Wendy Hui Kyong Chun

Programmed Visions: Software, DNA, Race

Why are images proliferating at a time when their power to index reality is waning? How have non-transparent technologies, such as computers, become conflated with transparency? *Programmed Visions* argues that the answer lies in the surprising emergence of software and the transformation of race from a biological to a cultural category in the 20-year period from 1943-1963. It contends that these phenomena reveal fundamental shifts in the relationship between what is visible and invisible, imaginable and readable. They indicate a simultaneous decline in and frenzy of visual knowledge.

Software—unforeseen by the early computer designers and non-existent at a physical level—has become a privileged way of explaining the relationship between nature (hardware) and culture (software), the operations of heredity (genetic program), and the functioning of narrative (rhetorical software). Software's power stems from its programmability and from the ways it concretizes causality: high-level procedural languages reduce language to a series of imperatives, which in turn generate visible effects in an invisible yet understandable manner; the general structure of software facilitates "black boxing," since each level can be analyzed independently. Race was, and still is, a privileged way of understanding the relationship between the visible and invisible: it links visual cues with unseen forces. Although post WWII it no longer credibly links physical differences with innate mental differences, race remains a valid category. In the work of population geneticists, racial groups have become "breeding populations"; in the work of molecular biologists, racial groups are defined by the probability of having a combination of mainly unexpressed genetic material. Culturally, race has become more on display than ever, even as the question of what race indexes—cultural or genetic differences, the results of economic injustice—remains unresolved. Race and software thus mark the contours of our current understanding of visual knowledge as "programmable visions."

In examining software and race, *Programmed Visions*, does not focus on them as simple causes. Rather, drawing from Michel Foucault's archaeological methodology, it examines what made race and/or software necessary and possible. *Programmed Visions* draws from a wide array of texts and phenomena, from Claude Shannon's "A Mathematical Theory of Communication" to Sadie Plant's *Zeroes and Ones*, from analogue computation to the NAACP's response to the lynching of Emmett Till, in order to map the contradictory relevance of visual knowledge.

Chapter One: On Software, or the Persistence of Visual Knowledge
This chapter examines software's perpetuation of "visual culture" by analyzing its striking parallels to both ideology (something which does not exist, yet structures our vision) and ideology critique (the act of revealing the invisible driving the visible). These parallels are mainly due to "automatic" higher-level programming languages, which encapsulate the gendered command and control structure of WWII computation.

Chapter Two: The Racial Imperative
Building on work by Lisa Gannett, this chapter traces the persistence of race within the work of population geneticists, such as Theodosius Dobzhansky. It links race to software by revealing structural similarities between them as forms of visual knowledge, as well as similarities between software and genes (both enable a separation of pattern from form), and between genes and race (the more they cannot be rigorously defined, the more powerful they become).

Chapter Three: Building Analogies
Analyzing analogue computation and information theory (specifically the work of Vannevar Bush and Claude Shannon), this chapter argues that the move from analogue to digital computing did not simply stem from the inadequacies of analogue computing, but also from the increased control enabled by discrete programming languages. It also maps the transformation to "information" that accompanied the move from analogue to digital computing—with stored program computing, message and signal became one.

Chapter Four: Automatic Memory
This chapter addresses the two drives towards automation important to the development of software: John von Neumann's theory of automata and Grace Murray Hopper's work with automatic computing. Treating von Neumann's "First Draft of a Report on the EDVAC" and his "The General and Logical Theory of Automata" as two points in a trajectory, this chapter traces the ways in which von Neumann's interest in self-reproducing automata drove him to emphasize software in his later years. The second half of this chapter outlines Hopper's technical and programmatic work with "automatic programming" (what we would call programming today), explaining how automatic programming—using the computer itself to program—has been key to the emergence of high-level programming languages as repeatable, machine-independent marks.

Chapter Five: The Genetic Programme
Focusing on the work of François Jacob, this chapter explores the importance of stored-memory programming to early explanations of gene regulation. Drawing from the work of Lily Kay and Richard Doyle, it reveals Jacob's quick and contradictory movements between hardware and software (regulatory circuits and genetic message as program).

Chapter Six: Race as Spectacle
This concluding chapter traces the continuing relevance of race at a time when its function as a biological index was largely discredited. Looking at the proliferation of racial images and the struggle over their meaning in the 1950s and early 1960s (such as images of Emmett Till's open coffin), it argues that the very undecideability and insufficiency of race as a form of visual knowledge led to its visual proliferation, as a struggle ensued over what racial images signified. Contrasting the 1950s UNESCO statements with early civil rights documents, it investigates the curious dovetailing of academic repudiations of the biological relevance of race with the struggle over the devastating effects of "separate yet equal."

Epilogue: Language as Technology
This epilogue makes explicit *Programmed Visions*' implicit engagement with critical theories of language, tracing the ways in which the notions of software and information have infiltrated the language of thinkers key to this project, such as Michel Foucault and Jacques Derrida.

Chapter one has been published in *grey room*18; I have completed the initial research for all the chapters, and have presented portions of Chapter two, five and six at the Modern Languages Association, Duke University and at the Humanities Center at Harvard respectively.