Wendy Hui Kyong Chun, *Programmed Visions: Software and Memory*
Ceci Moss and Lisa Gitelman
New Media Society 2012 14: 359
DOI: 10.1177/1461444811429927d

want a flyover media history to teach critically to undergraduates, and those willing to appreciate general theories for both the noble projects they begin as and for the occasionally productive messes they leave behind.

References

Gleick J (2011) *The Information: A History, A Theory, A Flood*. New York: Pantheon.

Mills CW (1959/2000) *The Sociological Imagination* (40th anniversary edn). New York: Oxford University Press.

Wendy Hui Kyong Chun
Programmed Visions: Software and Memory. Cambridge, MA: The MIT Press, 2011.
xiii + 239 pp. ISBN 9780262015424, $32.00 (hbk)

Reviewed by: Ceci Moss and Lisa Gitelman, *New York University, USA*

Software is everywhere. *Programmed Visions: Software and Memory* is part of a growing literature that seeks to understand the meaning of software and its power. Chun starts with the proposition that software is a thing, that computer programming has 'hardened' though processes related to its commodification into a particularly powerful kind of thing. Its metaphors are legion – hard and soft, for example – while it works itself as a kind of *uber*-metaphor, since the logic of programmability sprawls beyond software in the guise of neoliberal governmentality. Chun understands the computer interface and its masterful, 'so-called users' as symptomatic of a larger instrumentalization of knowledge (p. 17). In the rush to capitalize on knowledge, temporality gains new value and the future becomes a commodity that can be bought and sold, and with it, the past. Chun sees 'computers as capable of being the future, because, based on past data, they shape and predict it' (p. 9). Crucially, data are mined in order to foretell that future and organize the past towards its subsequent application. The title of the book is a reference to these 'programmable visions.'

This book is split into two parts. The first addresses the active and fetishistic quality of software, its confusion with *logos* and how this renders it 'invisibly visible (or visibly invisible)' (p. 15). The second section expands the conversation to consider a history interwoven between biology and computers, demonstrating ways that that the logic of programmability undergirds and connects the two, making them 'complementary strands of a double helix' (p. 10). Both of the book's parts – each with its own introduction and containing two chapters – are framed by an overall introduction, conclusion, and epilogue, while punctuated by four short meditations relevant to the whole. Like software, this book has a lot of layers. Chun is a talented reader throughout, tackling the metaphors of computing all the way down to the metal. She reads circuitry, she reads logos, and she reads everything in between. Written with a high degree of abstraction and with a sometimes incantatory style, *Programmed Visions* nonetheless visits some of the classics of computing history.

The first chapter, 'On Sourcery and Source Codes', pursues a descriptive or ontological project aimed at understanding what code is and the ways that it is and has been mystified. Here software as supposed *logos*, as fetish, and as an almost mystical motor for action or 'sourcery' are notions that get traced back to the roots of computing: to military research and development in the USA. In two subsections, 'Yes Sir!' and 'Bureaucracies Within the Machine,' Chun discusses the command and control structures of software languages in relation to their application to female operators or ‘coders’ and male commanders working with ENIAC in the 1940s and SAGE in the 1950s. In each case gendered workflows, separate yet interrelated, gave way to an abstracted interaction with the machine, one in which the programmer both knows and remains ignorant of its operations, and hence is dually empowered and enslaved.

Chun’s second chapter, ‘Daemonic Interfaces, Empowering Obfuscations’, makes the compelling argument that a computer is a metaphor for metaphor itself, in that it functions as a means by which other metaphors are filtered and arranged. Computers envisioned as ‘universal machines’ become universal precisely because of their power to transform anything into anything, effectively acting as a ‘transference machine’ animating input and output. Software as an ‘invisible program that governs’ affects representation across multiple cultural arenas as well as the process and logic of governance (p. 57). Nowhere is the metaphoric aspect of computing more explicit than the interface, which Chun takes as her focus in this chapter. The direct and instant temporality of ‘real time’ interfaces, which she traces back to well-known work of Vannevar Bush and Douglas Englebart, have emerged as a response to the postmodern drive to map and be mapped. Throughout the chapter, Chun asks what a truly critical reaction would be to constant mapping, one that would be wary both of nostalgia and a liberal notion of subjectivity. She concludes that any such move would have to inhabit mapping from the interior, through an engagement with the very artificiality of metaphor.

The remaining half of the book turns to examine computer memory and its connection to programmability by drawing out the close relationship between biology and computing. The question of the archive and memory take center stage in these sections, especially how something as ephemeral as new media can achieve an illusion of permanence by way of repetition and transmission. Chapter 3 looks at the development of Mendelian genetics with the aim of complicating the familiar narrative that genetic science came out of computer science; rather she situates them as dual instances of biopolitical rationality. Beginning with a close reading of Erwin Schrödinger’s *What is Life?*, Chun observes that the desire to assemble an archive of data (in the case of genetic science, of DNA) to chart an invisible line of causality in order to predict future outcomes is an archaeological phenomenon, in the Foucauldian sense. For DNA and code alike, the visible and the articulable are fed through a logic of programmability, wherein language becomes an active generator for what becomes visible.

Chapter 4, ‘Always Already There, or Software as Memory,’ looks at how software has achieved a semblance of permanence, despite its vapory and ‘soft’ quality. Devoting her attention towards John von Neumann’s architecture for stored computer memory, she asserts that, despite their differences, memory and storage begin to become conflated at this moment through analogies to analogies, wherein analogies are extended out into the world in order to read it, and thus, make it familiar. This process is reflective of a yearning for fixity, which is realized in the stabilization and optimization of hardware. The digital, like software, contains an axiomatic property, which gives it a semblance of having always existed. The elusive work of repetition and regeneration – or what Chun terms the ‘undead of information’ – that allows for computer memory to operate becomes obfuscated, and we interpret our computers as archives, enduring and permanent.

An epilogue, ‘In Medias Race’, points to the unfinished work of *Programmed Visions*, since Chun identifies ‘a thread that has been largely invisible and yet central’ to the book. This thread consists of ‘parallels between *software* and *race*’ (p. 179). Readers will look forward to more on that subject.