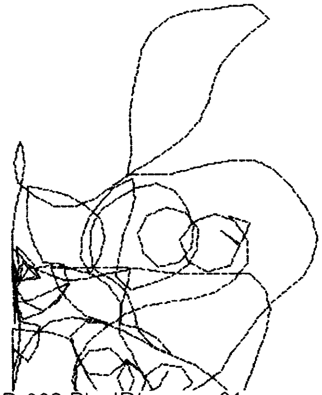
D-002-VectorDiagram-01



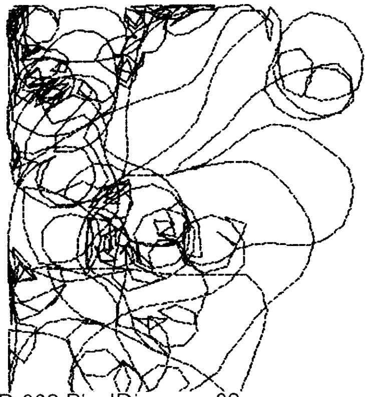
D-002-VectorDiagram-02



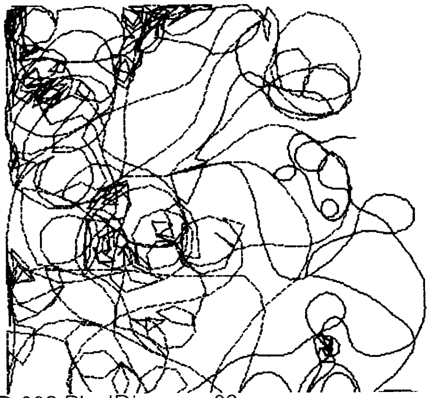
D-002-VectorDiagram-03



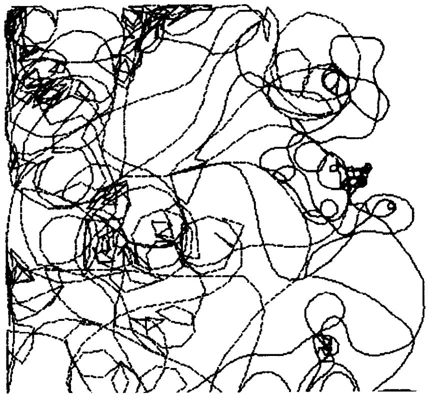
D-002-PixelDiagram-01



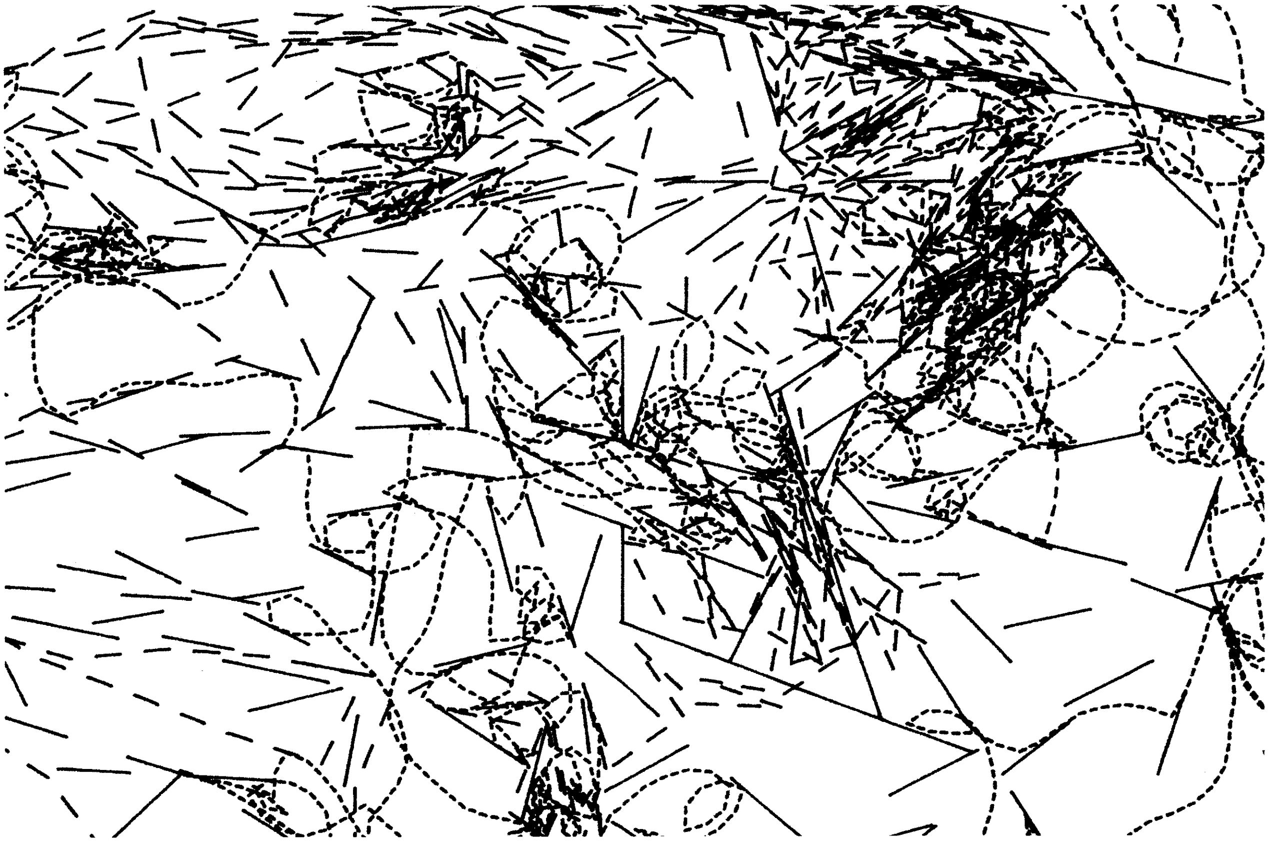
D-002-PixelDiagram-02



D-002-PixelDiagram-03

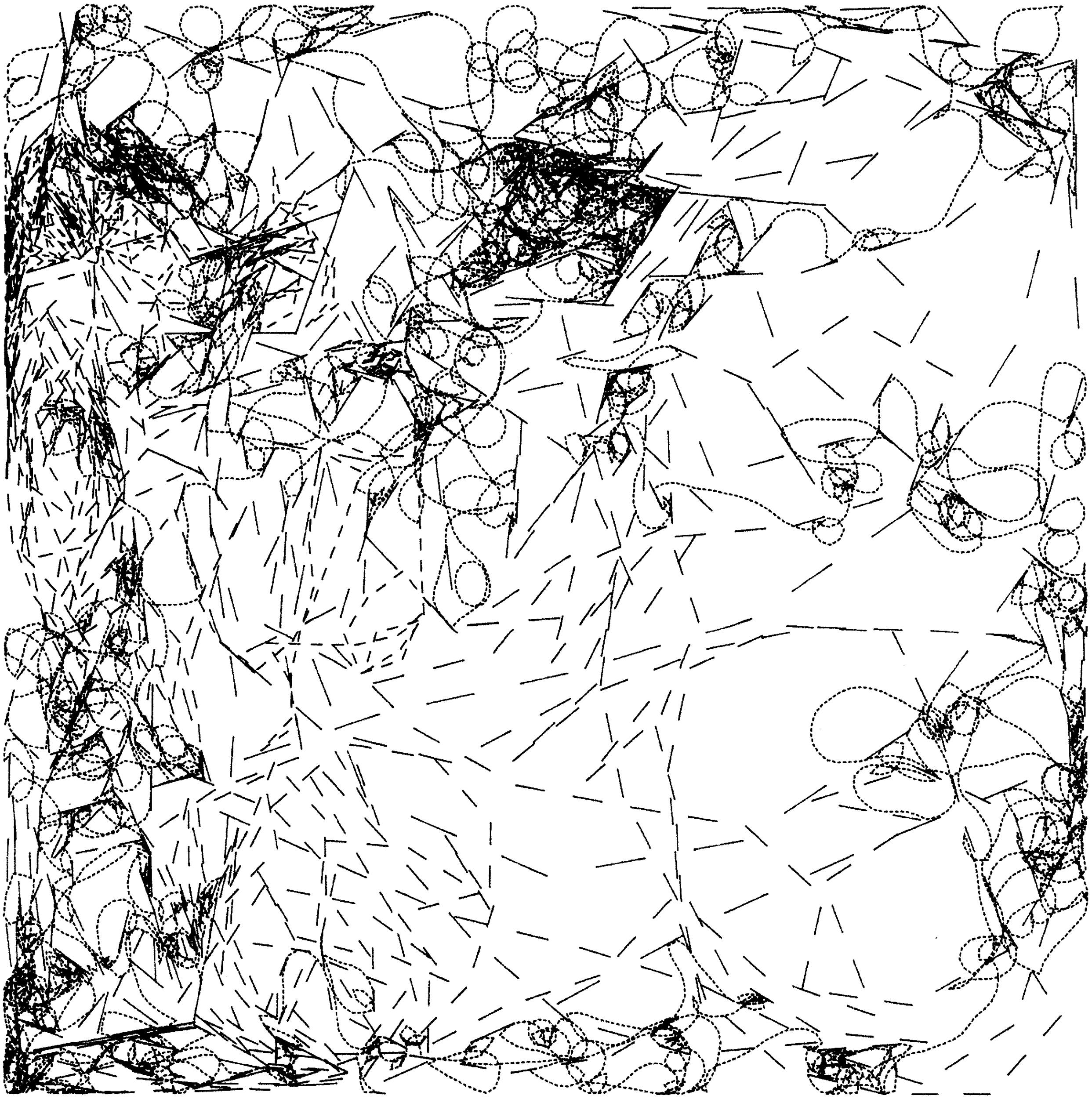


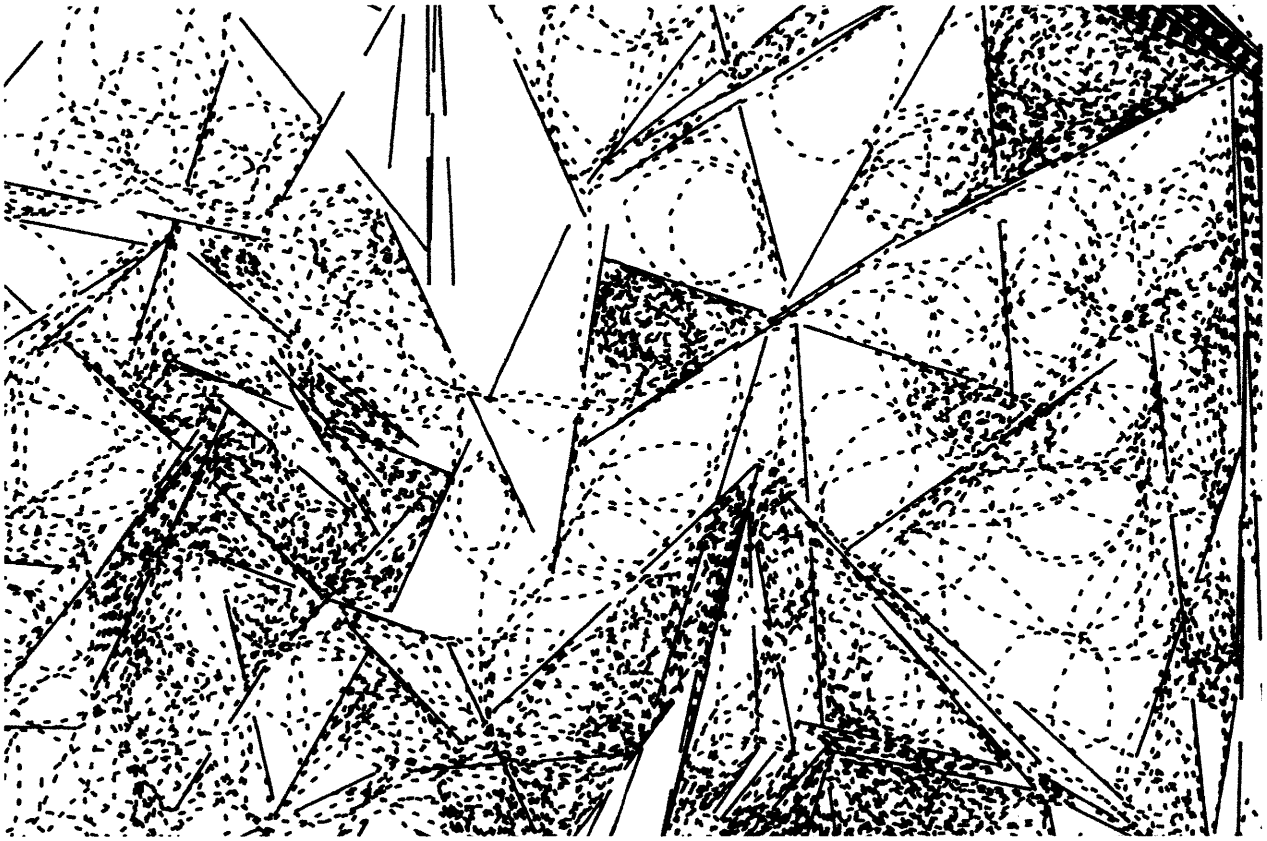
D-002-PixelDiagram-04



D-002-001: Curvy Walker

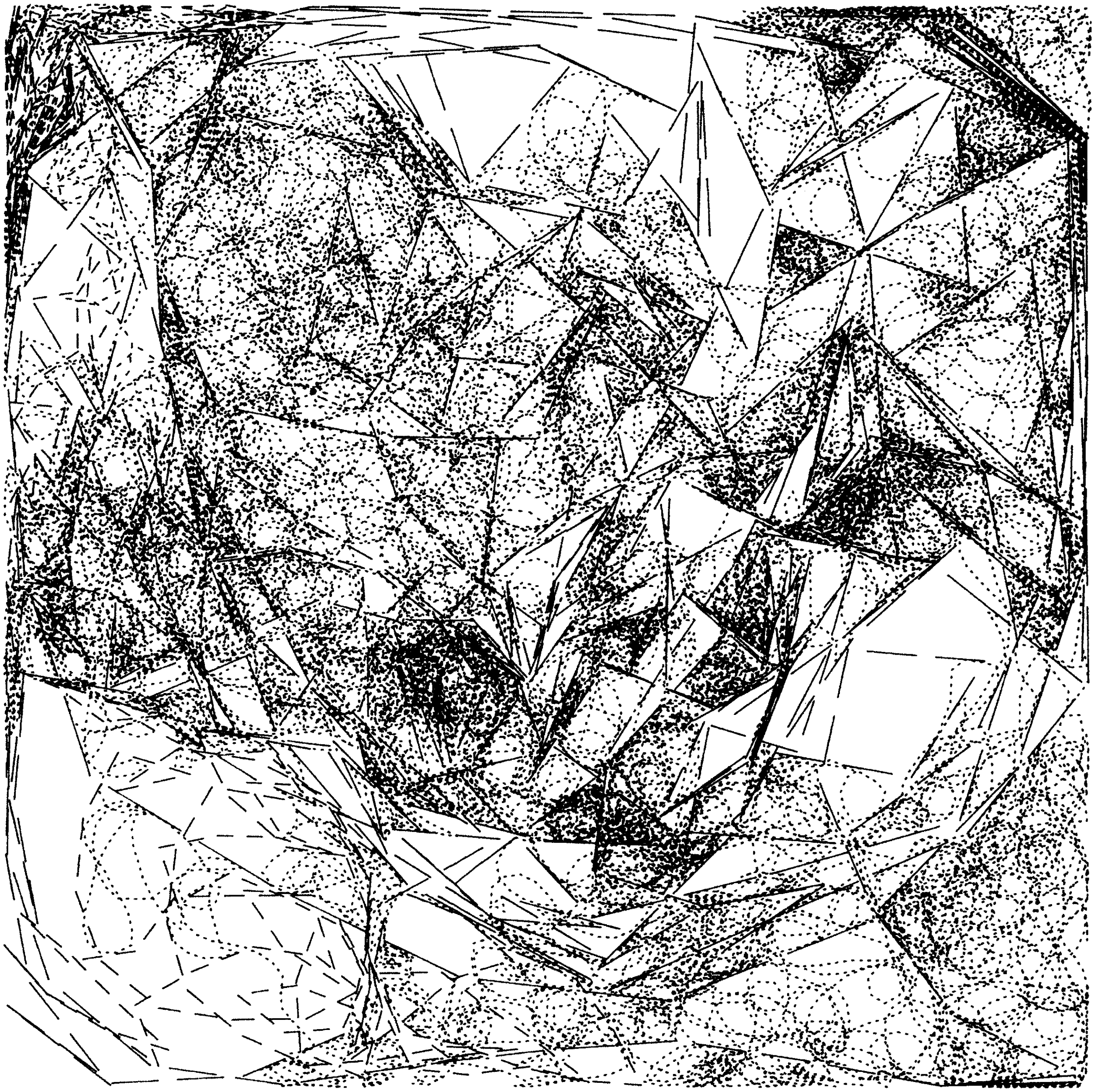
Method 'D' involves algorithms that treat the paper as space and ink as object. "Line" remains open to interpretation. In series D-002, marks are made by a particle "walking" in a random but generally curving path within a boundary on the page. The invisible particle leaves a dashed trail, which it is never allowed to make contact with. If the particle is nearing collision with its tail, its angular acceleration increases–it steers out of the way. The gaps between the dashes are openings, where the particle may move through. Over the course of the drawing the particle speed decreases, (causing the curves to be tighter and smoother) the proportion of dash to gap decreases. Run 001 uses a new black pen and creates a square boundary.



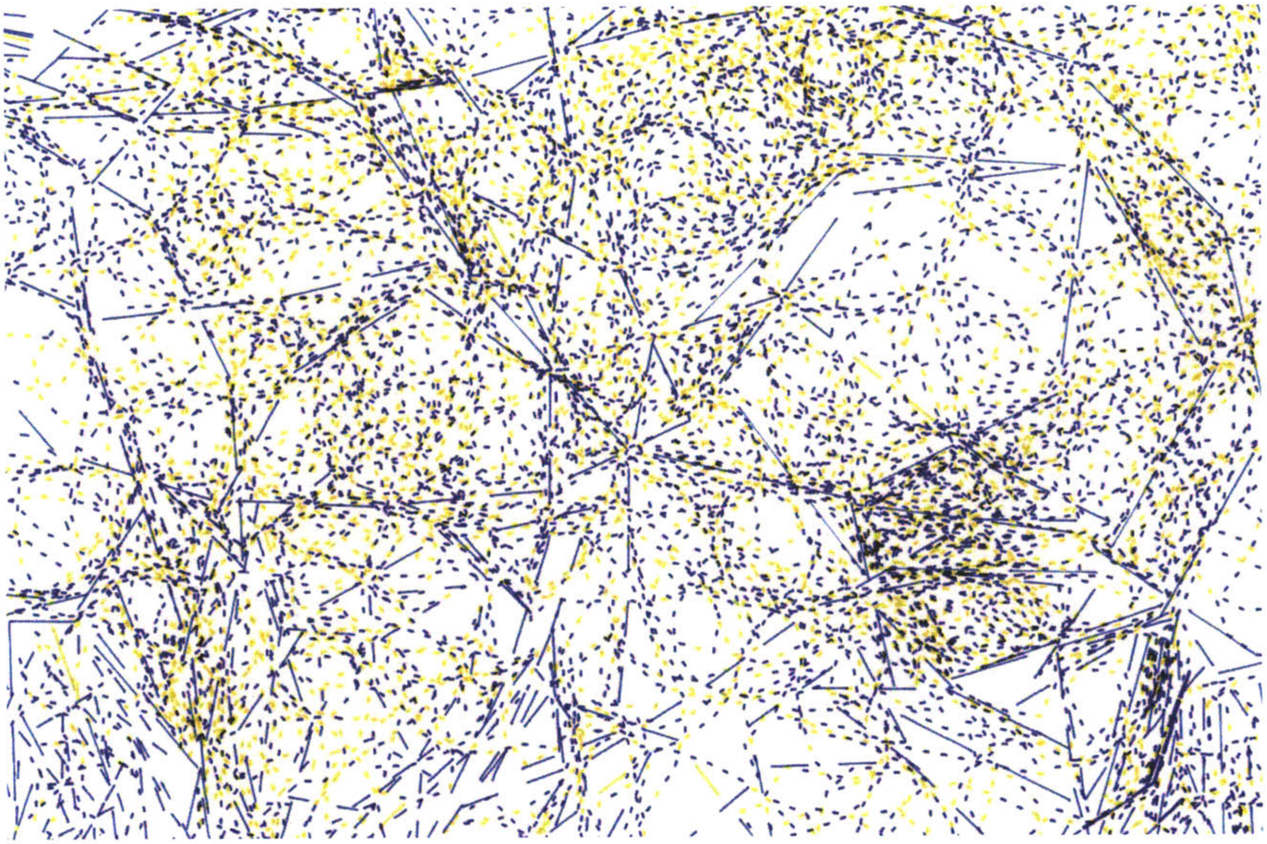


D-002-003: Patient Curvy walker

Method 'D' involves algorithms that treat the paper as space and ink as object. "Line" remains open to interpretation. Method 'D' involves algorithms that treat the paper as space and ink as object. "Line" remains open to interpretation. In series D-002, marks are made by a particle "walking" in a random but generally curving path within a boundary on the page. The invisible particle leaves a dashed trail, which it is never allowed to make contact with. If the particle is nearing collision with its trail, its angular acceleration increases–it steers out of the way. The gaps between the dashes are openings, where the particle may move through. Over the course of the drawing, the particle speed decreases, (causing the curves to be tighter and smoother) the proportion of dash to gap decreases. Run 002 uses an almost new black pen within a square boundary. Relative to run 001, the rate of speed decrease is lower and the path longer.

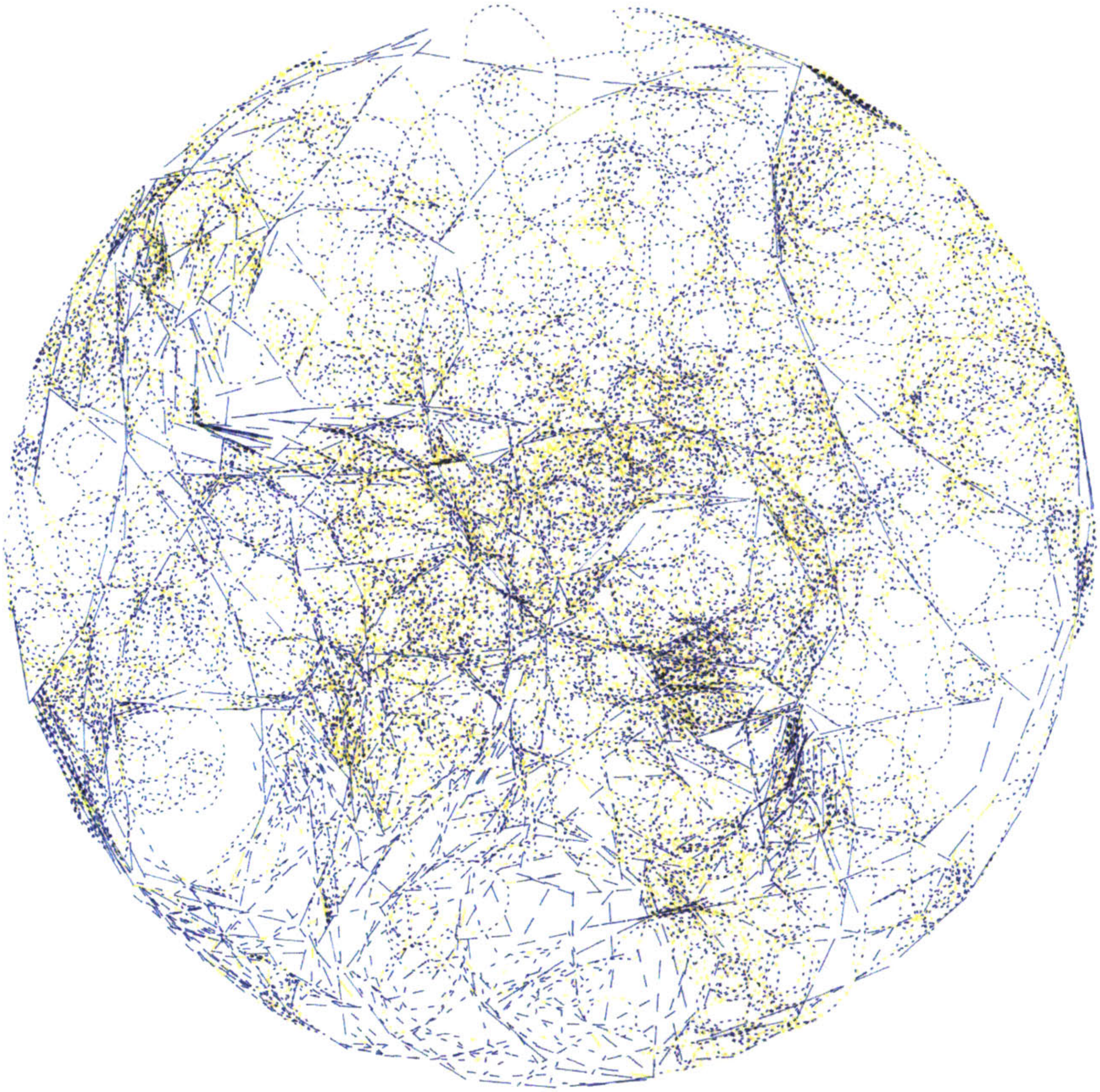


D-002-003

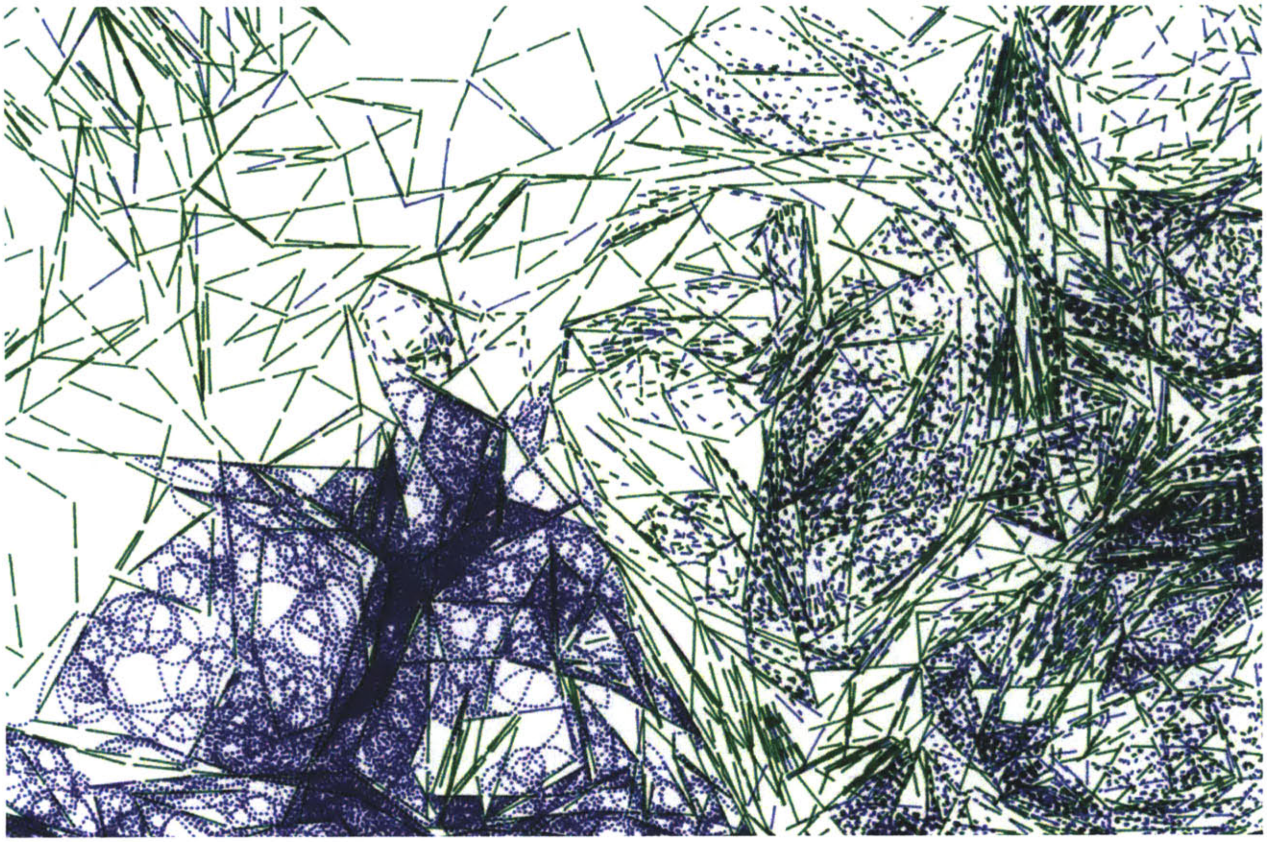


D-002-006: Walking From Blue to Yellow

Method 'D' involves algorithms that treat the paper as space and ink as object. "Line" remains open to interpretation. In series D-002, marks are made by a particle "walking" in a random but generally curving path within a boundary on the page. The invisible particle leaves a dashed trail, which it is never allowed to make contact with. If the particle is nearing collision with its tail, its angular acceleration increases–it steers out of the way. The gaps between the dashes are openings, where the particle may move through. Over the course of the drawing, the particle speed decreases, (causing the curves to be tighter and smoother) the proportion of dash to gap decreases. Run 008 adds the variable of a second ink color to further emphasize the axis of time in the making of the drawing. Later marks have an increasing likelihood of being made with the yellow, as opposed to blue, pen. Also, run 006 uses a circular boundary frame.

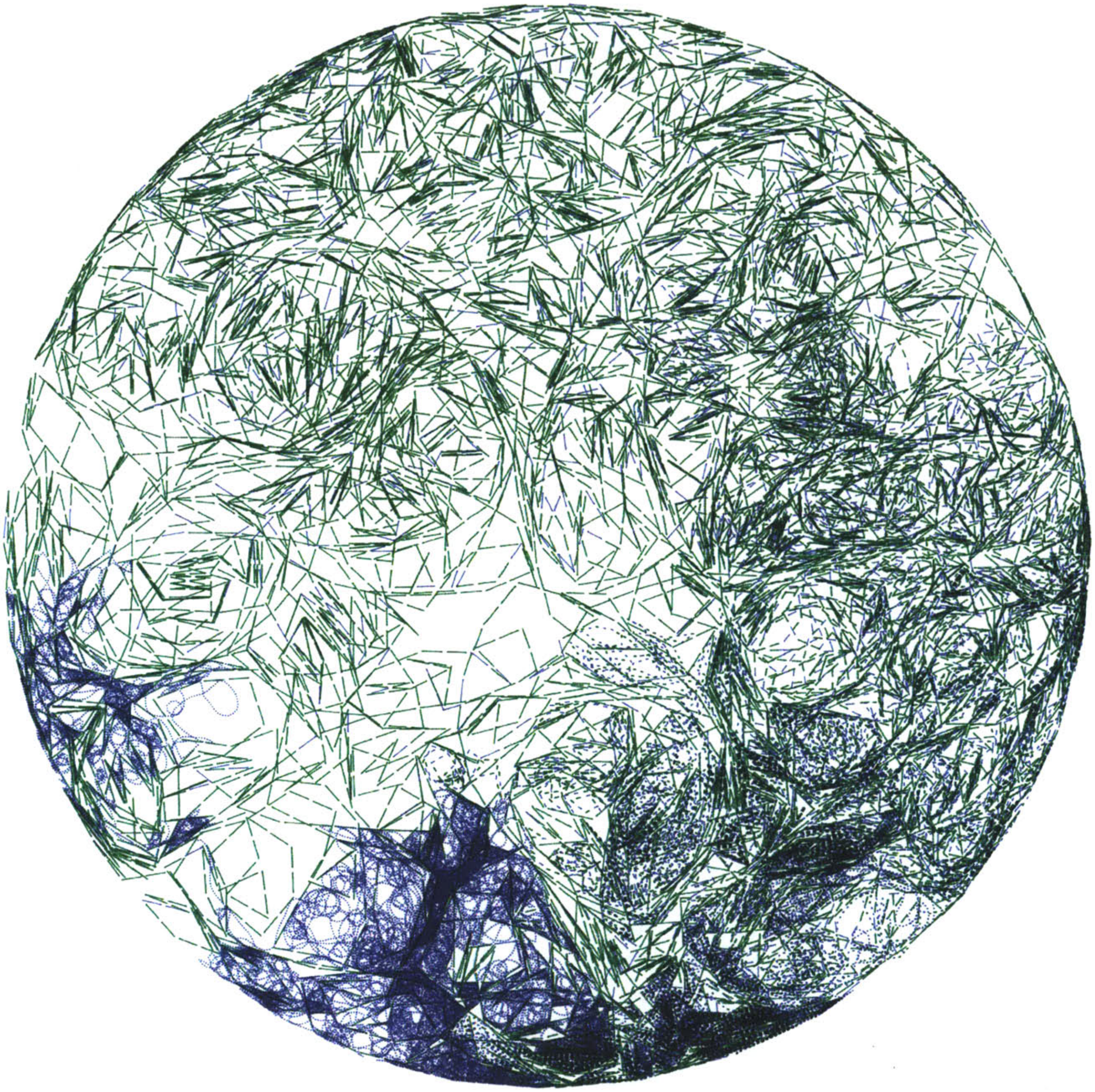


D-002-006

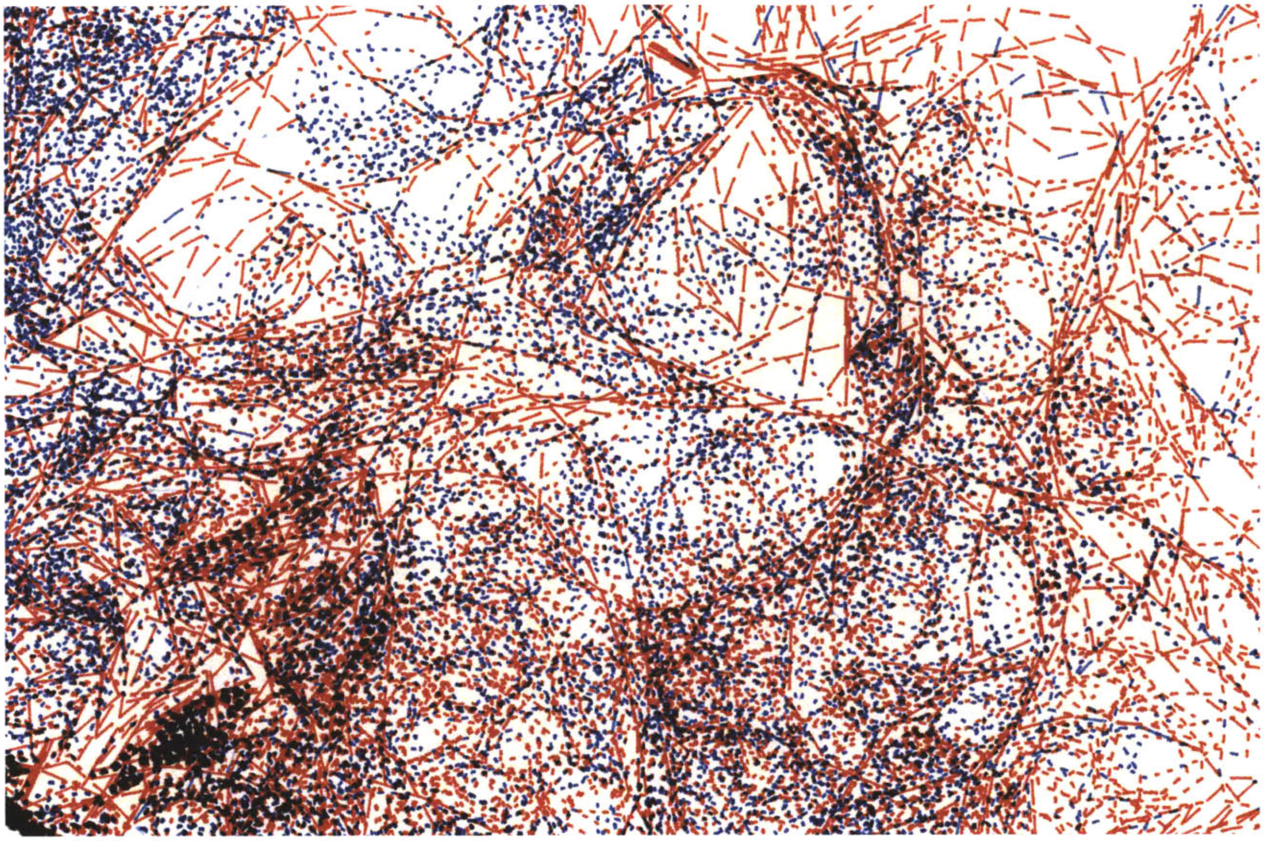


D-002-007: Walking From Green to Blue

Method 'D' involves algorithms that treat the paper as space and ink as object. "Line" remains open to interpretation. In series D-002, marks are made by a particle "walking" in a random but generally curving path within a boundary on the page. The invisible particle leaves a dashed trail, which it is never allowed to make contact with. If the particle is nearing collision with its tail, its angular acceleration increases–it steers out of the way. The gaps between the dashes are openings, where the particle may move through. Over the course of the drawing, the particle speed decreases, (causing the curves to be tighter and smoother) the proportion of dash to gap decreases. Run 007 adds the variable of a second ink color to further emphasize the axis of time in the making of the drawing. Later marks have an increasing likelihood of being made with the blue, as opposed to green, pen. Also, run 007 uses a circular boundary frame and the start speed is slower relative to previous runs.

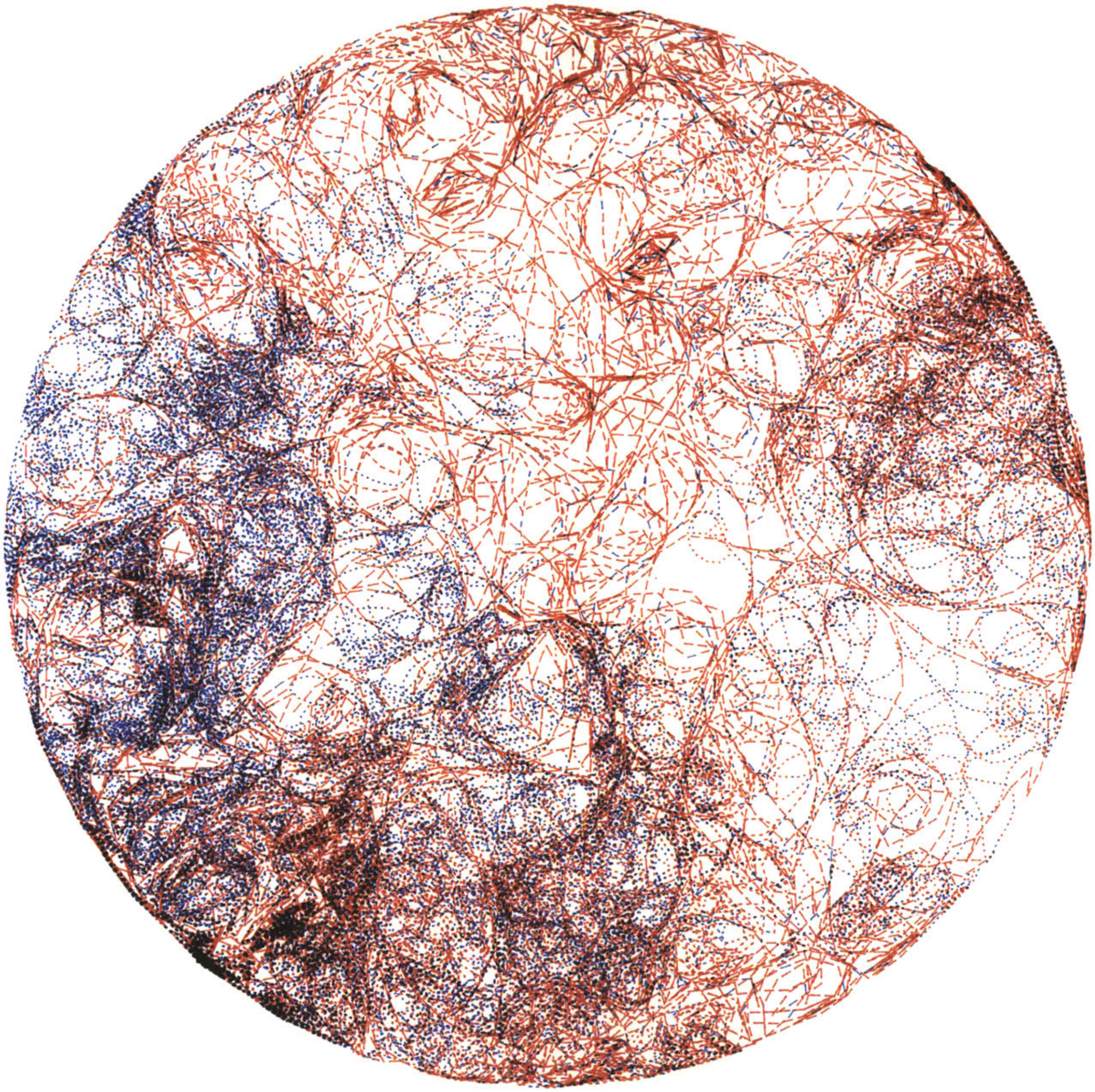


D-002-007

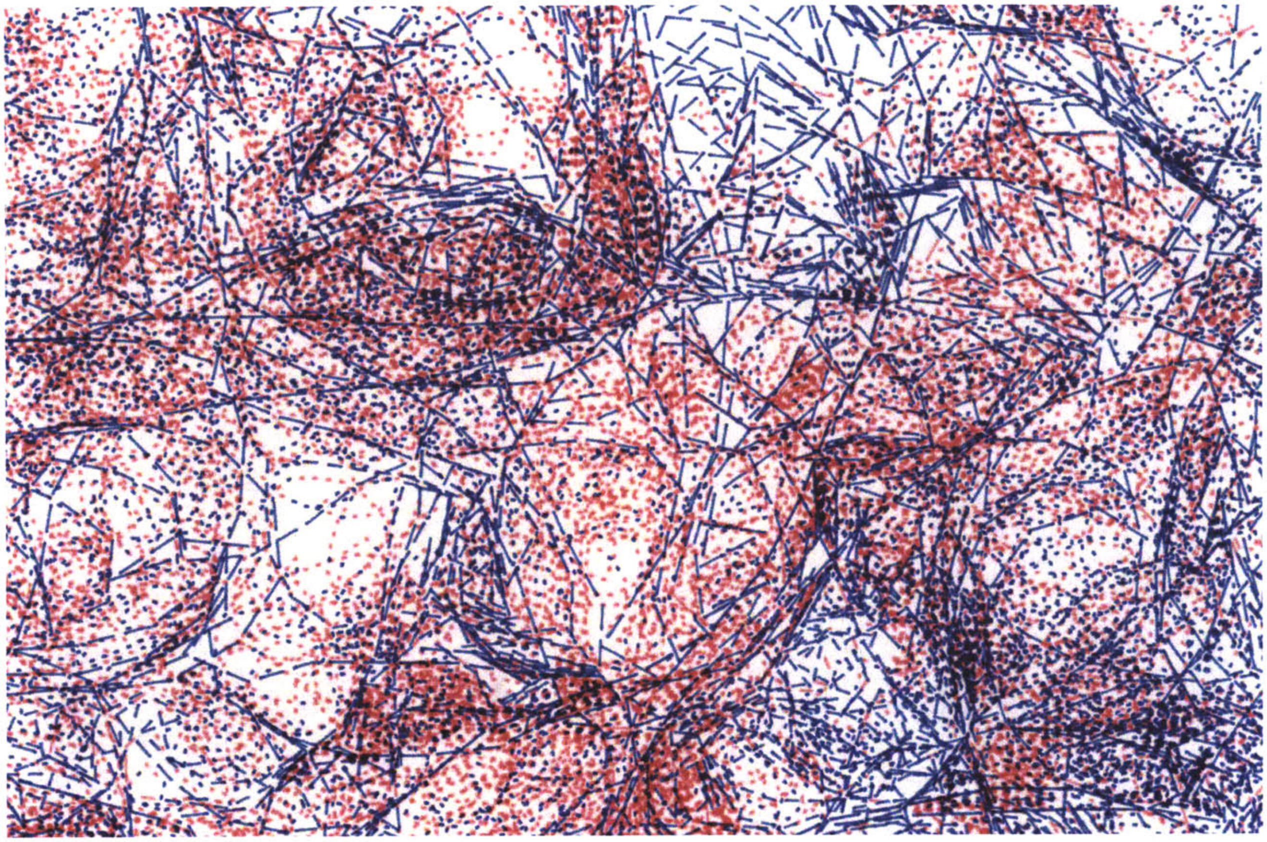


D-002-008: Walking From Red to Blue

Method 'D' involves algorithms that treat the paper as space and ink as object. "Line" remains open to interpretation. In series D-002, marks are made by a particle "walking" in a random but generally curving path within a boundary on the page. The invisible particle leaves a dashed trail, which it is never allowed to make contact with. If the particle is nearing collision with its tail, its angular acceleration increases–it steers out of the way. The gaps between the dashes are openings, where the particle may move through. Over the course of the drawing the particle speed decreases, (causing the curves to be tighter and smoother) the proportion of dash to gap decreases. Run 008 adds the variable of a second ink color to further emphasize the axis of time in the making of the drawing. Later marks have an increasing likelihood of being made with the blue, as opposed to red, pen. Also run 008 uses a circular boundary frame.

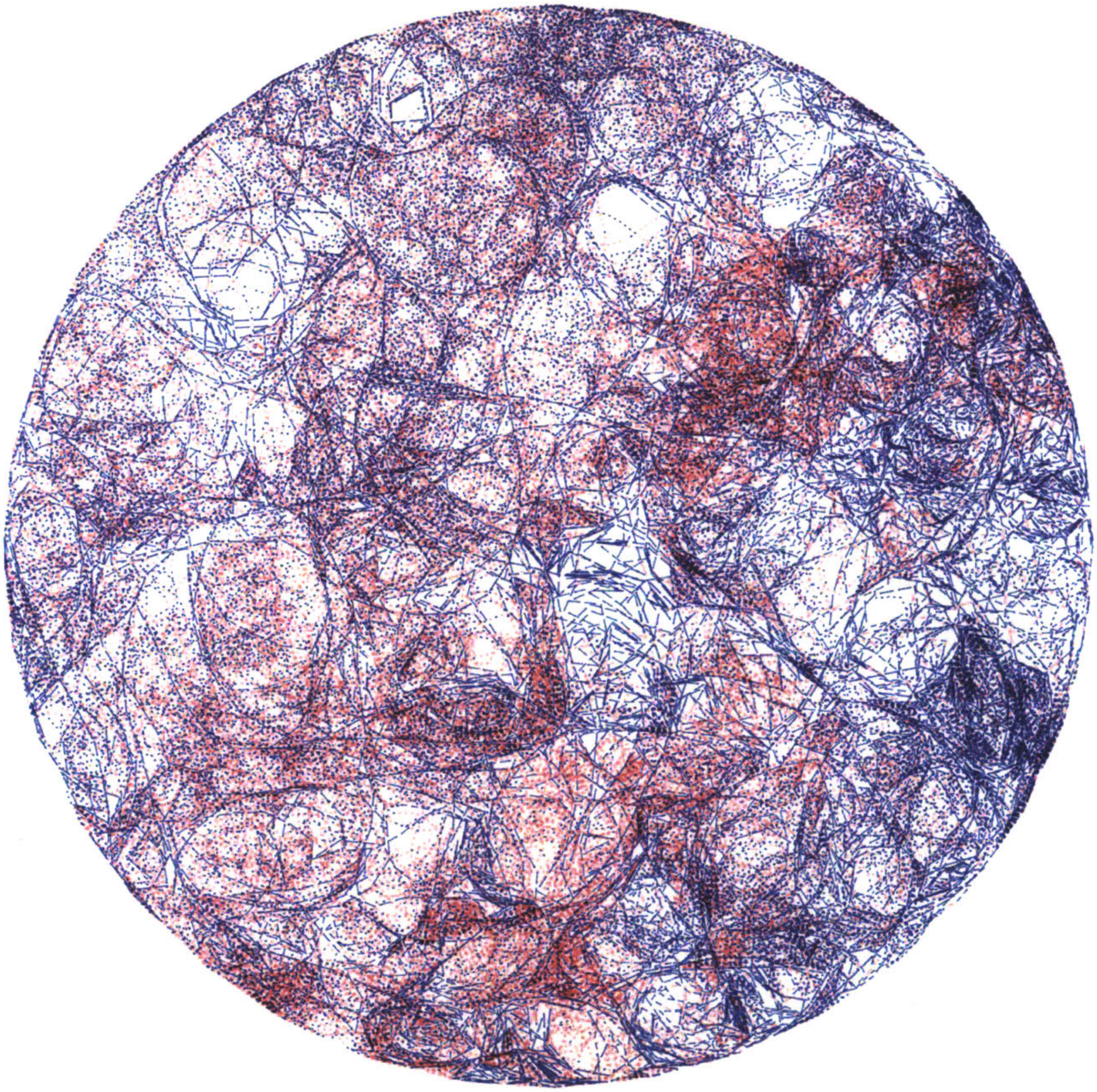


D-002-008



D-002-009: Walking From Blue to Magenta

Method 'D' involves algorithms that treat the paper as space and ink as object. "Line" remains open to interpretation. In series D-002, marks are made by a particle "walking" in a random but generally curving path within a boundary on the page. The invisible particle leaves a dashed trail, which it is never allowed to make contact with. If the particle is nearing collision with its tail, its angular acceleration increases–it steers out of the way. The gaps between the dashes are openings, where the particle may move through. Over the course of the drawing the particle speed decreases, (causing the curves to be tighter and smoother) the proportion of dash to gap decreases. Run 009 adds the variable of a second ink color to further emphasize the axis of time in the making of the drawing. Later marks have an increasing likelihood of being made with the blue, as opposed to red, pen. Also run 009 uses a circular boundary frame and is the result of 8 hours of calculation.



D-002-009

Method G

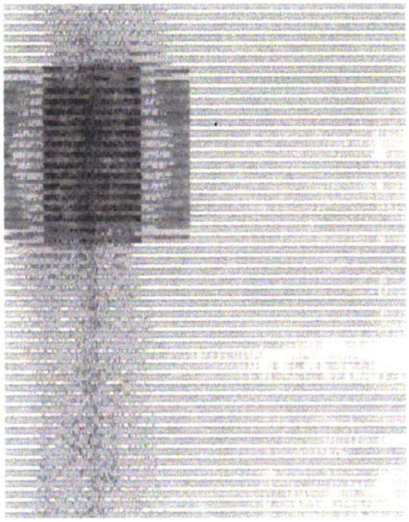
Method G uses a lattice to represent the drawing. At each lattice point, six bits are stored. These bits are translated into parameters that determine pen strokes in the drawing. On the algorithmic temporal axis, the initial state is composed. In discrete steps forward in time, the bits shift in each of six directions based on simple rules designed to simulate fluid diffusion.

Series G-001

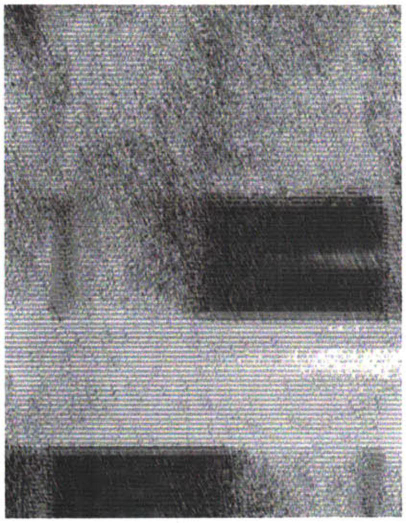
A a moment of a changing 450x200 unit lattice is mapped onto paper by tallying the "on" bits at each intersection. This value determines the length of a vertical line.

Series G-002

A moment of a changing 450x200 unit lattice is mapped onto paper by translating the 6-bit value at each intersection into length and angle information of each stroke.

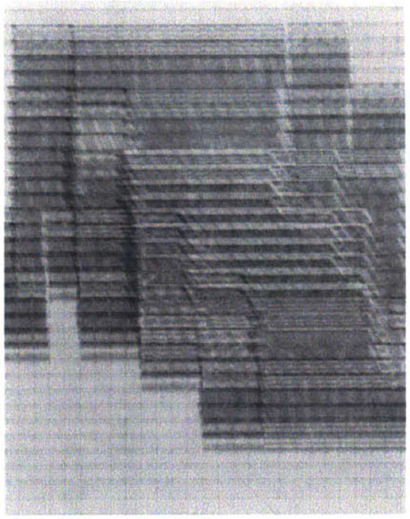


D-001-036

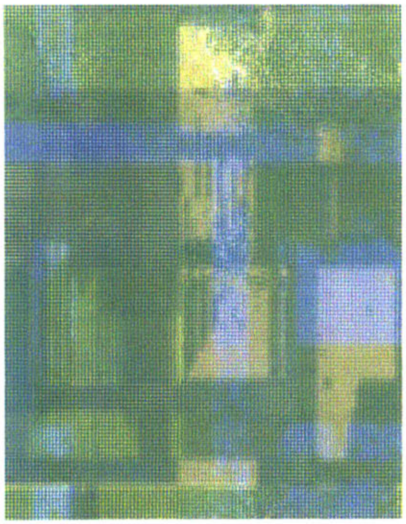


D-002-380

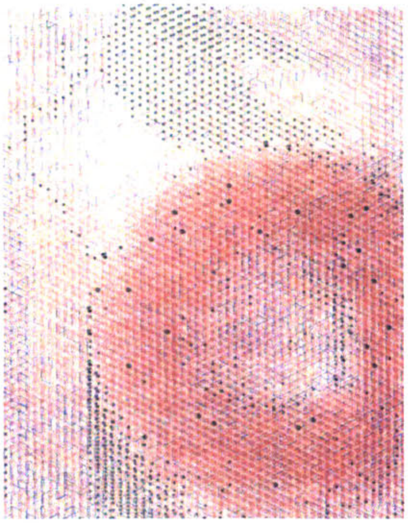
G-002-682



G-003-019



G-004-205



G-005-190

Series G-003

A moment of a changing 450x200 unit lattice is mapped onto paper by translating the 6-bit value at each intersection into length and angle information of sets of strokes.

Series G-004

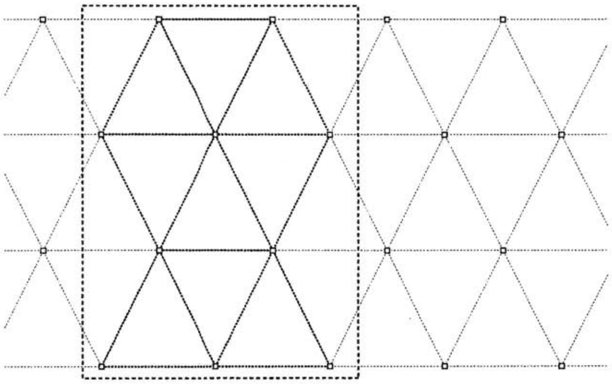
Multiple moments of a changing 100x100 unit lattice are simultaneously drawn. The bits at each intersection determine dimensions and offset of a rectangle.

Series G-005

Multiple moments of a changing 100x200 unit lattice are simultaneously drawn. Circles mark vertex points where certain rule was applied. Each bit corresponds to a single line.

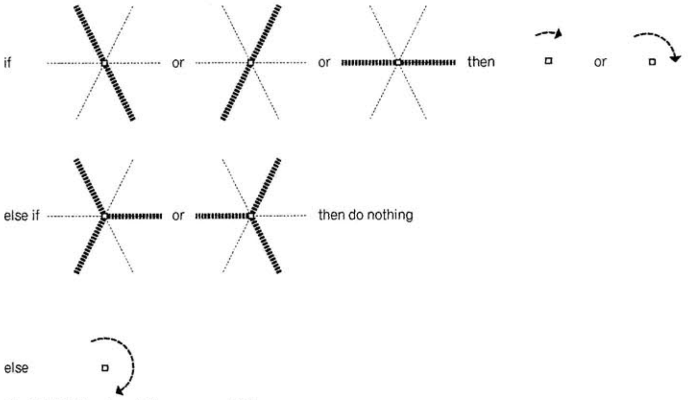
Series G-001

The lattice structure and behavior are implemented based on the algorithm published by Neil Gershenfeld.[1] Each intersection on the lattice is represented by six bits, which correspond to on or off values for each link to a neighbor point. The values are asserted based on a simple geometry. All bits within a closed rectangle are set to "on," while all others are set to "off." (G-001-VectorDiagram-01)



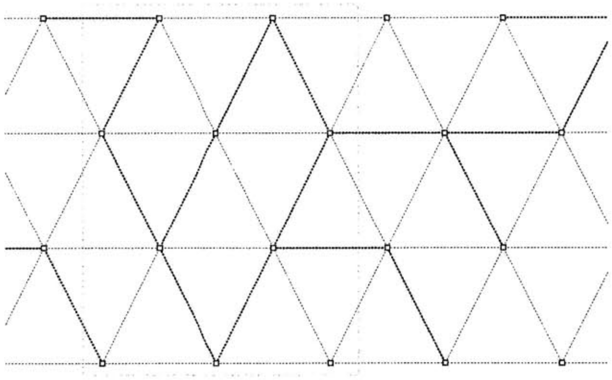
G-001-VectorDiagram-01

At each algorithmic step, a new 6-bit configuration is calculated based on the previous state. Two special cases represent collisions, in which case the bit values "bounce" off one another. In all other cases, the values "flow" through the intersection point (G-001-VectorDiagram-02)

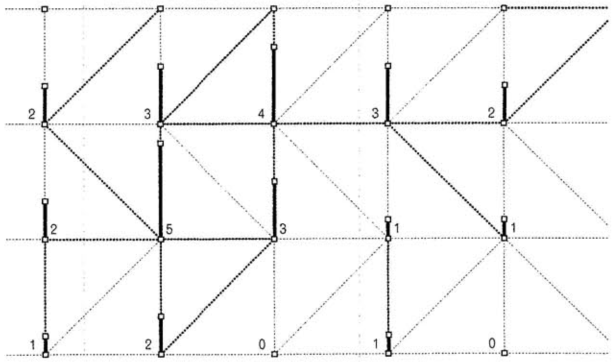


G-001-VectorDiagram-02

Each run is a paused moment, which references the state of the entire lattice. (G-001-VectorDiagram-03) The lattice is skewed to a regular Cartesian grid and lines are drawn from a corresponding grid point on paper to a vertically offset point. The length of that line is calculated by tallying the on bits at each lattice intersection. (G-001-VectorDiagram-04)



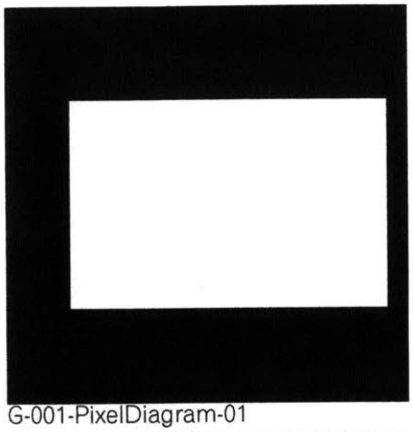
G-001-VectorDiagram-03

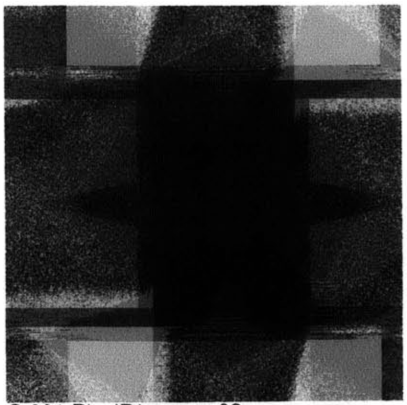


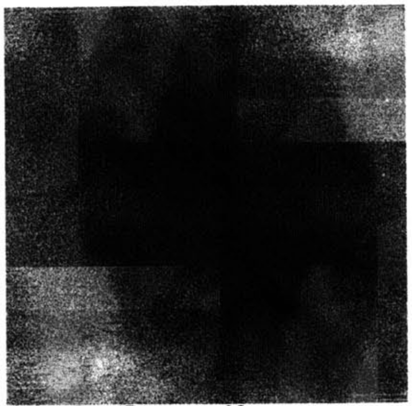
G-001-VectorDiagram-04

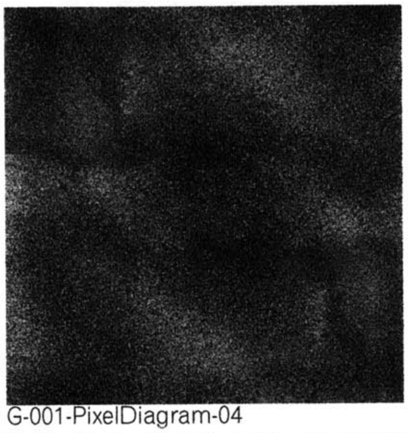
[1] Gershenfeld, The Nature of Mathematical Modeling.102

Two forms of pixel representations document the computational process. The first (G-001-PixelDiagram-01, G-001-PixelDiagram-02, G-001-PixelDiagram-03, and G-001-PixelDiagram-04) associates each lattice point with one pixel. The grayscale value of each pixel is associated with the number of on bits at that point. If all six bits are on the pixel is rendered white. If all are off the pixel is rendered black. Tallies between 1 and 5 are intermediate grays.

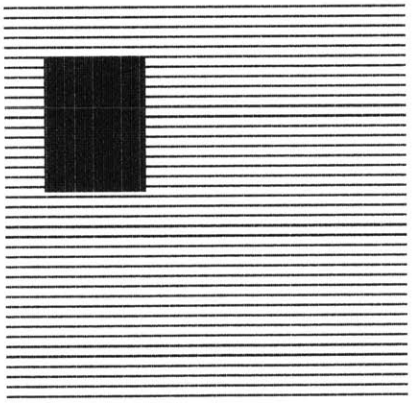

G-001-PixelDiagram-01

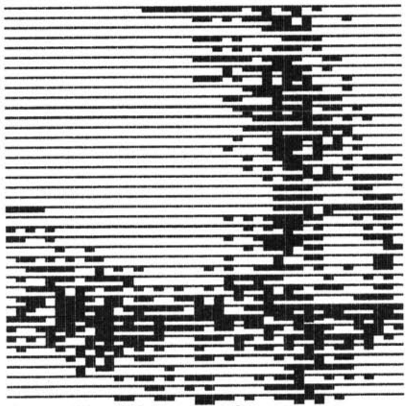

G-001-PixelDiagram-02

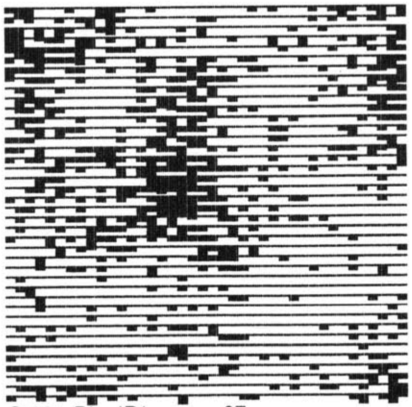

G-001-PixelDiagram-03

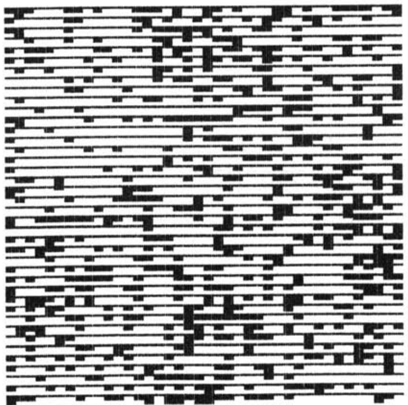

G-001-PixelDiagram-04

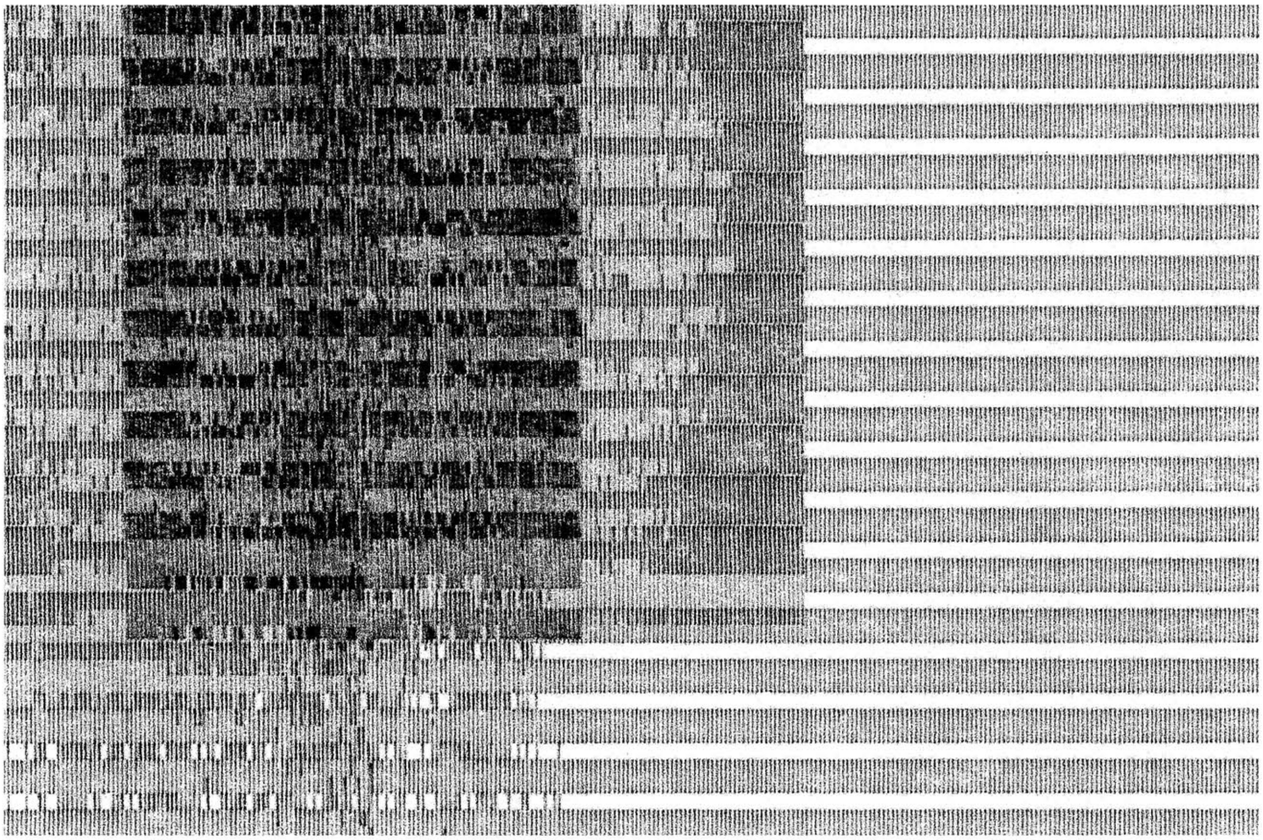
The second form of pixel representation (G-001-PixelDiagram-05, G-001-PixelDiagram-06, G-001-PixelDiagram-07, and G-001-PixelDiagram-08) associates the image with the paper. A lattice with resolution lower than that of the image is used and lines are rendered in manner that crudely approximates the lines on paper.


G-001-PixelDiagram-05


G-001-PixelDiagram-06

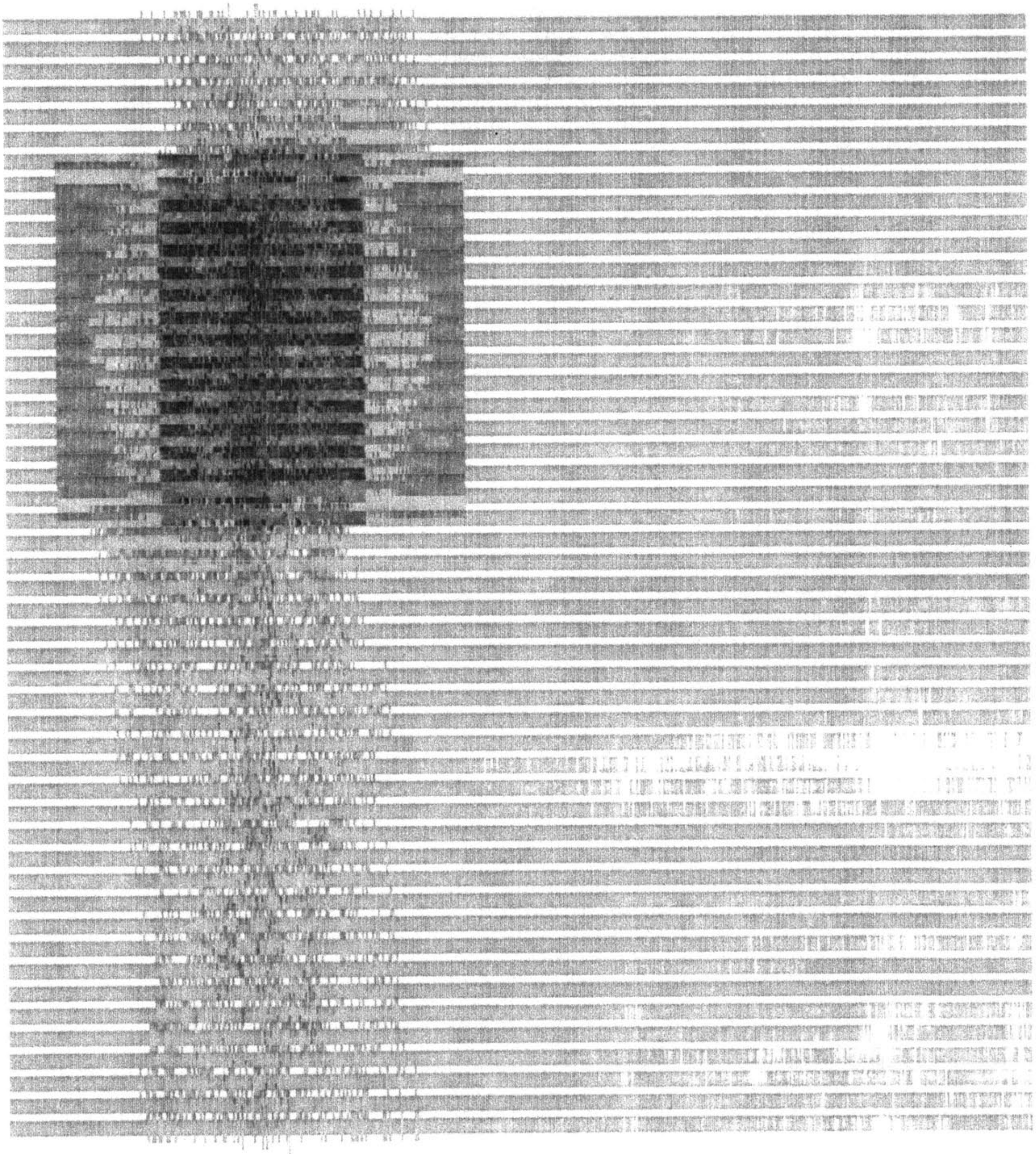

G-001-PixelDiagram-07


G-001-PixelDiagram-08



G-001-036: Recently, a Rectangle; Drawn With a Worn Pen

Method 'G' uses a lattice to represent the drawing. At each lattice point, six bits are stored. These bits are translated into parameters that determine pen strokes in the drawing. On the algorithmic temporal axis, the initial state is composed. In discrete steps forward in time, the bits shift in each of six directions based on simple rules designed to simulate fluid diffusion. Series 001 uses a 450-unit wide by 200-unit high triangular lattice. These bits are summed to compute the length of each vertical stroke on paper. The system will eventually will move towards homogeneity, but after only a few dozen steps, (36 until the moment that is captured here) the initial state, a rectangular area with all "on" values, has been merely degraded. Run 036 is drawn with a worn (to the extent of near failure) pen to allow overlapping lines to register.



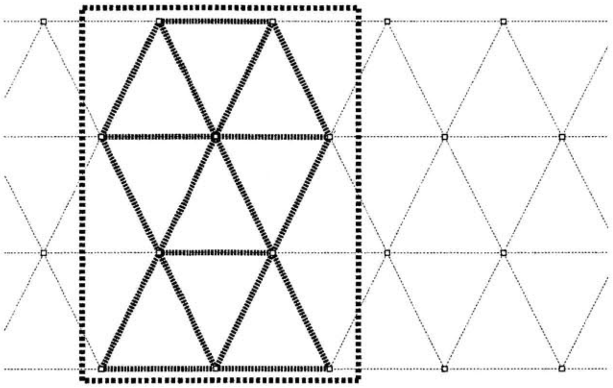
G-001-036

Series G-002

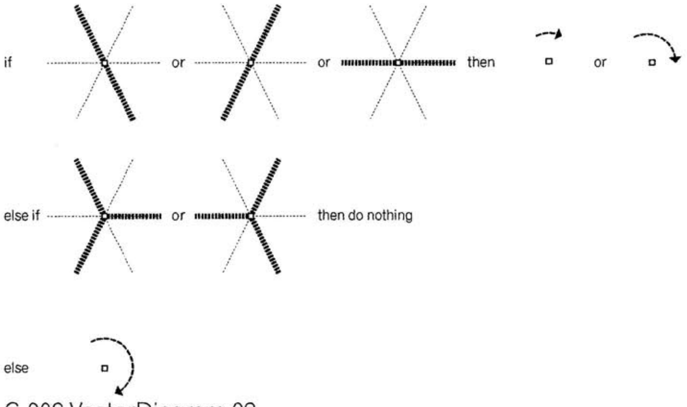
The lattice structure and behavior are implemented based on the algorithm published by Neil Gershenfeld.[2] Each intersection on the lattice is represented by six bits, which correspond to on or off values for each link to a neighbor point. The values are asserted based on a simple geometry. All bits within a closed rectangle are set to "on," while all others are set to "off." (G-002-VectorDiagram-01)

At each algorithmic step, a new 6-bit configuration is calculated based on the previous state. Two special cases represent collisions, in which case the bit values "bounce" off one another. In all other cases, the values "flow" through the intersection point (G-002-VectorDiagram-02)

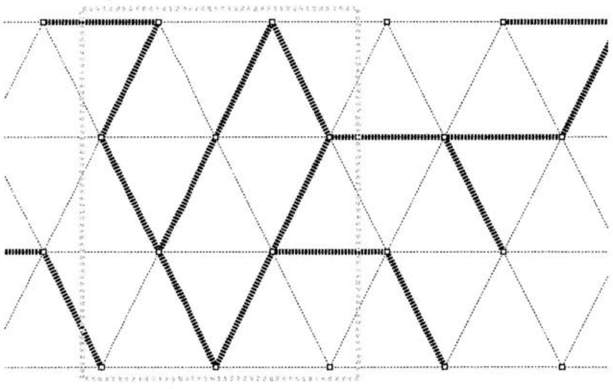
Each run is a paused moment, which references the state of the entire lattice. (G-002-VectorDiagram-03) The lattice is skewed to a regular Cartesian grid and lines are drawn from a corresponding grid point to an offset point. The length of and angle that line is calculated by tallying the on bits at each lattice intersection. Fewer on bits leads to a shorter and more vertical mark. If all six bits are on the line is at it's maximum possible length and angled 45 degrees from vertical. (G-002-VectorDiagram-04)



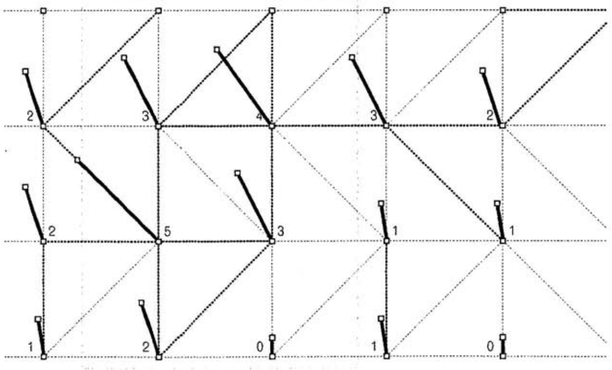
G-002-VectorDiagram-01



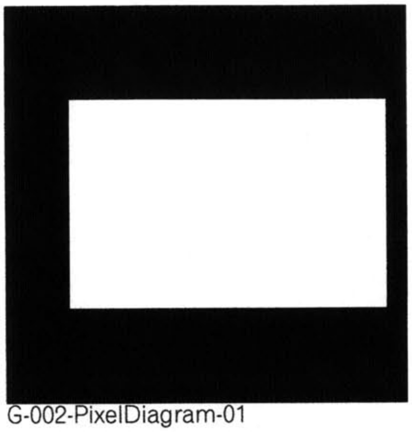
G-002-VectorDiagram-02

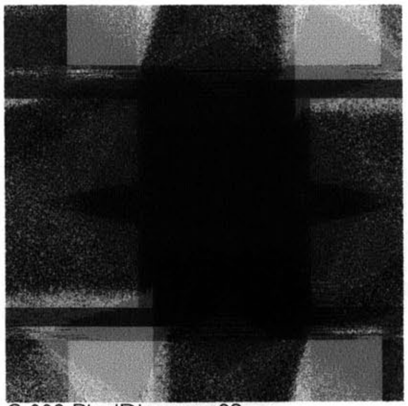


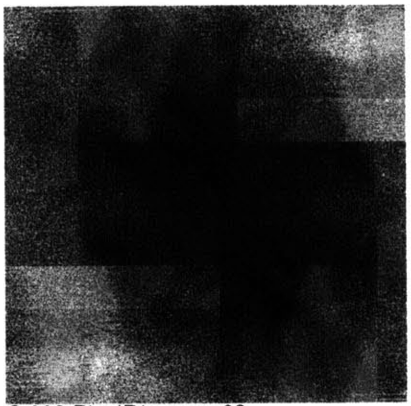
G-002-VectorDiagram-03

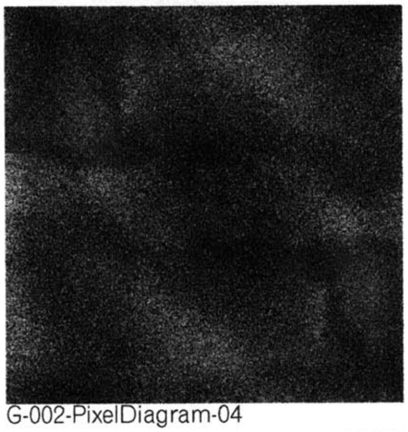


G-002-VectorDiagram-04

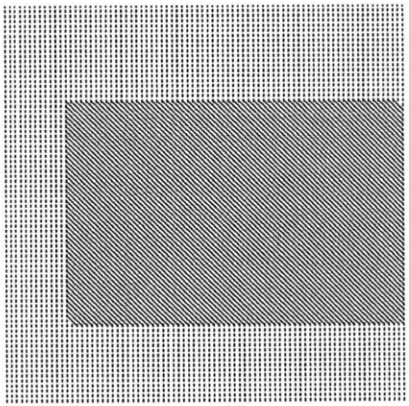

G-002-PixelDiagram-01

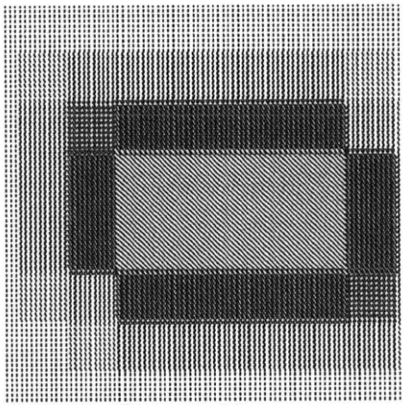

G-002-PixelDiagram-02

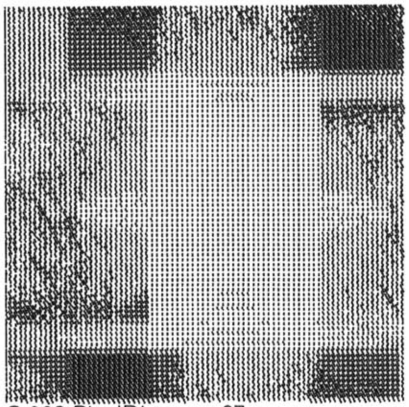

G-002-PixelDiagram-03

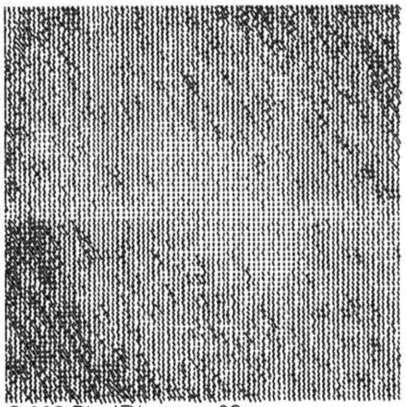

G-002-PixelDiagram-04

Two forms of pixel representations document the computational process. The first (G-002-PixelDiagram-01, G-002-PixelDiagram-02, G-002-PixelDiagram-03, and G-002-PixelDiagram-04) associates each lattice point with one pixel. The grayscale value of each pixel is associated with the number of on bits at that point. If all six bits are on the pixel is rendered white. If all are off the pixel is rendered black. Tallies between 1 and 5 are intermediate grays.


G-002-PixelDiagram-05

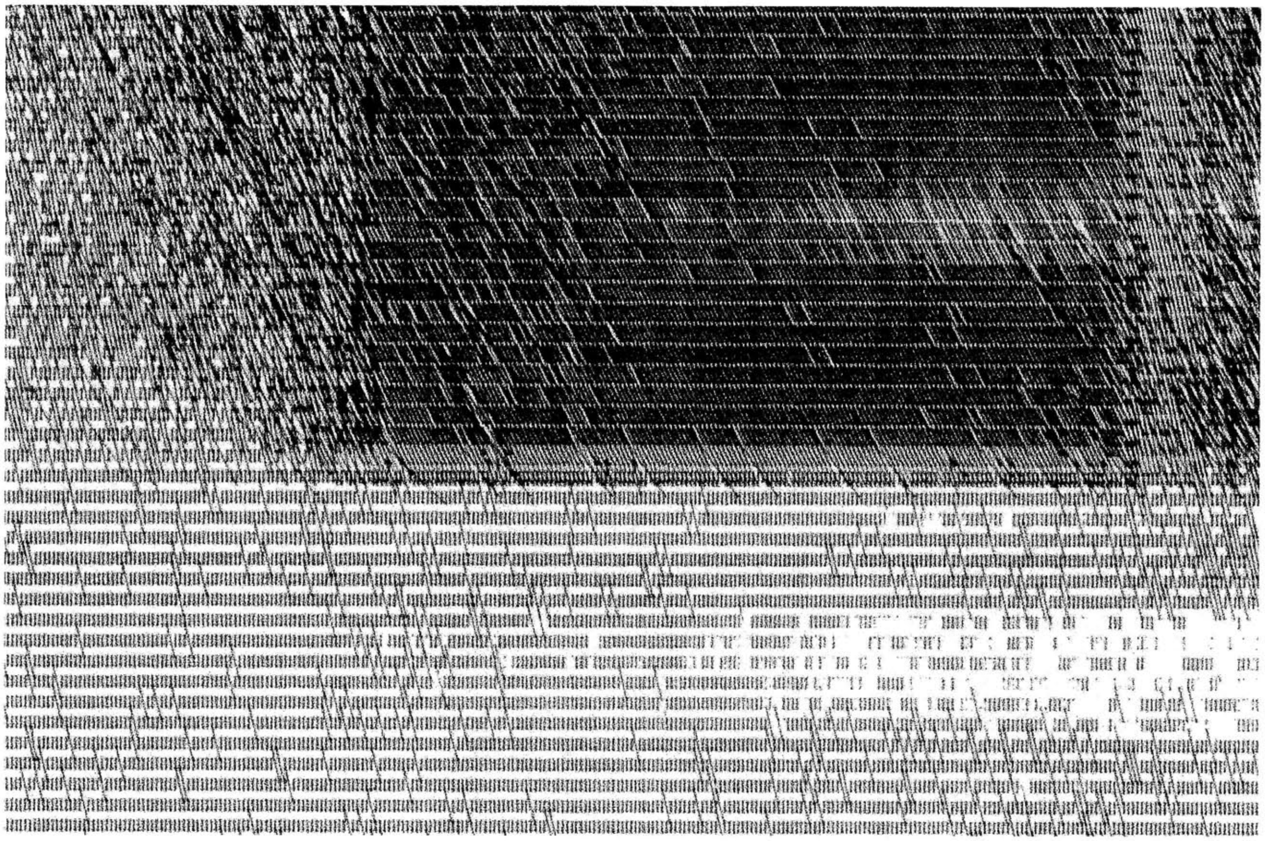

G-002-PixelDiagram-06


G-002-PixelDiagram-07


G-002-PixelDiagram-08

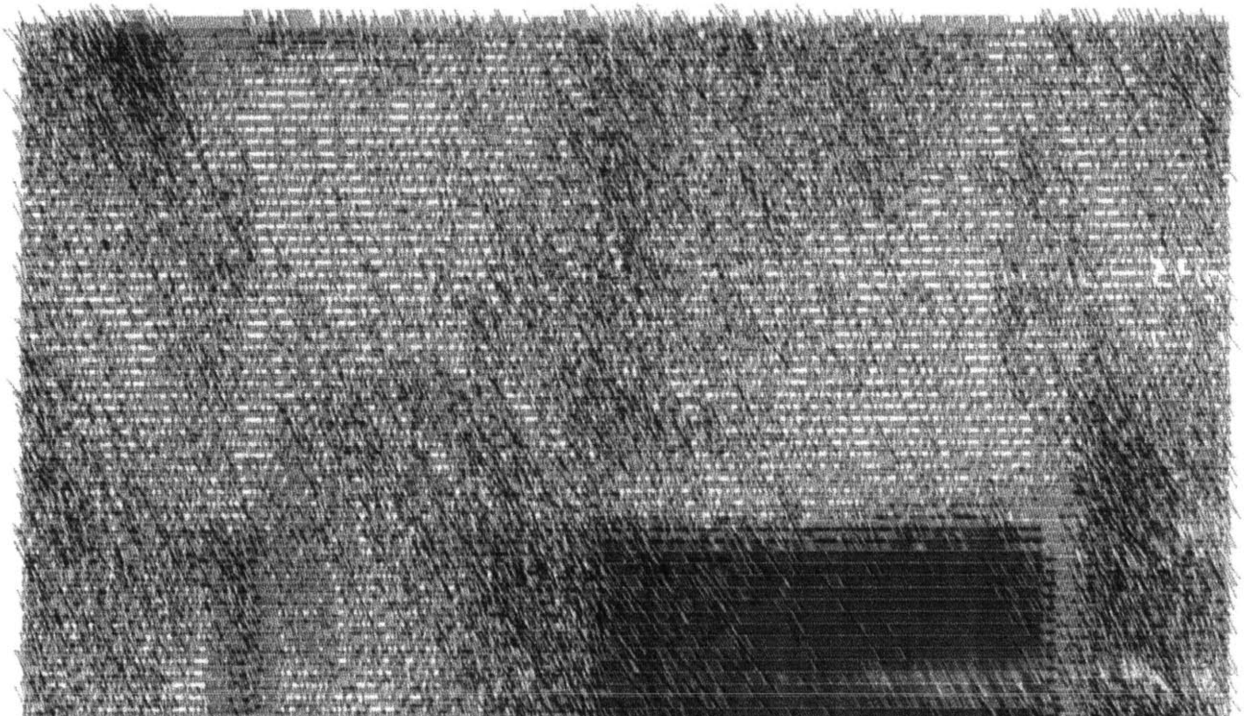
The second form of pixel representation (G-002-PixelDiagram-05, G-002-PixelDiagram-06, G-002-PixelDiagram-07, and G-002-PixelDiagram-08) associates the image with the paper. A lattice with resolution lower than that of the image is used and lines are rendered in manner that crudely approximates the lines on paper.

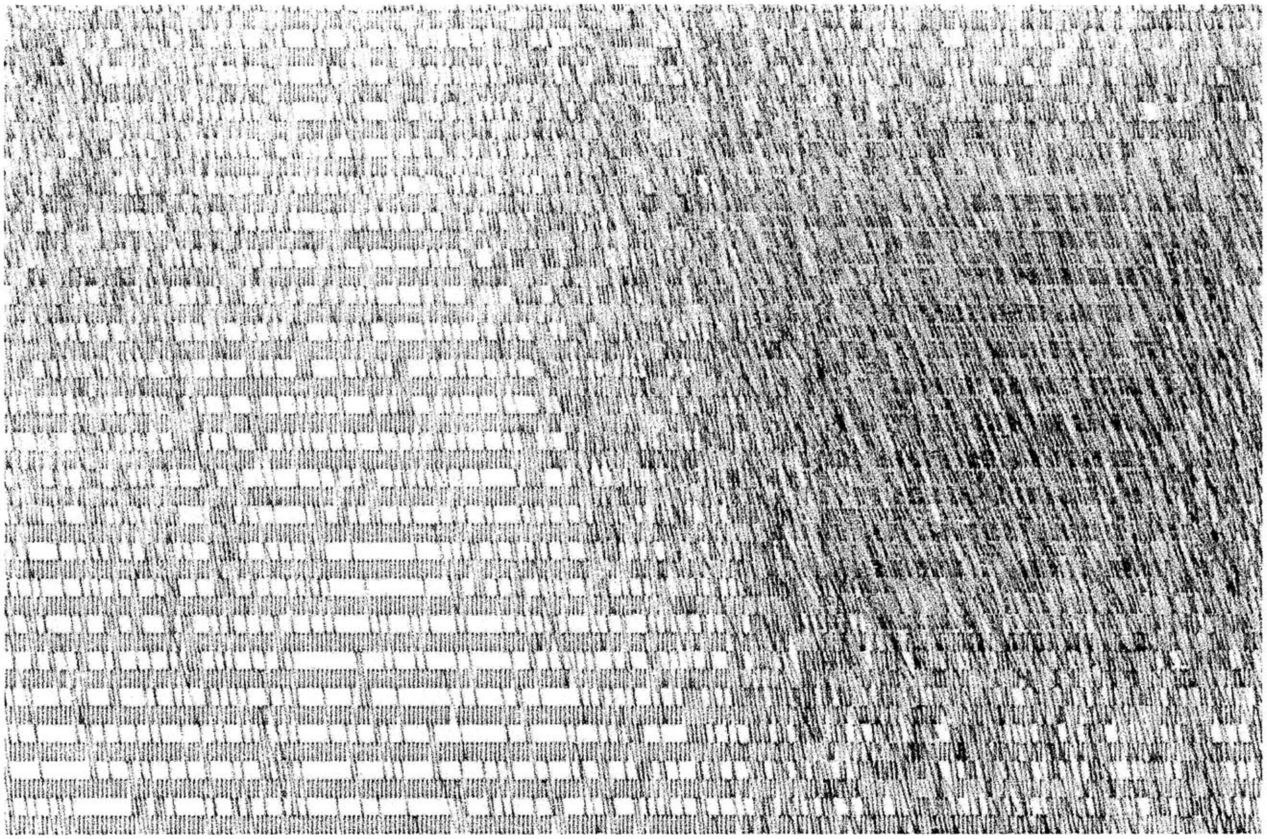
[2] Ibid



G-002-380: Once a Rectangle; Drawn With a Slightly Worn Pen

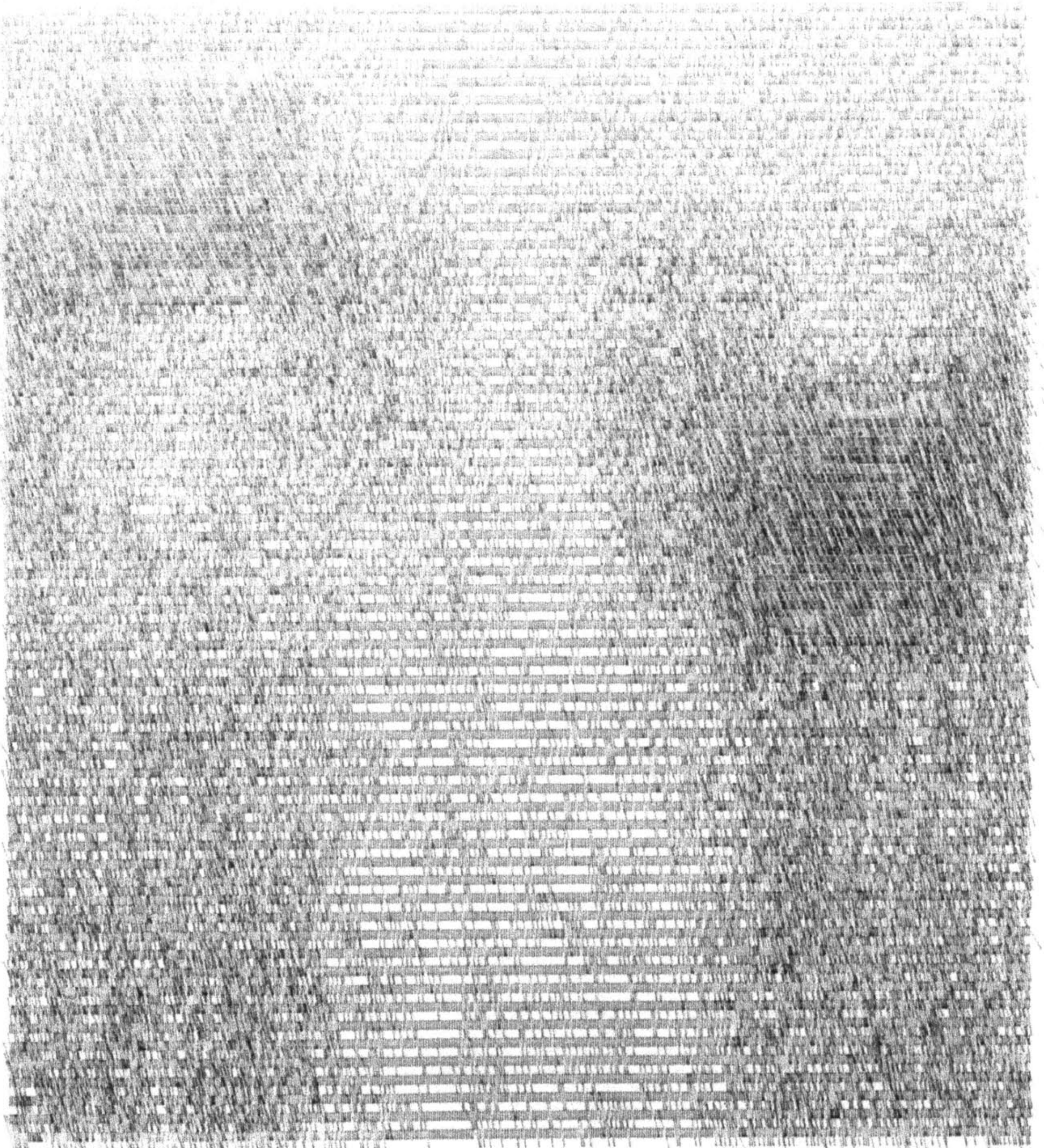
Method 'G' uses a lattice to represent the drawing. At each lattice point, six bits are stored. These bits are translated into parameters that determine pen strokes in the drawing. On the algorithmic temporal axis, the initial state is composed. In discrete steps forward in time, the bits shift in each of six directions based on simple rules designed to simulate fluid diffusion. Series 002 captures a moment of a changing 450x200 unit lattice. The lattice is mapped onto paper by translating the 6-bit value at each intersection into length and angle information of each stroke. With 379 time steps occurring before this run, the rectangle is long into the process of dissolution. Because each drawing and each corresponding lattice are closed systems, (the far left is linked to the far right; no state can ever move "off" the drawing) the size and proportion affects rate of rhythmic echoes of the initial state. The "shadow" of the rectangle appears at the lower left, a result of the proliferation of common bit positions that, as of time 380, have flowed without many collusions other states. Eventually, even the most stubborn shapes will completely disperse into a state of pure noise.





G-002-682: Long Ago a Rectangle; Drawn With a Failing Pen

Method 'G' uses a lattice to represent the drawing. At each lattice point, six bits are stored. These bits are translated into parameters that determine pen strokes in the drawing. On the algorithmic temporal axis, the initial state is composed. In discrete steps forward in time, the bits shift in each of six directions based on simple rules designed to simulate fluid diffusion. Series 002 captures a moment of a changing 450x200 unit lattice. The lattice is mapped onto paper by translating the 6-bit value at each intersection into length and angle information of each stroke. With 681 time steps occurring before this drawing, the "rectangle" (now only a memory of a seemingly arbitrary state) has dissolved. This step was chosen for drawing because of its in-between nature: in a few more steps, the figural areas will appear completely coincidental–barely legible areas of similar stroke types in loose clusters amongst noise.

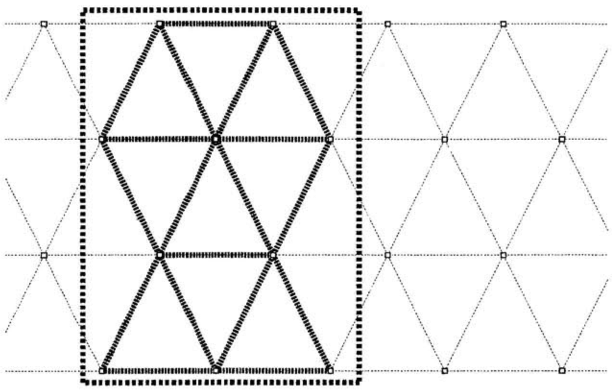


Series G-003

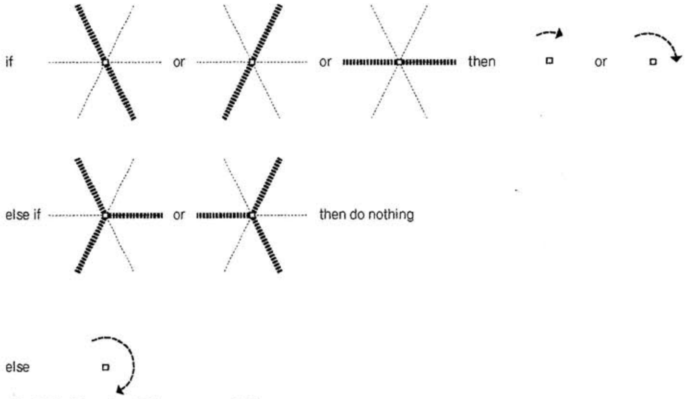
The lattice structure and behavior are implemented based on the algorithm published by Neil Gershenfeld.[3] Each intersection on the lattice is represented by six bits, which correspond to on or off values for each link to a neighbor point. The values are asserted based on a simple geometry. All bits within a closed rectangle are set to "on," while all others are set to "off." (G-003-VectorDiagram-01)

At each algorithmic step, a new 6-bit configuration is calculated based on the previous state. Two special cases represent collisions, in which case the bit values "bounce" off one another. In all other cases, the values "flow" through the intersection point (G-003-VectorDiagram-02)

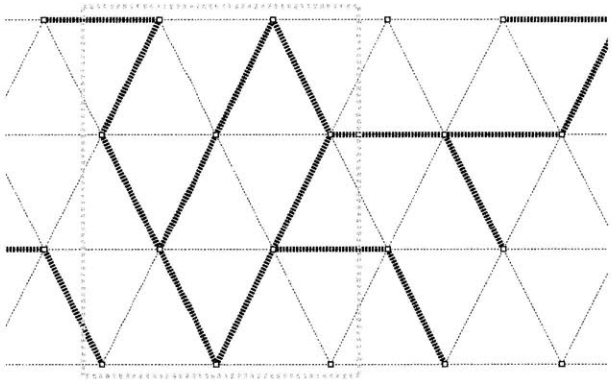
Each run is a paused moment, which references the state of the entire lattice. (G-003-VectorDiagram-03) The lattice is skewed to a regular Cartesian grid. The 6-bit value at each intersection grid corresponds not to an set of marked lines on the paper. These lines are drawn within a length equivalent to the horizontal spacing of the grid on paper. The number of on bits at each intersection influences the angle of each set. The lines have variable length because they always reach the same height. (G-003-VectorDiagram-04)



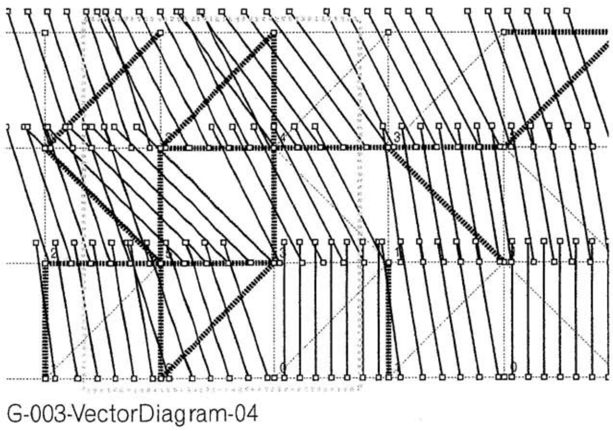
G-003-VectorDiagram-01



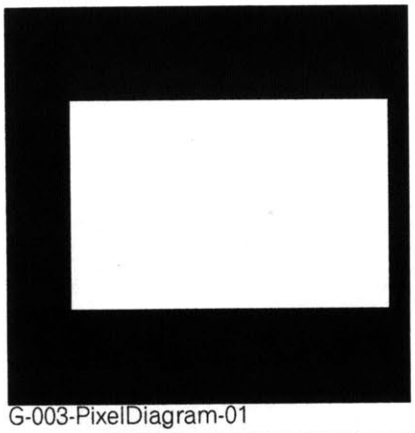
G-003-VectorDiagram-02



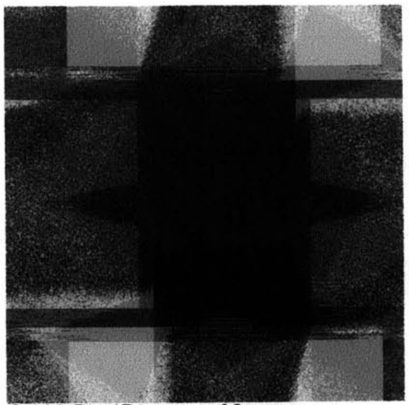
G-003-VectorDiagram-03



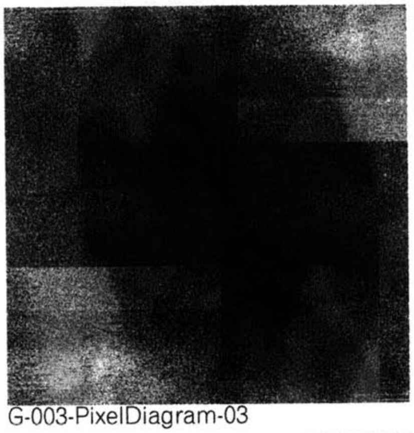
G-003-VectorDiagram-04



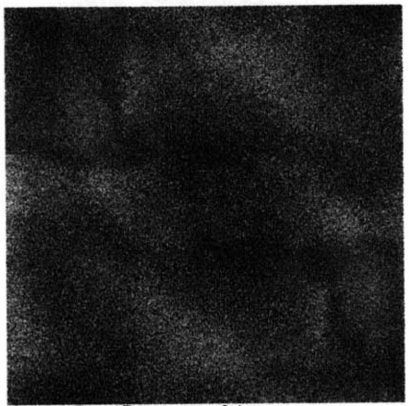
G-003-PixelDiagram-01



G-003-PixelDiagram-02

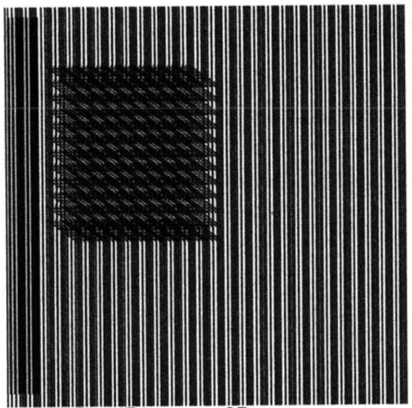


G-003-PixelDiagram-03

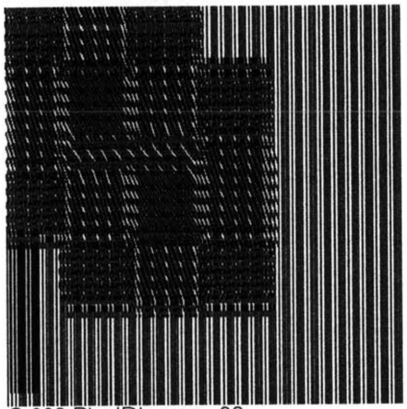


G-003-PixelDiagram-04

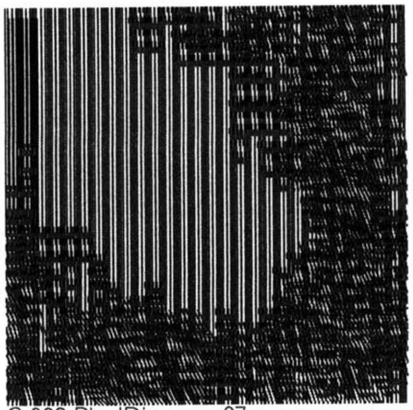
Two forms of pixel representations document the computational process. The first (G-003-PixelDiagram-01, G-003-PixelDiagram-02, G-003-PixelDiagram-03, and G-003-PixelDiagram-04) associates each lattice point with one pixel. The grayscale value of each pixel is associated with the number of on bits at that point. If all six bits are on the pixel is rendered white. If all are off the pixel is rendered black. Tallies between 1 and 5 are intermediate grays.



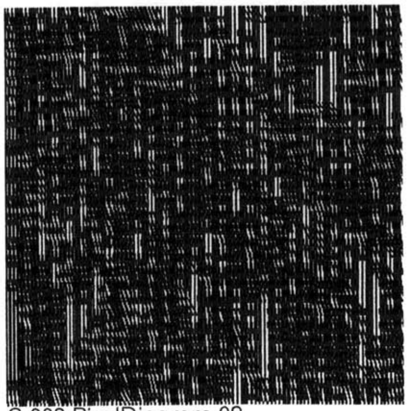
G-003-PixelDiagram-05



G-003-PixelDiagram-06



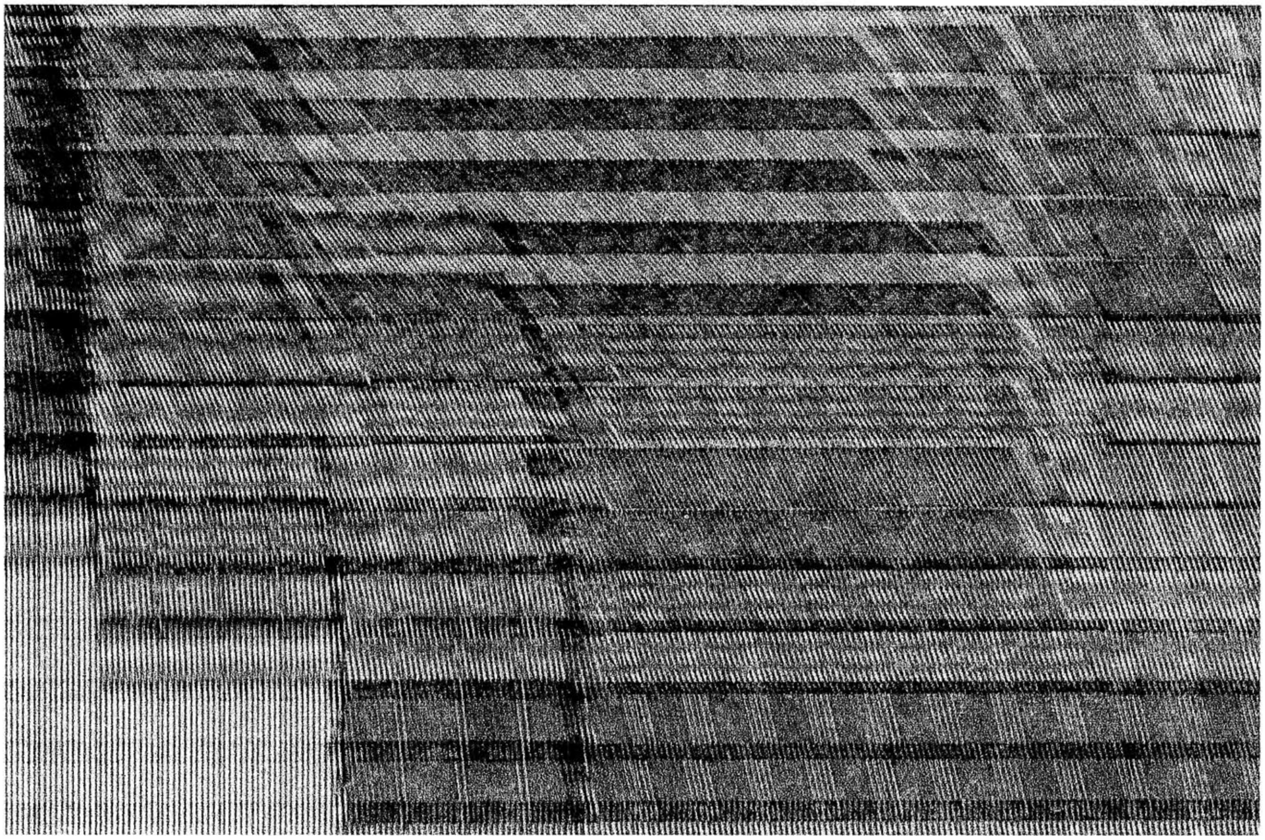
G-003-PixelDiagram-07



G-003-PixelDiagram-08

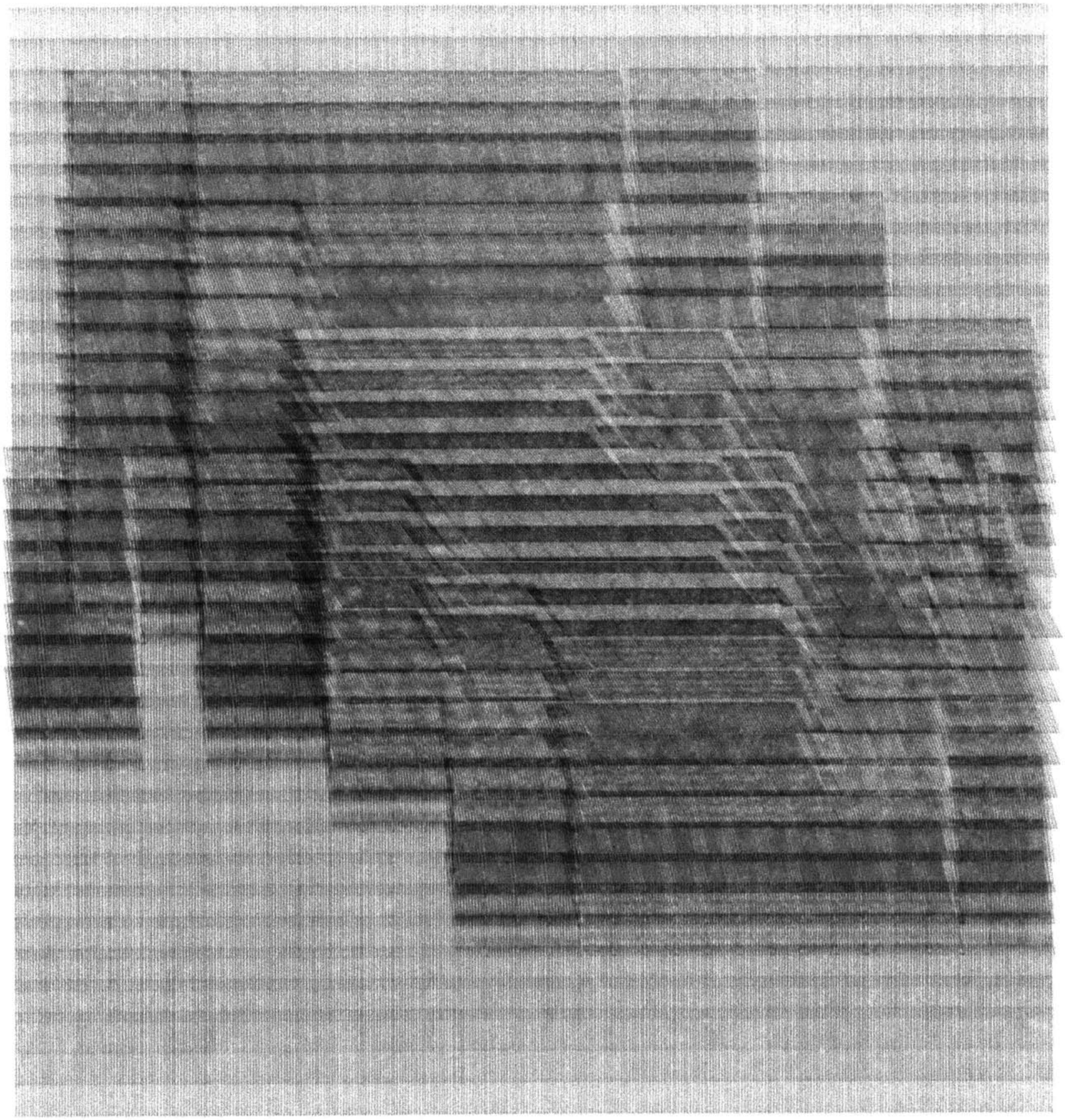
The second form of pixel representation (G-003-PixelDiagram-05, G-003-PixelDiagram-06, G-003-PixelDiagram-07, and G-003-PixelDiagram-08) associates the image with the paper. A lattice with resolution lower than that of the image is used and lines are rendered in manner that crudely approximates the lines on paper.

3 Ibid



G-003-019: The Moment of Deviation; Drawn With a Worn Pen

Method 'G' uses a lattice to represent the drawing. At each lattice point, six bits are stored. These bits are translated into parameters that determine pen strokes in the drawing. On the algorithmic temporal axis, the initial state is composed. In discrete steps forward in time, the bits shift in each of six directions based on simple rules designed to simulate fluid diffusion. In Series 003, a moment of a changing 450x200 unit lattice is mapped onto paper by translating the 6-bit value at each intersection into length and angle information of sets of strokes. This introduces an intermediate scale into the drawing. As patches of lines collectively change angle and length, their overlap with other patches produces an area of consistent texture. Even when only "slightly subject to time," the near-depth effect is dominant, relying on the ambiguity of conflicting meanings of the line: as particle in a field and as mark at the edge of a figure.



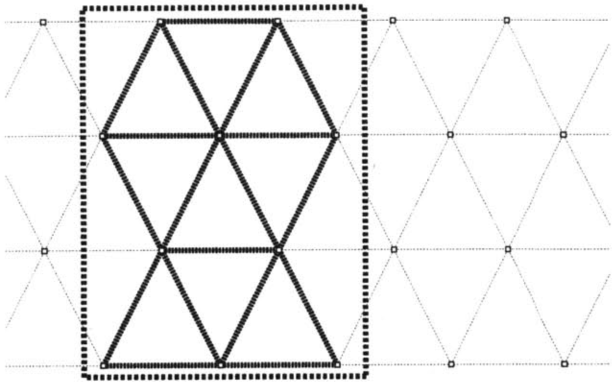
G-003-019

Series G-004

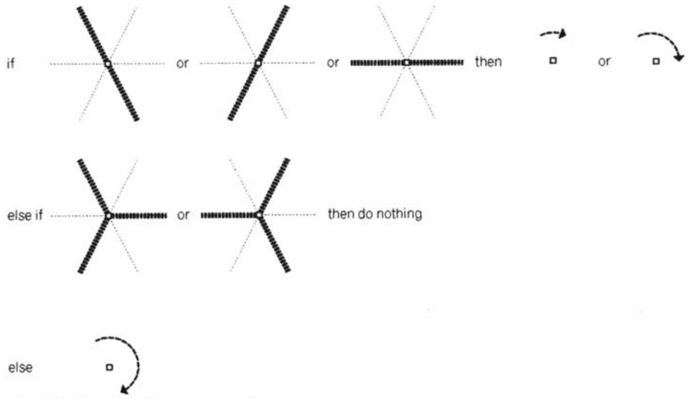
The lattice structure and behavior are implemented based on the algorithm published by Neil Gershenfeld.[4] Each intersection on the lattice is represented by six bits, which correspond to on or off values for each link to a neighbor point. The values are asserted based on a simple geometry. All bits within a closed rectangle are set to "on," while all others are set to "off." (G-004-VectorDiagram-01)

At each algorithmic step, a new 6-bit configuration is calculated based on the previous state. Two special cases represent collisions, in which case the bit values "bounce" off one another. In all other cases, the values "flow" through the intersection point (G-004-VectorDiagram-02)

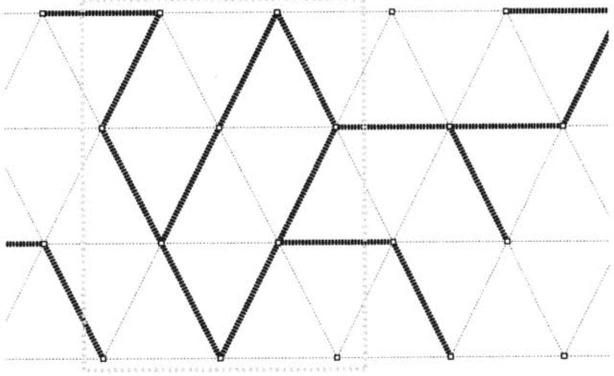
Each run is representation of multiple paused moments, each of which references the state of the entire lattice. (G-004-VectorDiagram-03) These moments are drawn on the same sheet of paper, in sequence. Before drawing, the lattice is skewed to a regular Cartesian grid. The 6-bit value at each intersection grid corresponds to four marked lines on the paper, which form a rectangle. The width and height of this rectangle is governed by number of on bits at each intersection. (G-004-VectorDiagram-04)



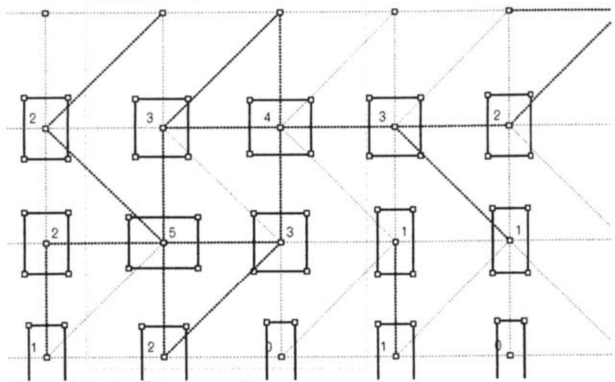
G-004-VectorDiagram-01



G-004-VectorDiagram-02

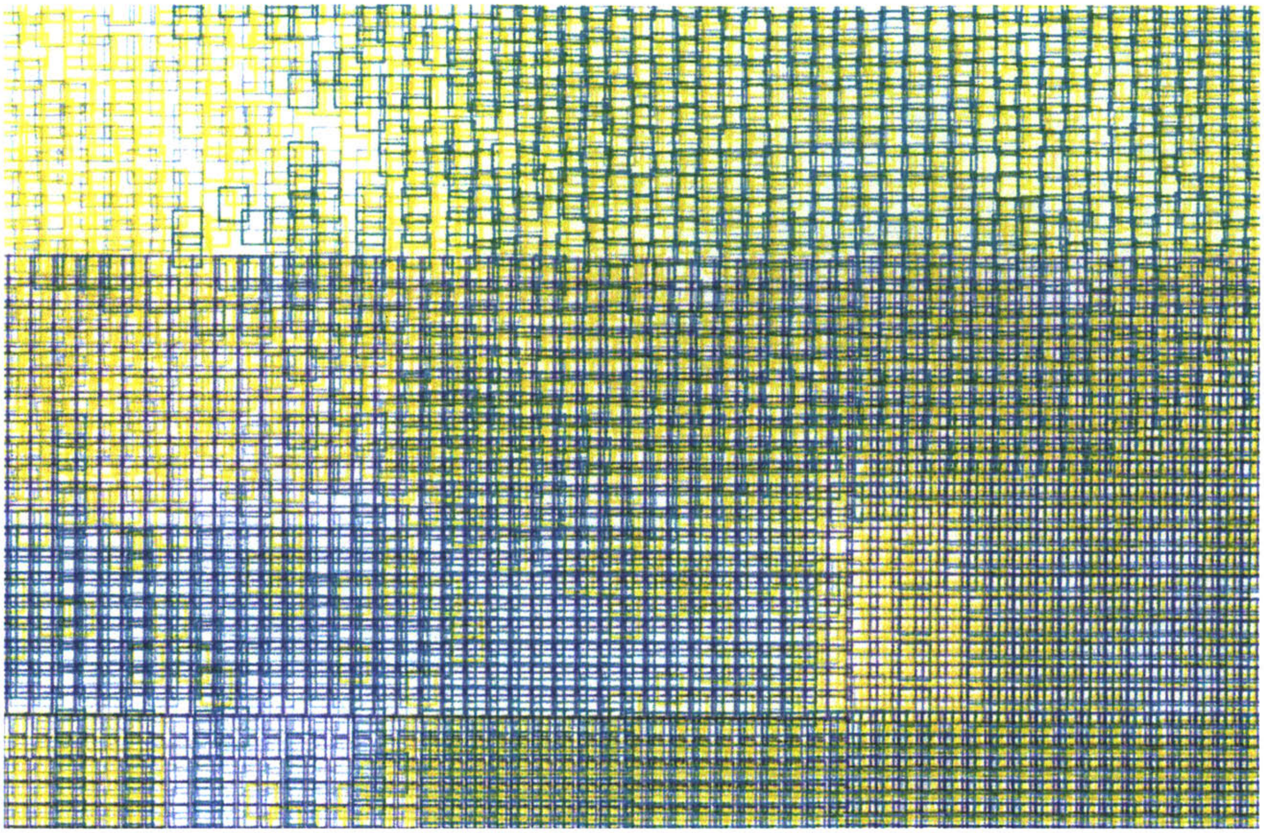


G-004-VectorDiagram-03



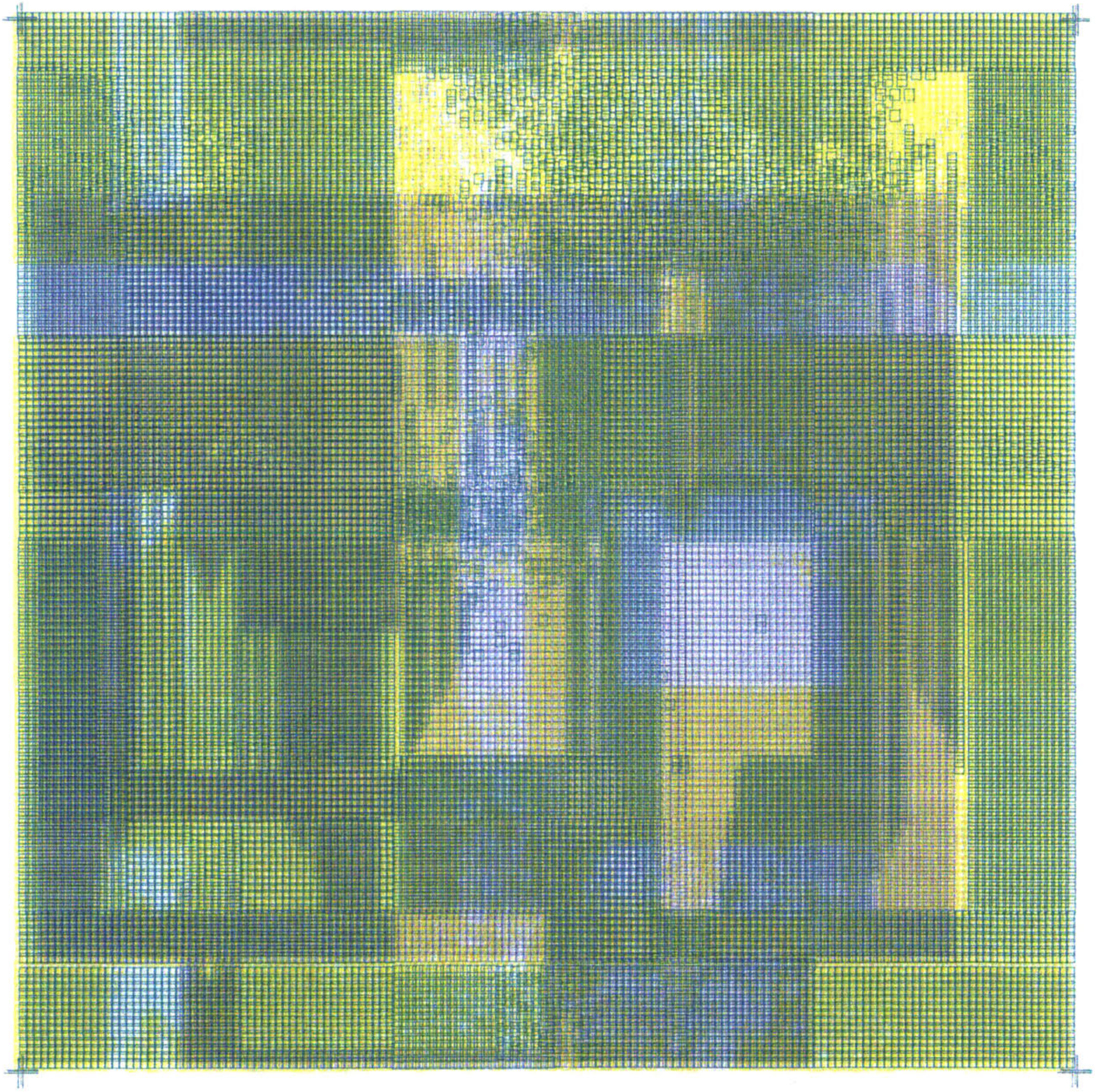
G-001-VectorDiagram-04

[4] Ibid



G-004-205: Figure and Ground Rectangle Diffusion

Method 'G' uses a lattice to represent the drawing. At each lattice point, six bits are stored. These bits are translated into parameters that determine pen strokes in the drawing. On the algorithmic temporal axis, the initial state is composed. In discrete steps forward in time, the bits shift in each of six directions based on simple rules designed to simulate fluid diffusion. In Series 004, multiple moments of a changing 100x100 unit lattice are simultaneously drawn. The bits at each intersection determine dimensions and offset of a rectangle. Run 205 draws times 50, 100, and 205 on the sheet using blue, green and yellow pens respectively.



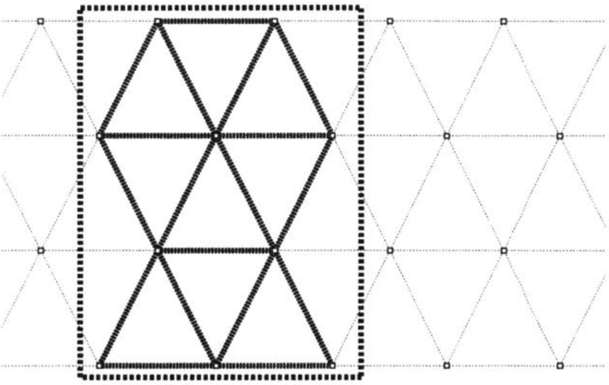
G-004-205

Series G-005

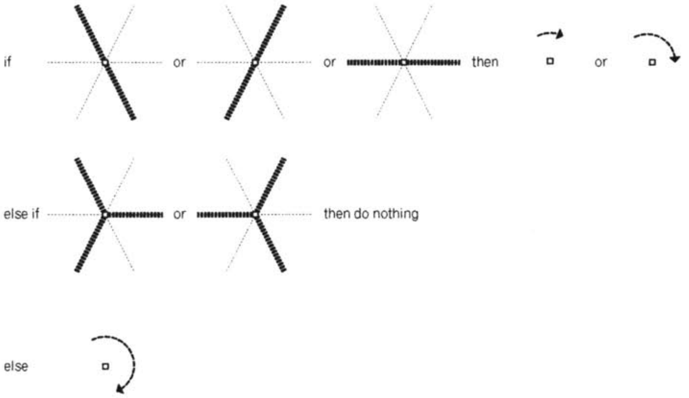
The lattice structure and behavior are implemented based on the algorithm published by Neil Gershenfeld.[5] Each intersection on the lattice is represented by six bits, which correspond to on or off values for each link to a neighbor point. The values are asserted based on a simple geometry. All bits within a closed rectangle are set to "on," while all others are set to "off." (G-005-VectorDiagram-01)

At each algorithmic step, a new 6-bit configuration is calculated based on the previous state. Two special cases represent collisions, in which case the bit values "bounce" off one another. In all other cases, the values "flow" through the intersection point (G-005-VectorDiagram-02)

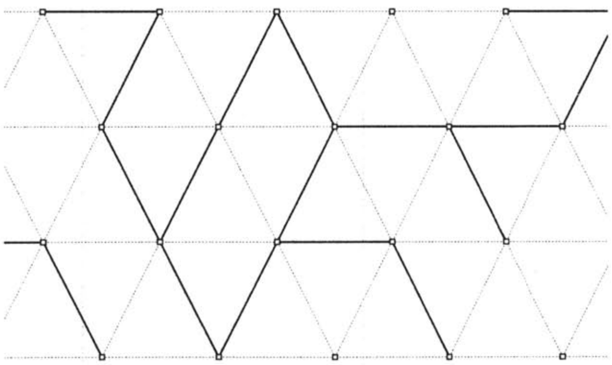
Each run is representation of multiple paused moments, each of which references the state of the entire lattice. (G-005-VectorDiagram-03) These moments are drawn on the same sheet of paper, in sequence. The lattice is mapped directly onto the paper. No translation takes place other than scaling to move from the unitless space of the lattice to the physical dimensions of paper. Any on bit is drawn as a line connecting corresponding lattice points. Circles mark any point in which special "collision" rules were applied. (G-005-VectorDiagram-04)



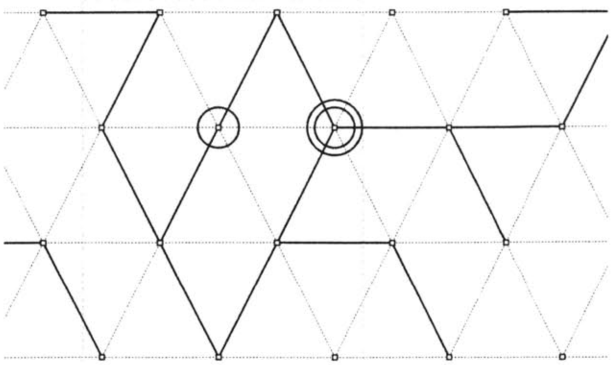
G-005-VectorDiagram-01



G-005-VectorDiagram-02

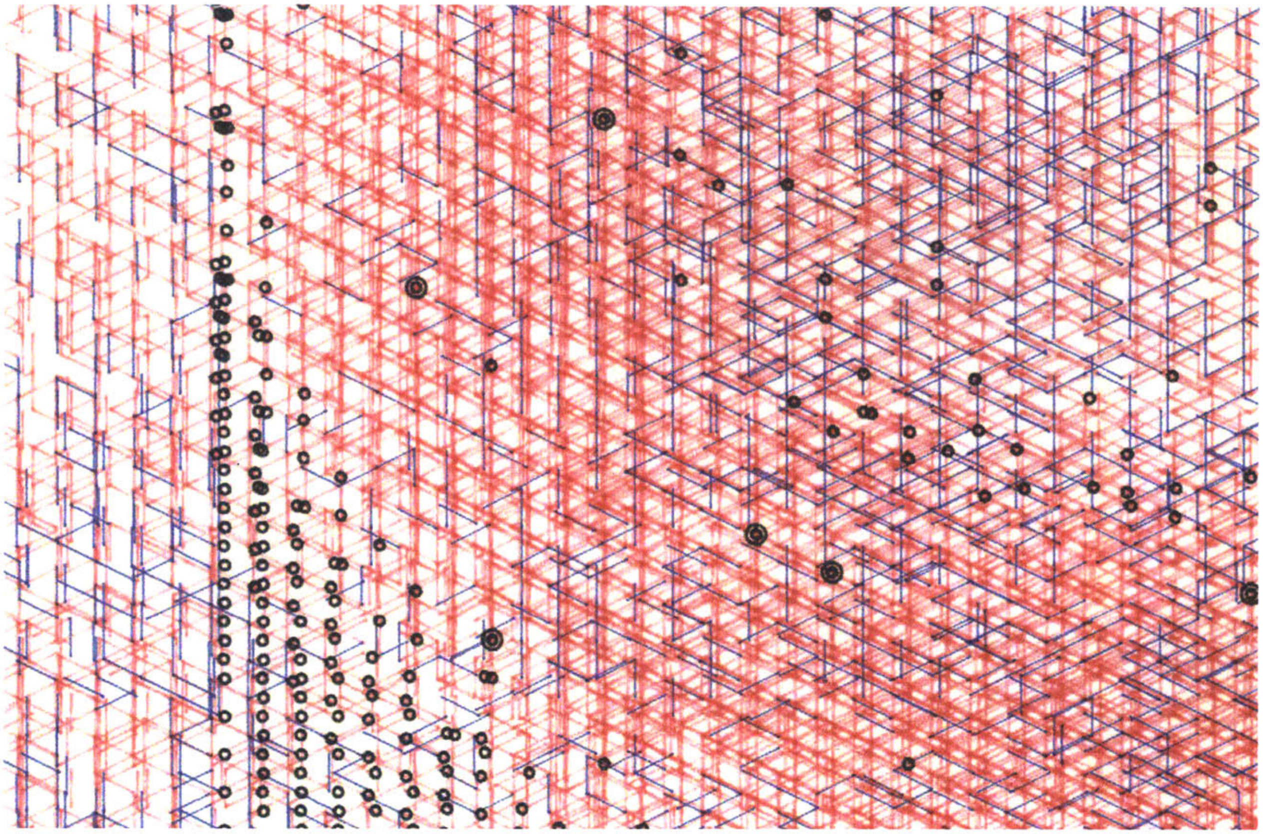


G-005-VectorDiagram-03



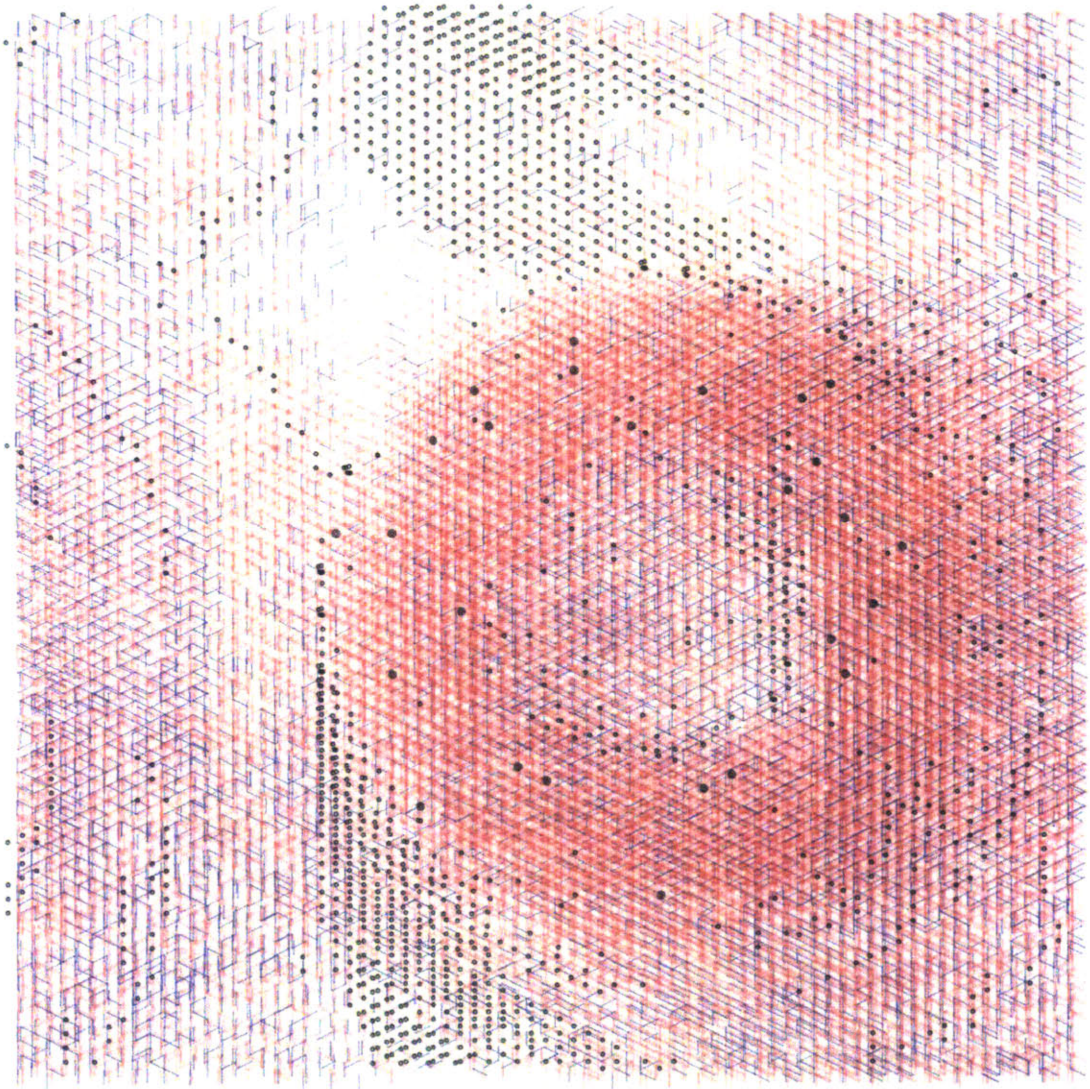
G-005-VectorDiagram-04

[5] Ibid



G-005-190: Circle Diffusing with Marked Anomalies

Method 'G' uses a lattice to represent the drawing. At each lattice point, six bits are stored. These bits are translated into parameters that determine pen strokes in the drawing. On the algorithmic temporal axis, the initial state is composed. In discrete steps forward in time, the bits shift in each of six directions based on simple rules designed to simulate fluid diffusion. In Series 005, multiple moments of a changing 100x200 unit lattice are simultaneously drawn. Small circles mark points where one of three rules, which govern the proliferation of "on" states at each vertex, were applied. Each on bit corresponds to a single drawn line. In this run, a thickened circular zone has been diffusing. Three cored pens mark the states at times 5, 100, and 190.



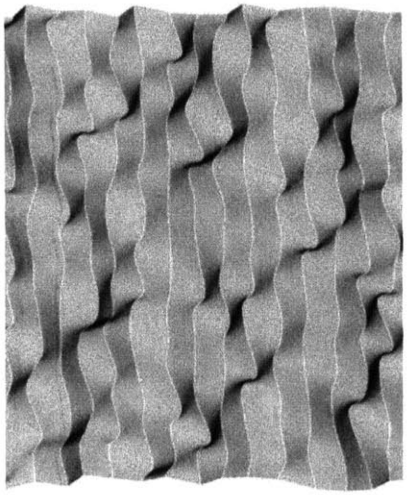
G-005-190

Method K

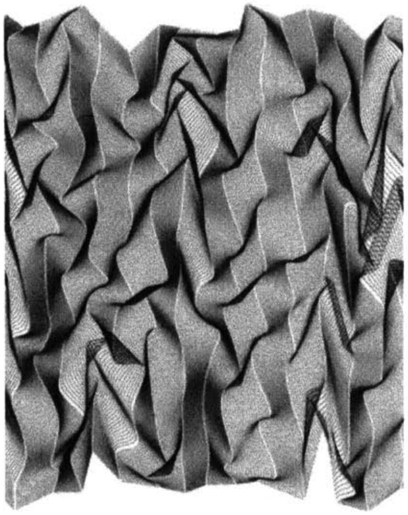
Method 'K' uses an underlying set of points as the structure of each drawing. These points are assigned behavior, which causes them to move. Each drawing represents a moment of the changing state of points, which are initially arranged in a regular grid. At each moment, the points are organized into sets, which generate a series of lines.

Series K-002

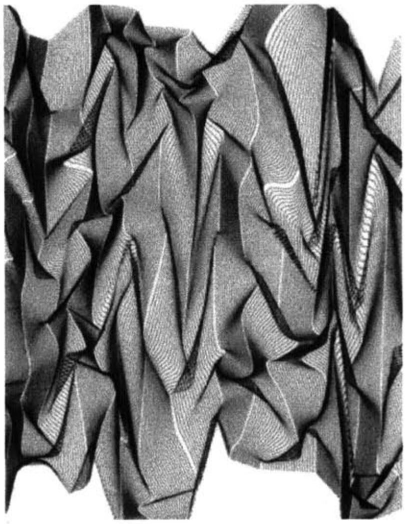
At each moment, 400 points are sorted and grouped based on their position on the x-axis. 20 splines are interpolated between each group.



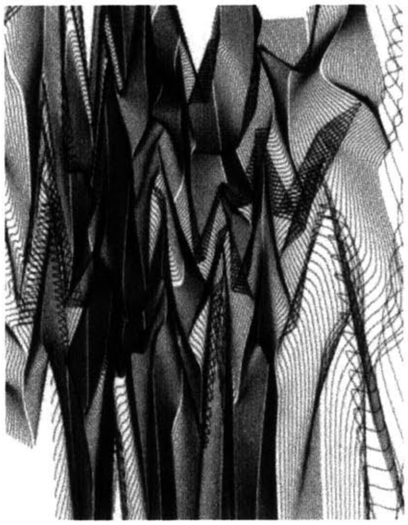
K-002-010



K-002-050



K-002-090



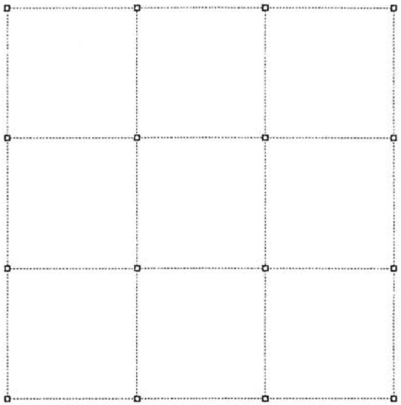
K-002-130

Series K-002

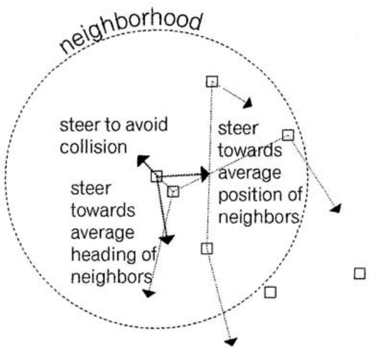
Initially, points are organized on a regular grid. (K-002-VectorDiagram-01) These points are associated with initially random velocity and acceleration vectors. Their acceleration is influenced by nearby points according to swarming "boid" rules published by Craig Renoylds.[6] (K-002-VectorDiagram-02) Points tend to form and move in clusters. (K-002-VectorDiagram-03)

At a certain interval, the movement of points is paused and a drawing calculated. The points are first sorted into groups of equal number based on their x position. (K-002-VectorDiagram-04) Within these groups, the points are sorted by their y position, and treated as a polyline. (K-002-VectorDiagram-05)

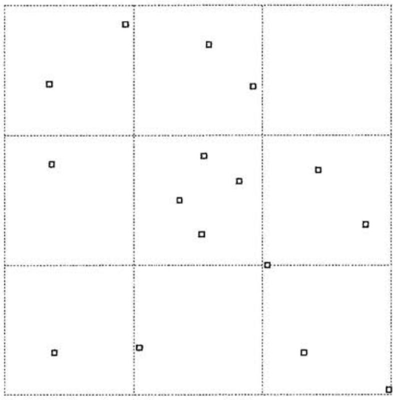
New polylines are interpolated between groups. (K-002-VectorDiagram-06) Each new polyline is subject to Catmull-Rom spline interpolation at a fine interval. Lines are drawn on paper between these points. (K-002-VectorDiagram-07)



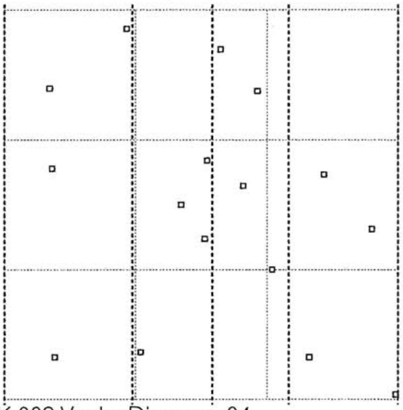
K-002-VectorDiagram-01


neighborhood
steer to avoid collision
steer towards average position of neighbors
steer towards average heading of neighbors

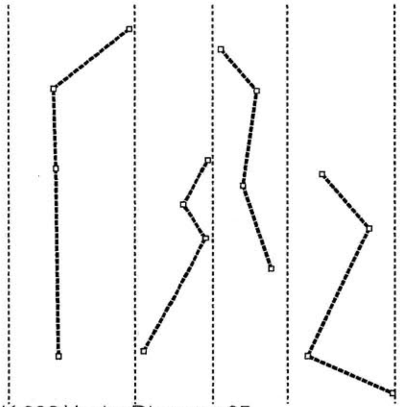
K-002-VectorDiagram-02



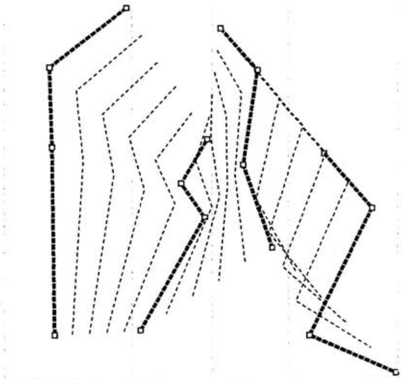
K-002-VectorDiagram-03



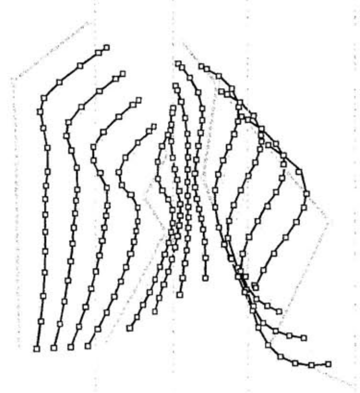
K-002-VectorDiagram-04



K-002-VectorDiagram-05

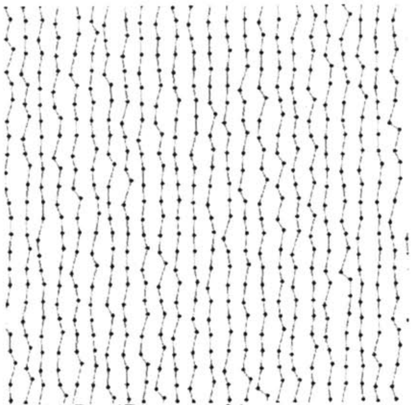


K-002-VectorDiagram-06

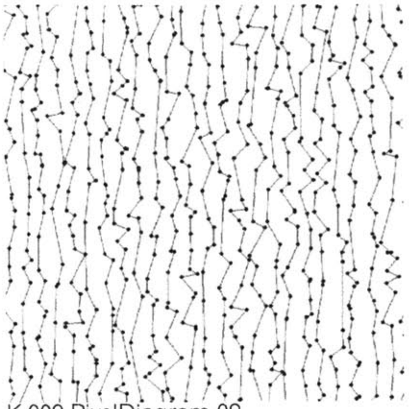


K-002-VectorDiagram-07

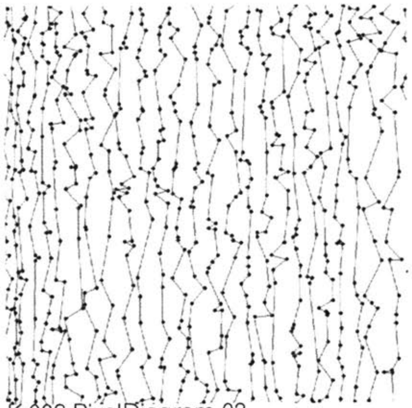
[6] Reynolds, "Flocks, Herds and Schools."



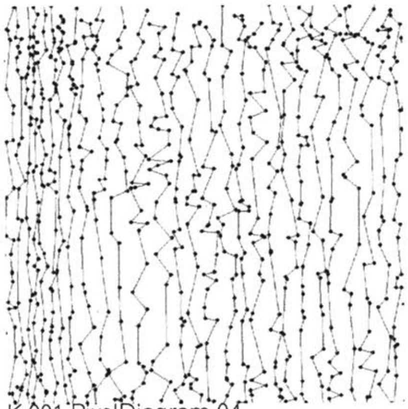
K-002-PixelDiagram-01



K-002-PixelDiagram-02

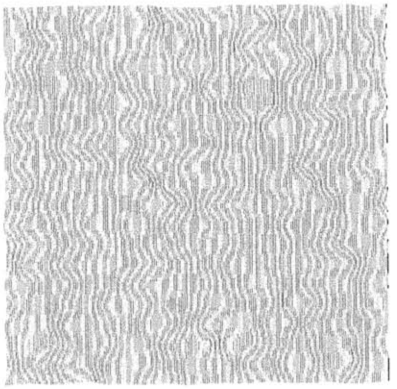


K-002-PixelDiagram-03

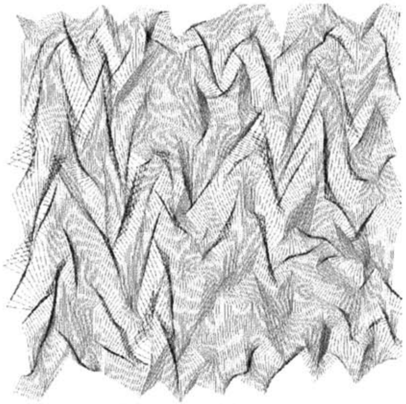


K-001-PixelDiagram-04

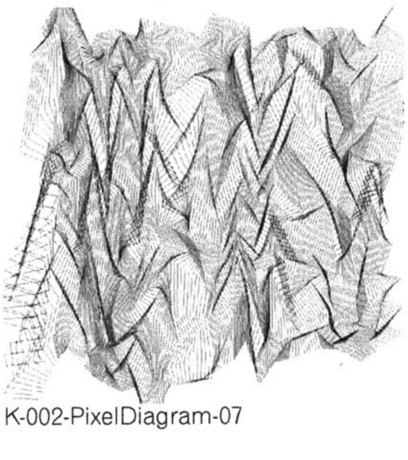
This series of representations shows the position of points and polylines, which are the result of horizontal sorting of points, at times 5, 30, 55, and 80.



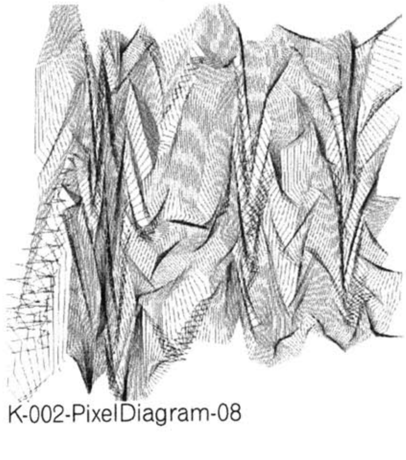
K-002-PixelDiagram-05



K-002-PixelDiagram-06

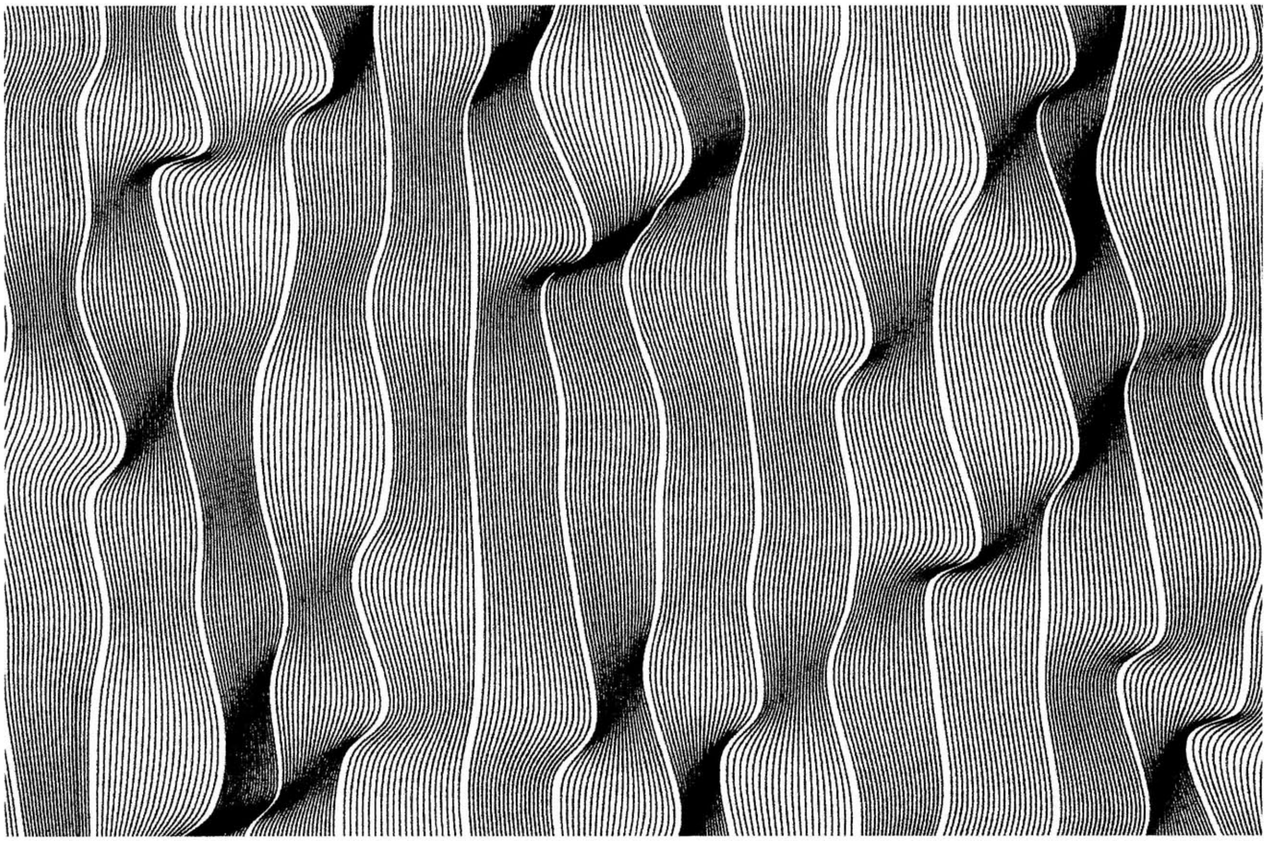


K-002-PixelDiagram-07



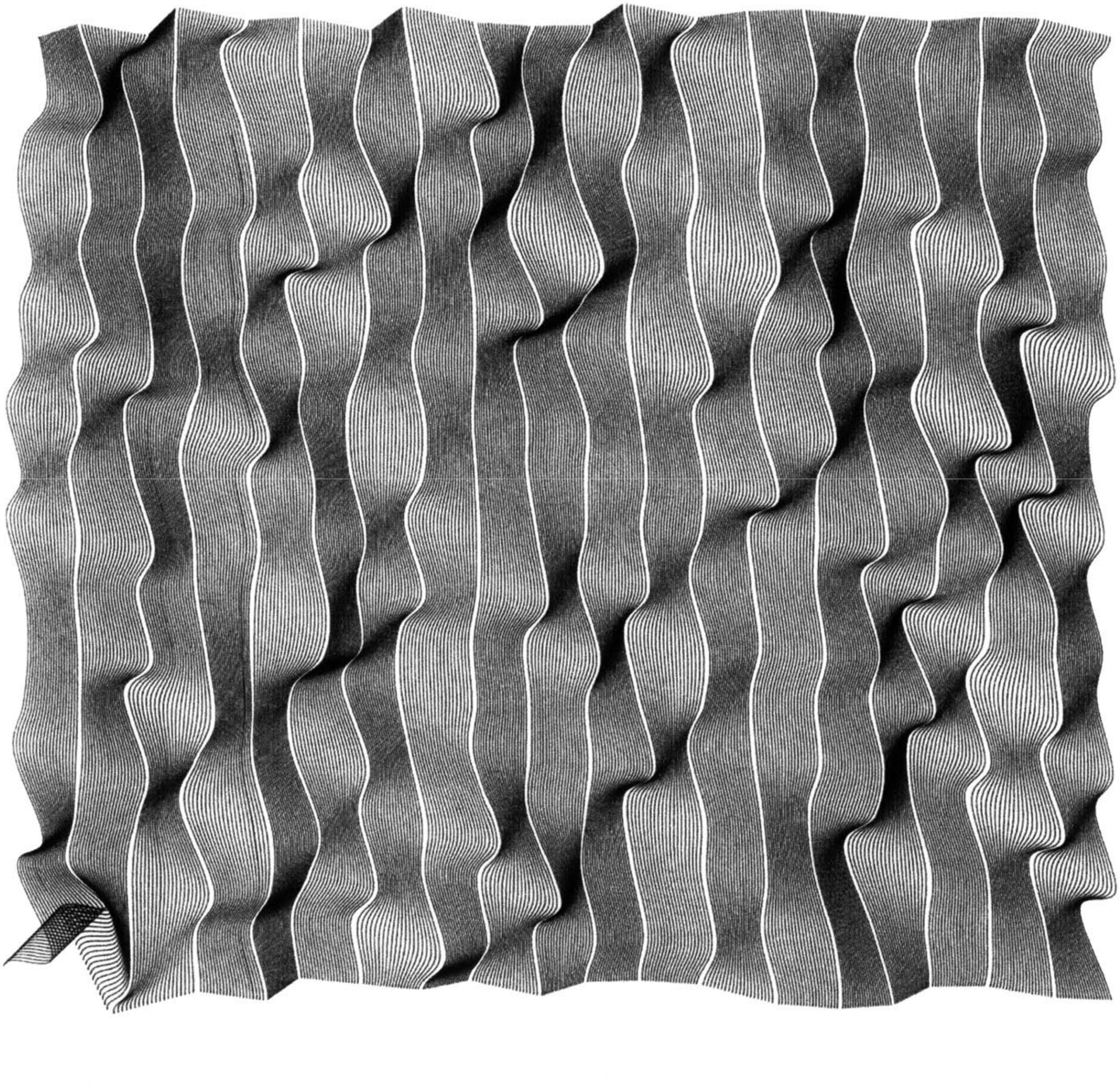
K-002-PixelDiagram-08

This series of representations shows the interpolated polylines between the primary polylines. These new lines with serve as the input for spline curves. Times 5, 30, 55, and 80 are shown.

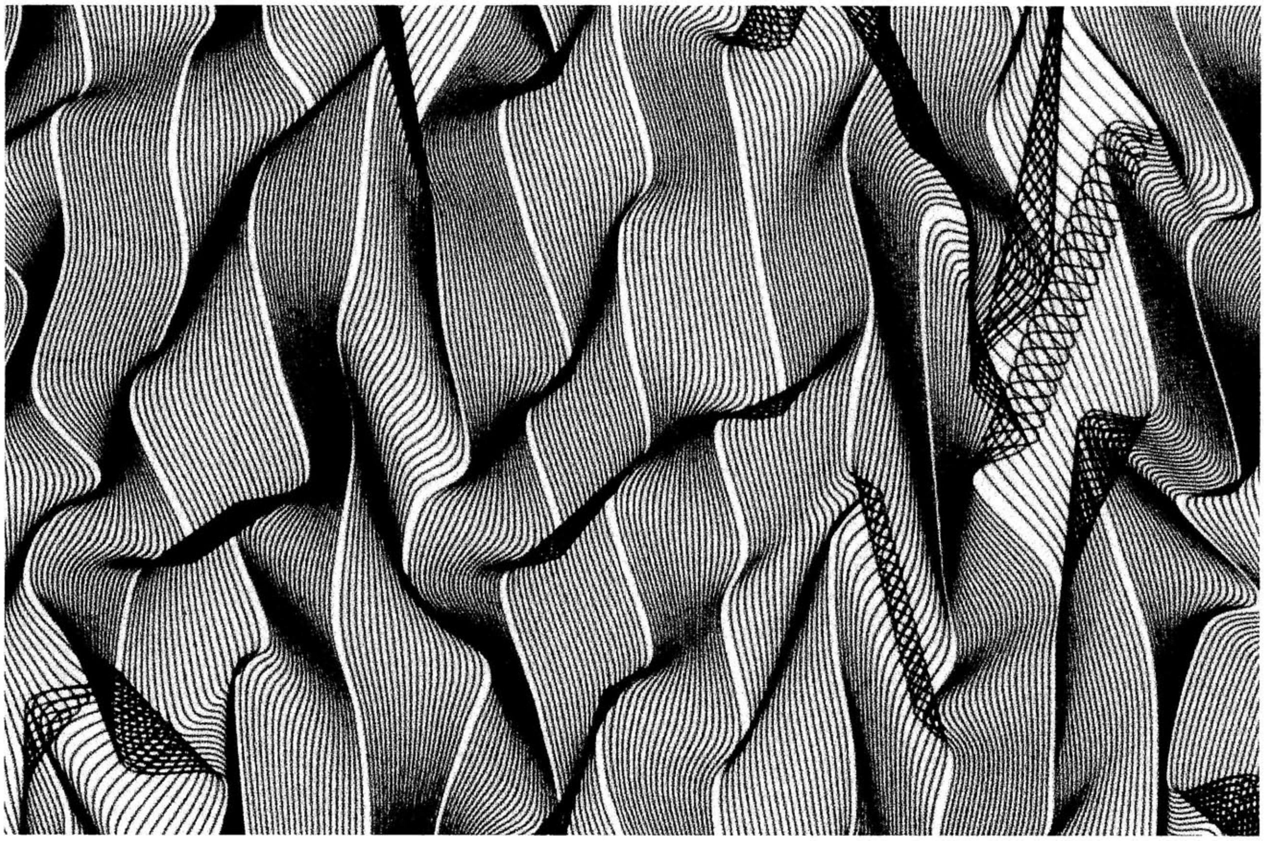


K-002-010: Swarming Striations at Time 10

Method 'K' uses an underlying set of points as the structure of each drawing. In series 002, the point positions are subject to swarming behavior. Sweeping laterally across the field groups the points according to their x position. Each group is then sorted according to their y position and a single line is drawn through the sorted points in each group. Every 1 in 50 lines is omitted. The points are initially organized on a grid; by run 10, they have only slightly deviated from their initial state.

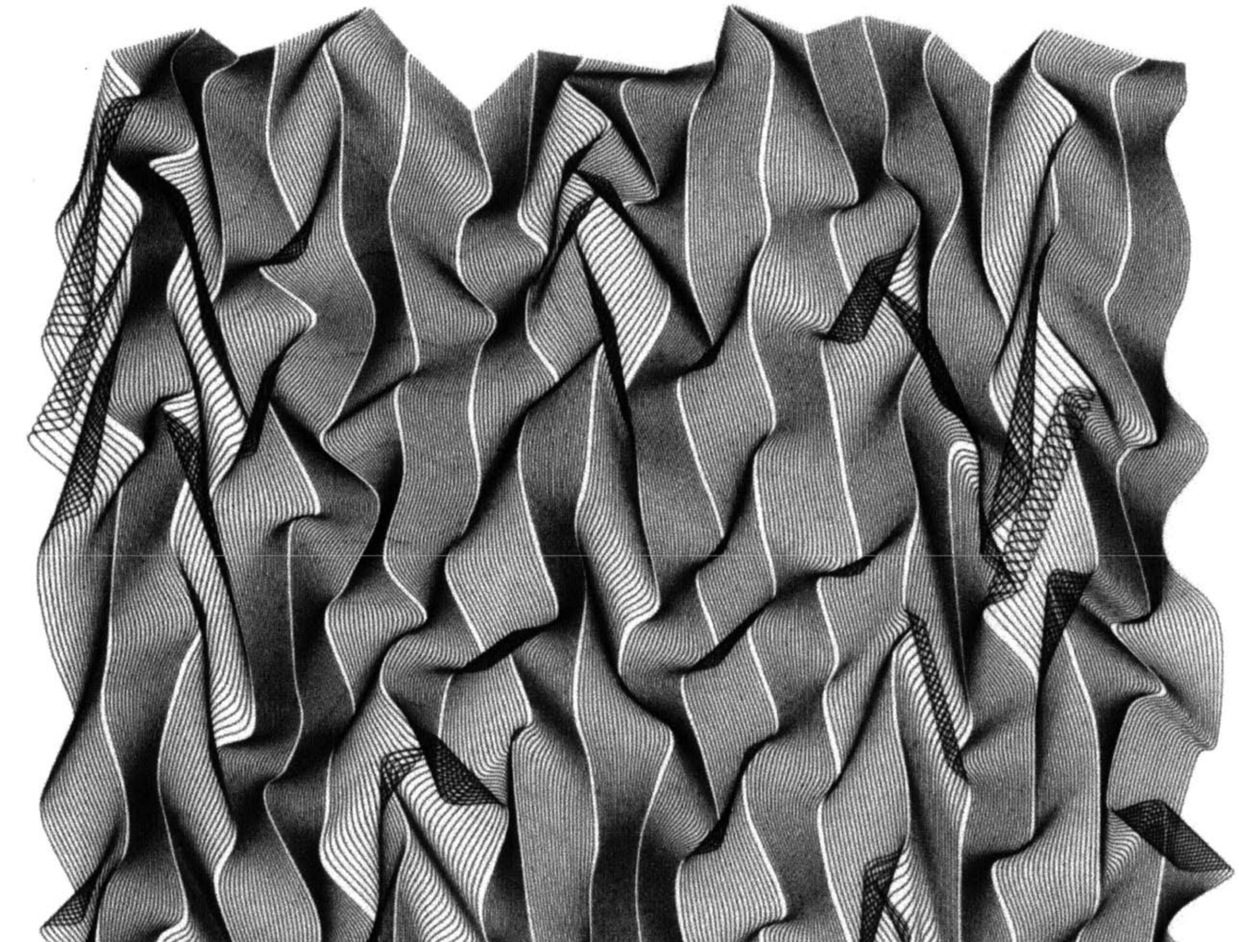


K-002-010

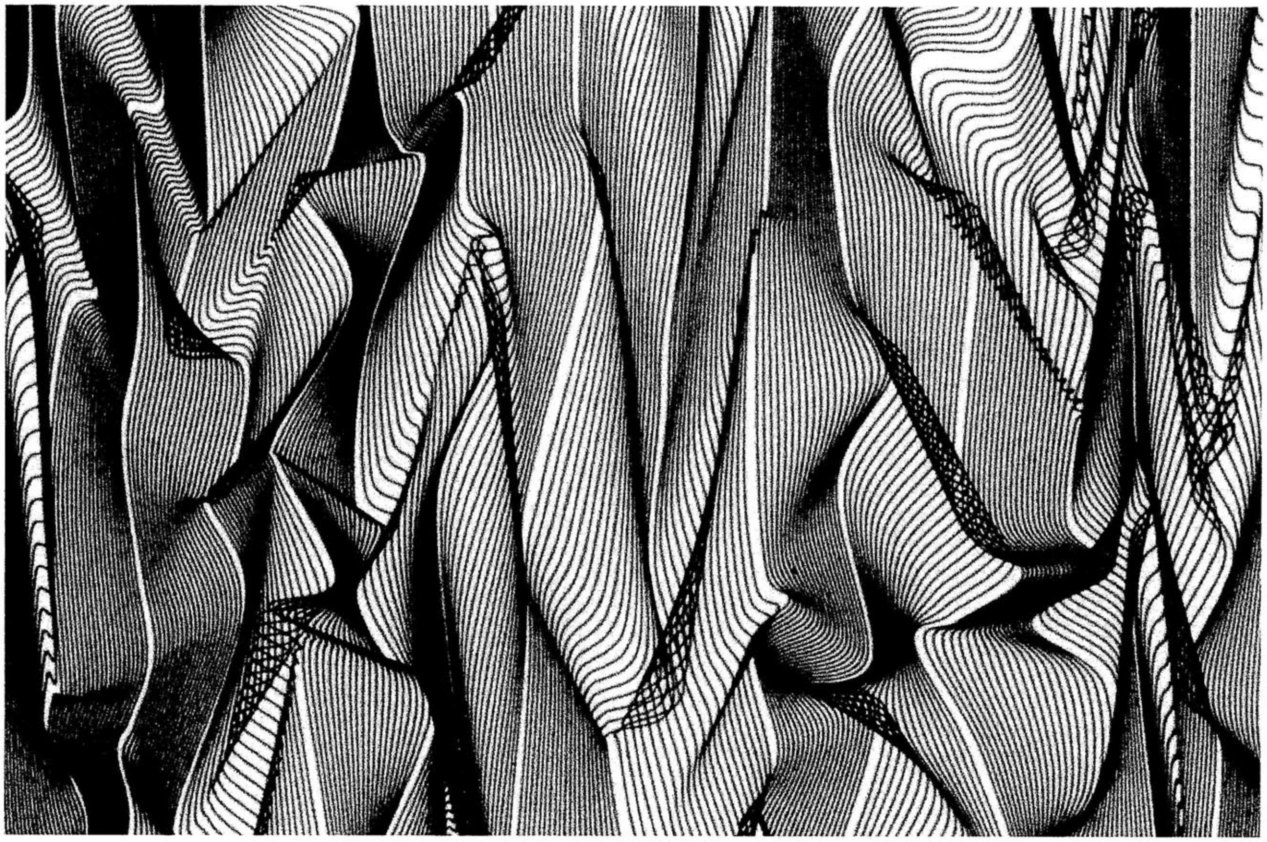


K-002-050: Swarming Striations, at Time 50

Method 'K' uses an underlying set of points as the structure of each drawing. In series 002, the point positions are subject to swarming behavior. Sweeping laterally across the field groups the points according to their x position. Each group is then sorted according to their y position and a single line is drawn through the sorted points in each group. Every 1 in 50 lines is omitted. The points are initially organized on a grid; by run 50, the grid barely registers and the points have begun to register more prominently as clusters within a field.

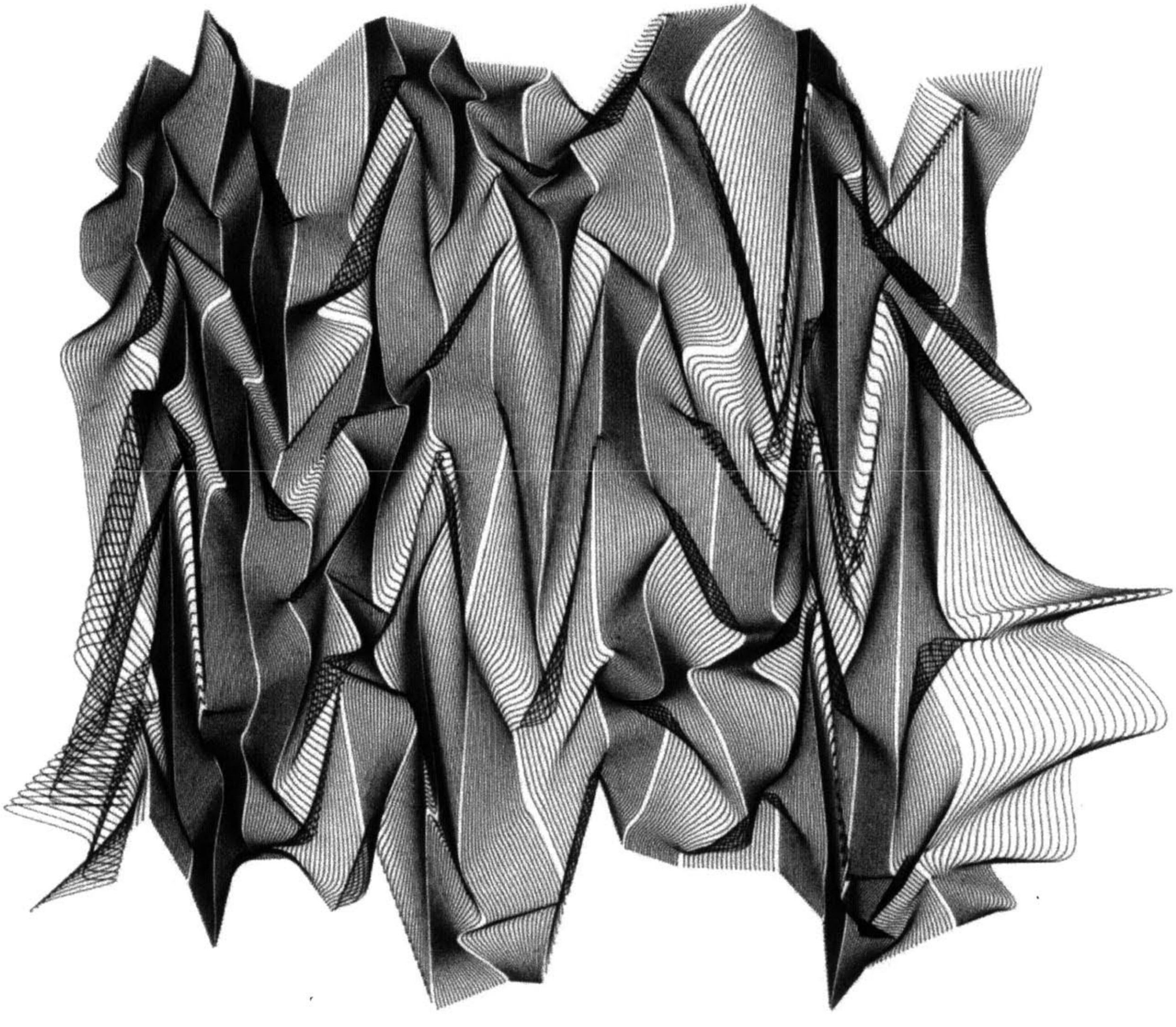


K-002-050

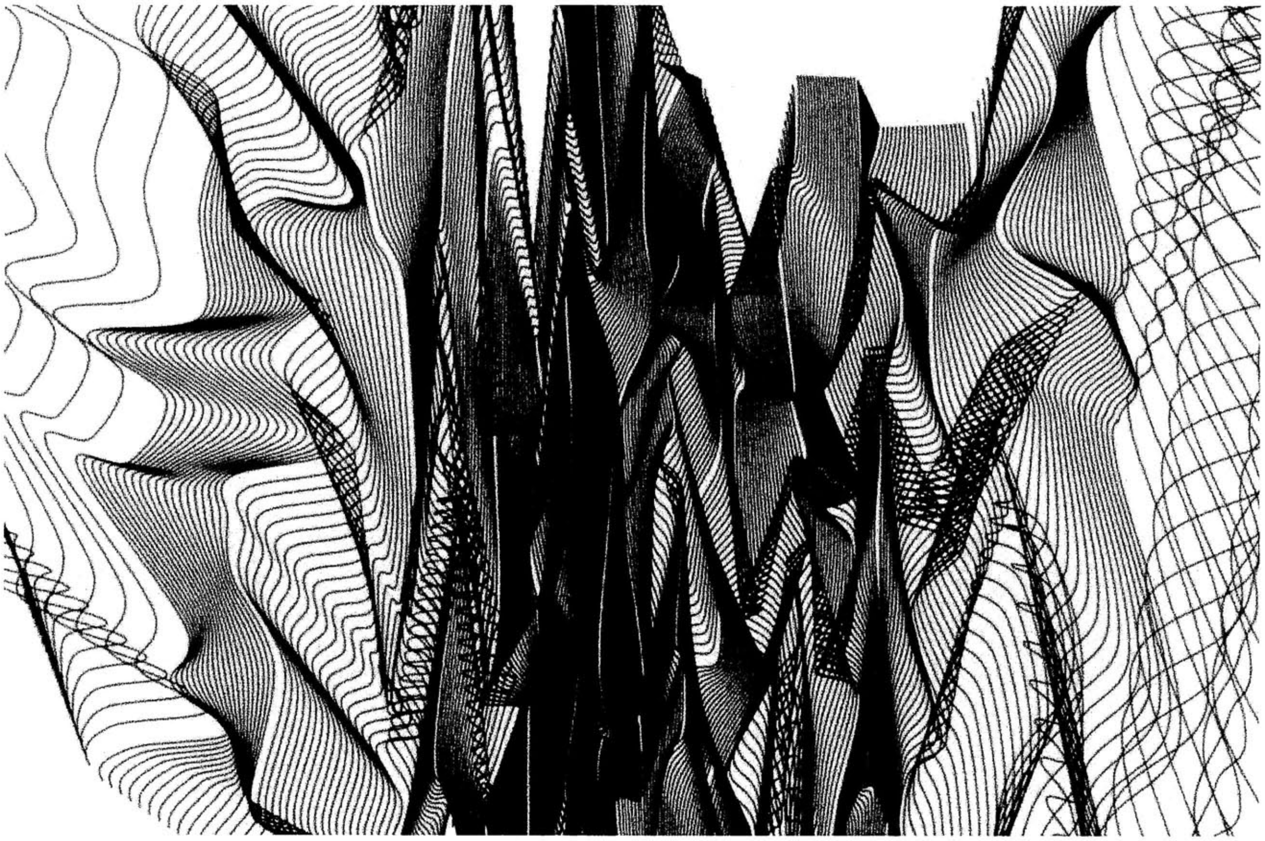


K-002-090: Swarming Striations at Time 90

Method 'K' uses an underlying set of points as the structure of each drawing. In series 002, the point positions are subject to swarming behavior. Sweeping laterally across the field groups the points according to their x position. Each group is then sorted according to their y position and a single line is drawn through the sorted points in each group. Every 1 in 50 lines is omitted. The points are initially organized on a grid; by run 90, the grid is a distant memory and points are so unevenly spaced that ridges and cliffs and other breaks in the apparent continuity of the striates surface register.

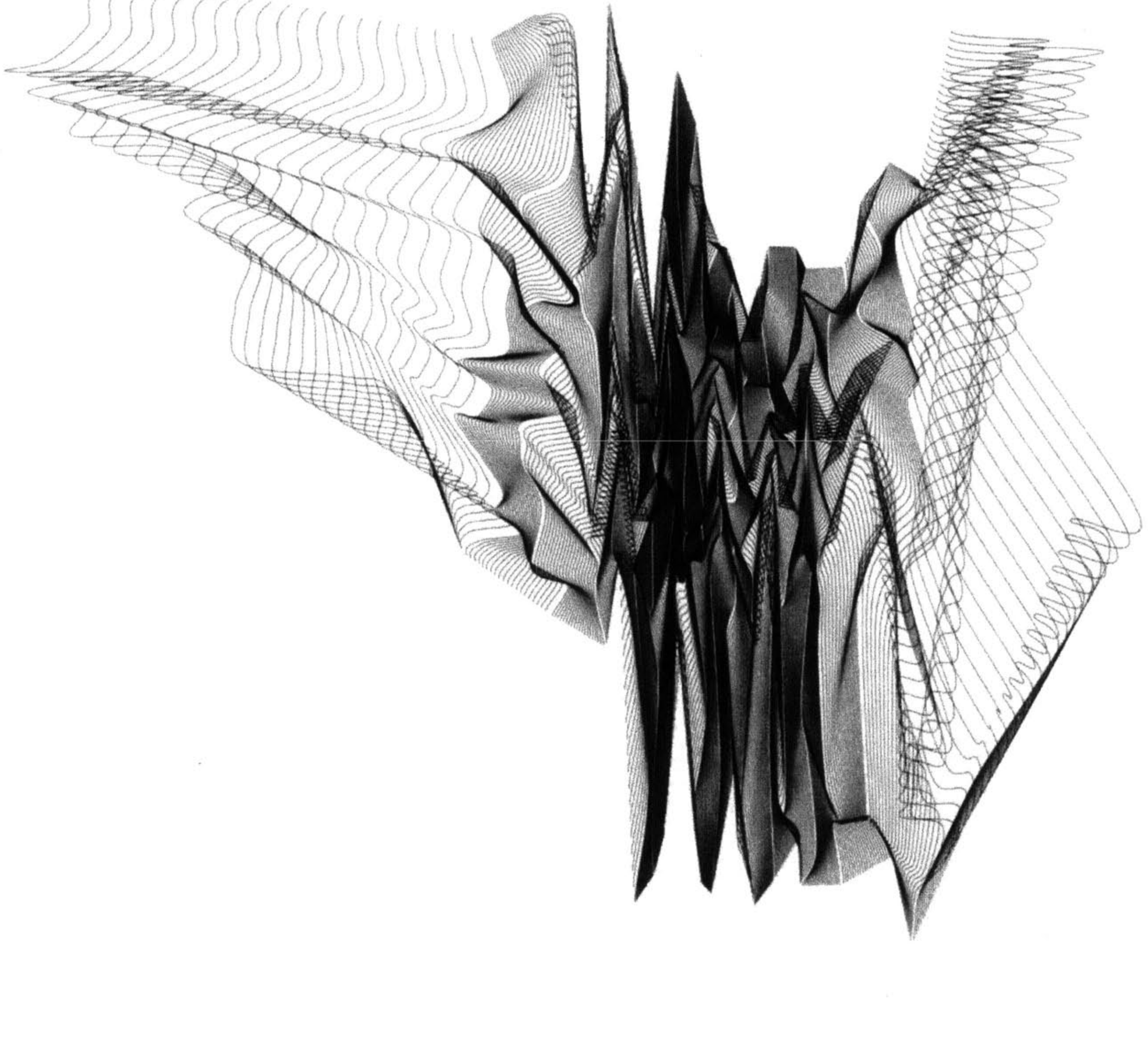


K-002-090



K-002-130: Swarming Striations at time 130

Method 'K' uses an underlying set of points as the structure of each drawing. In series 002, the point positions are subject to swarming behavior. Sweeping laterally across the field groups the points according to their x position. Each group is then sorted according to their y position and a single line is drawn through the sorted points in each group. Every 1 in 50 lines is omitted. By run 130, the points have formed a swarm near the center.



K-002-130

References

Gershenfeld, Neil. The Nature of Mathematical Modeling. Cambridge University Press, 1998.

Reynolds, C. W. "Flocks, Herds and Schools: A Distributed Behavioral Model." In ACM SIGGRAPH Computer Graphics, 21:25–34, 1987.

3. Evaluation

This section aims to provide evaluation of the drawings that resulted from this project. A system of features found in drawings is offered: emergent line and gradated tone. Then, more broadly, an admittedly subjective system for interpreting drawings that biases and foreshadows architectural implications is presented and a non-exhaustive set of reference drawings are discussed.

3.1 Emergent Characteristics

First, it's important to note that these features are defined as characteristics that are visible in the drawings but are not the direct result of a programmed action. These feature definitions must therefore take into account the computational structures of the apparatus so that their direct implications may be filtered. This is a distinction from a process of interpretation that would read the drawings as isolated and autonomous objects (which is valid, though distinct, and is partially explored as analytic device in section 3.2)

The first salient feature of these drawings is the most elemental: the emergent line. This feature notes visible lines which are not associated with a single mark nor those which are defined computationally. The most basic and obvious demonstration of this are drawings in which sets of lines can collectively be read as a single line. This feature is the most prevalent in the drawings of this project. Nearly all the drawings exhibit an emergent line to some degree. It is most saliently identifiable in the early methods. Within many of the individual swatches in A-002-001, for example, vertical lines are legible along the corresponding "pause points" of the marked lines. In A-004-002 and A-005-001, a single, though soft, vertical line is legible between the territory of the drawing in which the marked lines are relatively consistent and those in which the pen, because it is deteriorated, leaves a fleeting mark. Drawing C- 003-001 exhibits a number of emergent lines. Dark and light arc lines appear prominently in the left portion of the drawing. More subtle lines mark the edges of darker zones.

Tone in plotter drawings is always at least slightly indirect as it must be constructed out of lines. In some cases though, tonal variation is expected and was purposefully built into the computational definition of line. (Series C-004 is an example of this in which the drawing of lines toward a global focal point inevitably leads to denser lines near the center and space lines near the periphery) The drawings that are identified here have tonal variation that was initially unintentional or emergent. Gradated tone is specifically identified as a valuable feature due to its capacity to imply form. (discussed in section 3.2) The gradation may manifest either as a shift in value or hue. The salient and emergent tonal gradation in series A-007 was surprising. The variation in line darkness from start to finish was more than expected and, as a result, rows with lines spaces densely exhibited a distinct vertical variation from dark to light. Drawing C-004-001 exhibits tonal variation in between light and dark areas on the page. Lighter areas are found at the top and bottom, with a gradual shift toward darker areas at the center. The D-002 series tends to exhibit gradation of value and in the cases where two pens are used (D-002-006, D-002-007 and D-002-008) gradation of hue.

3.2 Depth

This subsection considers how the features discussed above, emergent line and emergent gradated tone, contribute to legible depth. Depth in drawing is the first step towards representational capacity. Tending toward representation can occur with respect to established conventional modes. (when a drawing can be read as a plan or section, for example) In that capacity, a system is projected onto the drawings, which allow for the interpretation of three-dimensional information. The kind of representation that should begin to register with these drawings is independent of convention. Stepping outside of convention requires a capacity to express depth without codes. Depth expression is defined by three types: environmental depth, formal depth, and material depth. These three types range from the most broad and macro (environmental) to the most narrow and micro. (the material)

Environmental Depth
In C-003-001, a horizon line becomes legible as a result of two vertical tonal gradients, which peak in darkness at the center. Environmental depth can also occur in drawings where the tonal gradation overshadows the legibility of individual lines. Such is the case with G-002-682 and portions of G-002- 380. In these drawings, some boundaries are so loose and gradual that no outline or edge is visible despite the clarity zones that share a common level of value.

Formal Depth
This kind of depth occurs as the lines register as edge or contour of a three-dimensional shape. This can correspond directly to an algorithmic constraint, as in the case with C-004-002 and C-004-002, where the use of a focal point suggests a perspective structure. The density of the lines tends to suggest the primacy of a surface, where each line is a seam and the edge lines are the result, again, of close proximity and only slight variation from one line to the next. A-007-001 and A-007-002 demonstrate formal depth with in a more controlled space. Without any focal point, the depth is manifest exclusively as a partial result of tonal gradation. Each row registers as a thick line, imbued with implied surface area.

Material Depth
This kind of depth is the result of a fine registration of noise or pattern. This material is unique among the three depth types in that it can occur at the surface of the paper. The texture of the paper does, or does appear to, have variation in depth. The K-002 series documents a range of depth conditions from the formal to the materiel. At the later runs, the registration is more of material depth, as the In series-002 the formal lines are imbued material depth independent of the paper surface. The materiel depth is a geometric condition.

3.2 Slippage between space of page and representational space

A discussion of depth type leads to a discussion of space. In the context of a two-dimensional drawing, the registration of depth indicates a third dimension. In the case of formal depth, this third dimension extends, to some degree, in the empty volume around the form. Without futilely debating interpretation, it can simply be noted that a formal figure always exists within a spatial ground. But what of the space of the paper? The registration of the material presence of the paper, and its flatness, is in conflict with the registration of depth. The viewer can become an active participant in shifting focus from the page to a deeper point "inside" the drawing, but depth cues of the drawing can suggest, if not outright command, the primacy of paper or deep space. The use of this characteristic slippage between two conflicting perceptual states as a factor for evaluation is aimed at positioning drawings relative to architectural representation and theoretical discourse.

In the cannon of architectural drawing, the simultaneous presence of paper and deep space is evident in a highly celebrated set of works by Daniel Libeskind titled ,"Micromegas"[1]. Jeffrey Kipnis, in his curation of the "Perfect Acts of Architecture" exhibition, describes Libeskind's graphite series relative to the subsequent ink prints. "...the pencil drawings are far more stirring... While empty page is confined to the border of the chock-full prints, it trespasses unchecked into the pencil drawings. The zoo of torrid, deep-space graphic effects–rifts, clefts, fissures, fractures, crevices, arroyos, nooks, chasms, vorticies and holes that Libeskind sculpted in the Micromegas prints with his virtuoso linework–lies in the drawings against the dry shallow of the pages with vertiginous result, the shallow further desiccated by the translucence of the graphite line."[2] Libeskind's drawings rely on the inconsistent conformity to local three-dimensional logics–deploying linear relationships that directly and indirectly imply axnometric or perspective projection. A flattening effect occurs as competing spatial systems convolve or lines break from their role as figural boundary and regain an referent-free autonomy. The imperfections of graphite and its translucency reinforce the paper material of the drawing. Libeskind's drawings further cultivate a loose formal vocabulary by violating a consistent rendering of 3-D forms as occluding solids. Even the most legible three-dimensional forms "in" Micromegas tend toward linear extremes. Again, this participates in establishing a balance between the deep and the shallow. Libeskind's drawings occupy unique territory in that they are drawing projects, not representation. Kipnis asserts "...the architect's signature and notion on each drawing indicate his confidence in them as works in their own right."[3] Still, they operate architecturally, with considerations of implied territories, joints, frames, and material mass.

Libeskind produced a later set of drawings,[4] which tend to behave within the space of paper. This series deploys fields of dense lines and gradated zones of transition between the legible individual line and emergent collective line. The occasional tendency to suggest formal depth is still apparent. More prominent, though, is environmental depth, as line spacing is subjected to

[1] Libeskind, Micromegas.
[2] Zenghelis, Kipnis, and Lowry, Perfect Acts Of Architecture.
[3] Ibid.
[4] Libeskind, Chamber Works: Architectural Mediations on Themes from Heraclitus.

gradually increasing variation. If this set of drawings is indeed a "perfect act of architecture" then it is evidence that the language of architecture can reside entirely in the domain of the line. This series also suggests a continuum between serialized art (or more broadly, computational art) and architecture. If Libeskind's earlier Micromegas began as a set of established architecturally referent linear systems, which break down as they collectively manifest in the drawing, this the drawings created as part of this project are the converse: beginning as line and building up the legibility of form and space.

Libeskind's shift from architecture legible as line to line legible as architecture invites a reference to the contemporary work of artist Daniel Zeller. In multiple examples, the character and behavior of Zeller's line operates exclusively in the two-dimensional domain of paper, but inevitably leads to material, formal and environmental depth. Artists Peter Peri[5] and Frances Richardson[6] draw in similar manners with linear interaction at the level of the paper clearly present while also contributing to a generative depth. In contrast to Zeller, who began his career as a sculptor, the work of Peri and Richardson exhibits a shift between paper and an environmental and material depth without a suggestion of form.

The potential to read flat lines and deep lines in the same drawing can be deployed a representational asset in architectural design. In 2000, ShoP Architects represented their winning proposal for Dunescape at P.S.1 MoMA with a set of drawings and renderings[7]. The salient drawing, which became emblematic of the project, later appeared on the cover of "Versioning: Evolutionary Techniques in Architecture."[8] Tied to a strategy of design and fabrication, the two-color linear system refers to edges of three-dimensional form. Red and blue lines correspond to independent terrain-like forms, divided into a series of 2-inch strands, which oscillate and are controlled by key sectional profiles. The lines also interact in the visual field of the image. Red and blue systems convolve, and because the lines indirectly convey degrees and types of occupation, this emergent purple figure has diagrammatic meaning.

3.2 Projections

In projecting this work–the apparatus, the drawings, and the evaluation–forward, it is most

[5] Relevant works include *Octave*, and *Hare*

[6] Relevant works include *C 1 0230502, 111298* and *110698*

[7] Sharples, Holden, and Pasquarelli, SHoP.

[8] Pasquarelli, Versioning.

relevant as a collective contribution. The apparatus is not intended for redeployment as a tool. Likewise, the drawings, though functional as works of art, are more valuable within the context of their making and their evaluation. Design education is the suggested territory in which this research would be most valuable and relevant. Faced with the urgent task of folding technical (or instrumental) knowledge into the design-knowledge-based economies of traditional design curricula, this work offers a model for a type of approach that emphasizes experimental representation. Furthermore, drawing is offered as a particularly ripe mode of operation for this experimentation.

References

Libeskind, Daniel. Chamber Works: Architectural Mediations on Themes from Heraclitus. Graphite, 1983.

———. Micromegas. Graphite, 1978.

Pasquarelli, SHoP/Sharples Holden, ed. Versioning: Evolutionary Techniques in Architecture. 1st ed. Academy Press, 2003.

Peri, Peter. Octave, 2003.

Richardson, Frances. C 1 0230502, 2002.

Sharples, Holden, and Pasquarelli. SHoP: Out of Practice. The Monacelli Press, 2012.

Zenghelis, Elia, Jeffrey Kipnis, and Glenn Lowry. Perfect Acts Of Architecture. Edited by Terence Riley. The Museum of Modern Art, New York, 2002.