DRTP 2 (1) pp. 123–127 Intellect Limited 2017

Drawing: Research, Theory, Practice
Volume 2 Number 1
 doi: 10.1386/drtp.2.1.123_7

MALCOLM RIO
Maryland Institute College of Art

The definition of line

Keywords
computation
drawing
line
algorithmic
process
vector

Line is the most rudimentary delineation of space: a perspective drawing begins with vanishing points along a horizon. Photographers and painters use lines to direct a viewer's gaze between the foreground, mid-ground and background of a composition; architects are obsessed with revealing and concealing spaces with lines of sight and movement. With a single line, the notion of space enters our reality, alluding to a 'there' and 'here' in which the line is read as an edge or boundary of and object or field. But line can also represent movement, connection or a positioning and locating: a vector. These lines communicate, even in their most ambiguous representation, a direction and magnitude between two points, representing the measure between two dichotomous positions in space: a perspective.

During my time as a graduate student in architecture at Rhode Island School of Design (RISD), I was given the opportunity to investigate the definition of line in a computation and representation advanced-studio led by Carl Lostritto, assistant professor at RISD. Through artistic experimentation and research, we studied and analysed the behavioural, formal and computational logics inherent in drawing and reading lines (Lostritto 2014). The studio oscillated between analysing the theoretical works of Marco Frascari, Deanna Petherbridge, Jesse Reiser and Nanako Umemoto, William Empson, Colin Rowe and Robert Slutzky, and Jeffery Kipnis among many others, and responding to these texts through computational methods of drawing. The two drawings featured here were

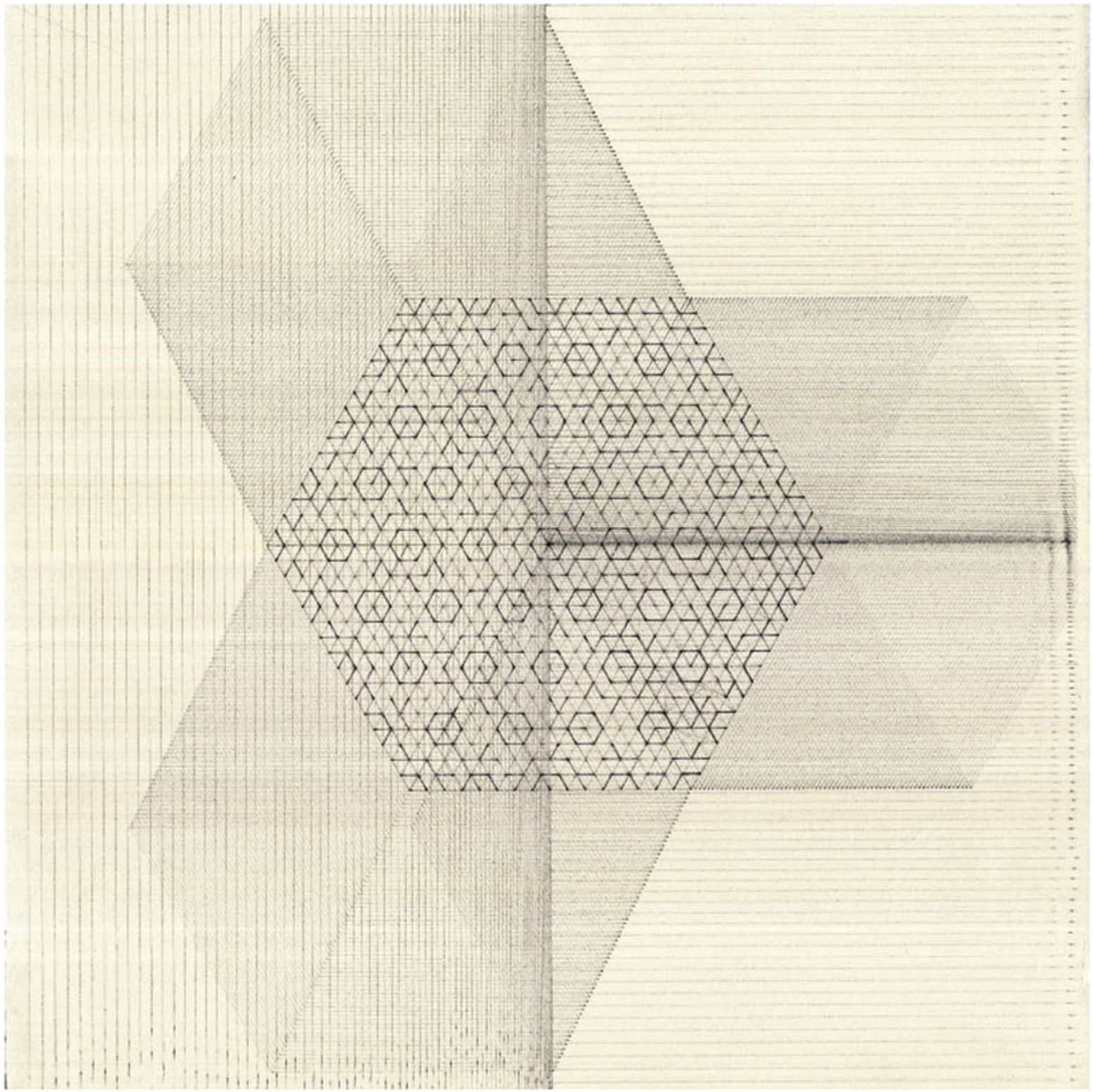
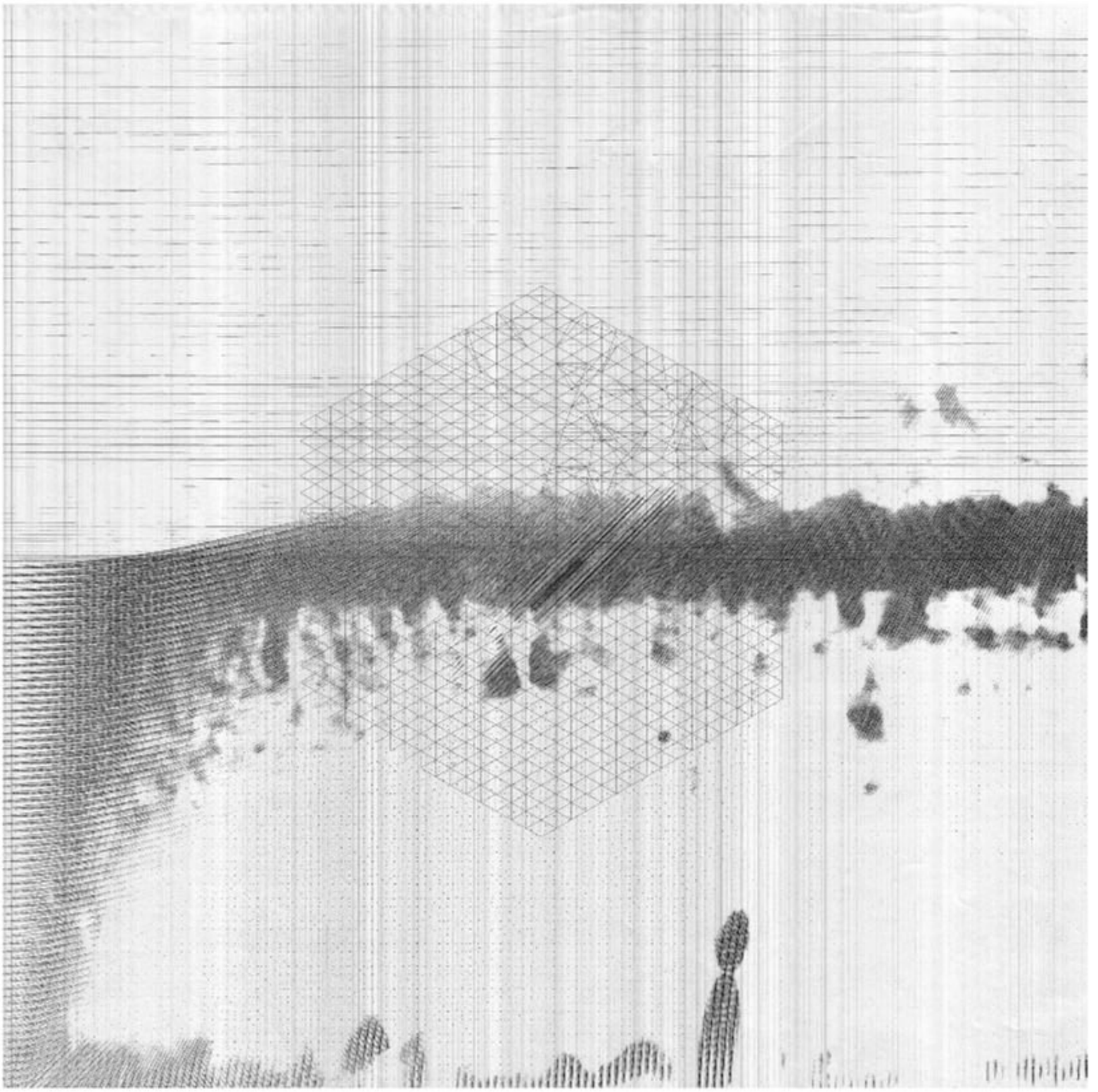


Figure 1: Malcolm Rio (2014), Algorithmic Drawing: Study #4. *© Malcolm Rio.*

Figure 2: Malcolm Rio (2014), Non-Repeatable Algorithmic Drawing. © *Malcolm Rio.*



produced through writing code in the Python's programming language using Pygame for graphical representation.[1] Pygame served as a medium to visually interface with and test the results of my lines of code before translating into Chiplotle as syntax to output with a Roland pen plotter.

Rather than referencing an architecture, these drawings explore the reading of space through ambiguity, pattern and the capacity of line to imply depth. Referencing surrealist techniques like those used by Jean Arp, René Magritte, Man Ray, Yves Tanguy and Paul Klee (Baron et al. 1999) I developed a script that used recursive loops to reference existing geometry and produced scaled versions that propagated in a fractal manner, blurring the notion of ground, perspective and subject. Exploiting standard conventions of axonometric projection, these drawings initially present themselves as objects within an undifferentiated field, challenging notions of ground by not privileging any page orientation. At closer resolution, the apparent object-ness dissolves and flattens into an interplay of lines and space that resist coalescing into figure and ground or a singular perspective. Beyond merely existing as an object, these drawings engage in notions of 'bigness' (Koolhaas and Man 1995) in novel ways, not on the scale of grand gestures but through non-repeating patters, self-referentiality and collective agglomeration rather than coherent wholeness.

Attention to a line's behaviour, formal character and computational logics was focal to my process. Behavioural logics considered whether a line was read as figure or ground based on its relations to others' lines and the page's boundaries, combined with a line's formal logic – weight, style and brush stroke – to vie with spatial elements of gravity, proximity and time. Computational logics referred to the math necessary for drawing each line self-referentially. These three synthesize into a combinatory logic that invites the eye and mind to enter into a discourse with line and space whether via the digital mediation of pixels across a screen or via the physical mark that results from a cause and effect between material, pressure and surface.

My methodology consisted of manipulating simple local relationships to test their ability to produce diverse global patterns/anti-patterns. The lack of control over the entire programing language in which I was operating and efforts to replicate technical glitches in physical drawing production brought about questions of authorship in my drawing process. As implied in its name, *Non-Repeatable Algorithmic Drawing* excepts and integrates the unpredictable and serendipitous into the drawing's creation to engage with the material reality and environmental conditions of the computational ecosystem, disciplinary conventions and physical limitations usually collapsed uncritically into 'output'. *Algorithmic Drawing: Study #4* represents the idealized patterns prior to the introduction of strategic relinquishing of total authorship that disrupted the modernist faith in fidelity between digital screen and the physical plotted drawing. The prescribed limitations within Python's programming language to control a drawing machine raised question whether it was valid to claim 'I made' this drawing.

1. Python is a general-use programming language (PL) that is commonly used because of its readability and condensed writing style, unlike other PLs. Specifically, it is 'an interpreted, object-oriented, high-level programming language with dynamic semantics' (Python Software Foundation (2001–16). Pygame is a module within the Python's PL that is geared towards writing games and multimedia programs. Its function is to condense the lines of code necessary for defining particular elements associated with game design – i.e., shapes and geometries, camera functions, colour, joystick control, text and cursor functions.

References

Baron, Jacques, Krauss, Rosalind and Baum, Timothy (1999), *Surrealism: Two Private Eyes: The Nesuhi Ertengun and Daniel Flipacchi Collections*, New York: Solomon R. Guggenheim Museum.

Koolhaas, Rem and Man, Bruce (1995), 'Bigness and the Problem of Large', in R. Koolhaas and B. Man, *S, M, L, XL*, New York: Monacelli Press, pp. 494–516.

Lostritto, Carl (2014), 'RISD Architecture Studio: Computing Drawing', http://lostritto.com/risd-2014spring. Accessed 30 October 2016.

Python Software Foundation (2001–16), 'What is Python? Executive summary'. Python Software Foundation, https://www.python.org/doc/essays/blurb. Accessed 15 November 2016.

Suggested citation

Rio, M. (2017), 'The definition of line', *Drawing: Research, Theory, Practice*, 2: 1, pp. 123–27, doi: 10.1386/drtp.2.1.123_7

Contributor details

Malcolm Rio is a graphic and architectural designer from Amherst, Massachusetts, USA. He is currently a faculty fellow at the Maryland Institute College of Art (MICA), where he teaches graphic design, architectural design and foundational studies. Before coming to MICA, he attended Rhode Island School of Design on a graduate fellowship for his Master of Architecture. He was recognized by the department with the 2015 Alpha Rho Chi award. His thesis explored the overlap between cybernetics, the structuring of bodies and architectural space. His undergraduate degree project focused on the intersections of image, identity and power, earning him Towson's 2012 Scholarship for Environmental Initiatives. He also holds a Bachelor of Science in philosophy and Bachelor of Fine Art in art and design from Towson University.

E-mail: mrio@mica.edu
Web address: www.malcolmrio.com