Jacob Gaboury

Queer Affects at the Origins of Computation

Much has been written on Alan Turing and the origins of artificial intelligence (AI) some seventy years ago. Turing's "imitation game" set the foundation for research into what has become the future promise of nearly all AI-driven industries today.[1] At the heart of Turing's work is the notion of intelligence as performative, that is, as an effect that need not demonstrate any internal awareness of intelligence as an abstract or conceptual goal. Turing famously likened this performative quality of intelligence to gender, which he imagined as equally transmutable and inessential—a comparison that opens up the possibility of a queer reading of AI through the discourses of performance, language, and affect. Nonetheless, in our hagiographic treatment of Turing as the so-called father of modern computing, we often miss those queer objects and relations that constitute the broader milieu of experimental mathematics during this period. Working alongside Turing at the University of Manchester Computing Center in the early 1950s was a gay man named Christopher Strachey. A prolific early programming language designer, Strachey is best known for developing what are arguably the first examples of computer music and computer games, along with a love letter–generating algorithm that is widely considered the earliest work of computational art. That Strachey developed so many groundbreaking programs at the precise moment Turing was theorizing the foundations of artificial intelligence speaks at once to his skill as a researcher and to his mutual interest and investment in experimental or non-normative uses for computational

1 Alan Turing, "Computing Machinery and Intelligence," *Mind* 59, no. 236 (1950): 433–460.

Jacob Gaboury, "Queer Affects at the Origins of Computation," *JCMS* 61, no. 4 (Summer 2022): 169–174.

technology. While their colleagues worked on applications in optics and aerodynamics, Turing and Strachey approached the computer with a distinctly different set of affects and investments, asking the machine to perform not only intelligence but also play, sincerity, camp, and even love. Examining the history of early computing through these two queer figures allows us to mark out a set of affective relations toward computational machines that presage the contemporary moment while critiquing our own investment in the normative intelligence of artificial systems.

In looking for a queer origin to the history of computation, nearly all scholars are drawn to the figure of Turing, considered by many to be the originator of modern computer science and arguably its most visible queer subject. As I have discussed elsewhere, Turing is a unique and captivating figure due in part to the visibility of his difference and the tragedy of his death.[2] While not a secret, Turing's sexuality was not widely acknowledged within computer science and mathematics for many years. Following the publication of Andrew Hodges's definitive biography of Turing nearly thirty years after his death, Turing became a figure of fascination both for his work in defining the function and limits of computational systems and for the ways he indexed a culture of early-twentieth-century sexuality and homophobia.[3] This commingling of the personal, political, and technical in Turing's work begins with Hodges, but it has subsequently gained traction among researchers invested in queer history and Turing's influence on the political claims of modern computer science.

Turing's most noted work in this regard is his widely influential "Computing Machinery and Intelligence," first published in 1950 while he was a researcher at the University of Manchester developing some of the earliest modern digital computers.[4] It is here that Turing first proposed the evocative question "Can machines think?" and argued in favor of machine intelligence through a reframing of thought as the successful performance of intelligent behavior. To make his case Turing proposed an imitation game that has come to be known as the Turing Test, whereby an examiner seeks to ascertain whether either of two unseen respondents is a machine based on their answers to a series of simple questions. Here Turing locates thought within a performative theory of intelligence, suggesting that if a machine can successfully emulate thinking by answering questions in a way that is indistinguishable from a human participant, then it has demonstrated a functional intelligence. Rather than weigh down this claim with ontological concerns over what thinking or intelligence are, Turing instrumentalizes thought as a presentation of passing, a successful rendering of the social and intellectual

2 Jacob Gaboury, "A Queer History of Computing: Part 1," *Rhizome*, February 19, 2013, https://rhizome.org/editorial/2013/feb/19/queer-computing-1/.

3 Andrew Hodges, *Alan Turing: The Enigma* (London: Simon and Schuster, 1983). For a discussion of Hodges's work on Turing and the rediscovery of Turing's sexuality, see Jacob Gaboury, "A Queer History of Computing, Part Five: Messages from the Unseen World," *Rhizome*, June 18, 2013, https://rhizome.org/editorial/2013/jun/18/queer-history-computing-part-five/.

4 There are many competing claims for the first modern computer, but several prominent features are that the machine be digital, electronic, and stored program. The early Manchester computers satisfy each of these criteria. Turing, "Computing Machinery and Intelligence."

cues expected of a human subject.[5] Already this work is surprising, less for its dramatic claim that machines could one day satisfy this imitation game than for the ways it refuses an essentialist notion of subjectivity, identity, and internality in favor of an outwardly presentational subject.

This claim is even more surprising when taken in its full context. While most contemporary Turing Tests are designed to assess the performativity of a generalized humanity, Turing's original test is an explicitly gendered one, in which the control for performativity is that of gender performance. That is, interrogators are asked to determine the gender of the game's unseen participants, not their humanity. This gendered language continues throughout, inflecting Turing's treatment of the computer and the gendered context of computation in this early period when most "human computers" were women performing high-level calculation by hand.[6] As Patricia Fancher notes in her work on Turing's embodied rhetorics, there is a queer valence to this thinking such that if we are to read Turing literally, "machine intelligence is like a man pretending to be a woman."[7] Moreover, Turing places bodily experience as central to machine intelligence, imagining a host of activities that he qualifies as intelligent, which a machine could not do and would struggle to perform: "fall in love, enjoy strawberries and cream, make someone fall in love with it, learn from experience, use words properly, be the subject of its own thought, have as much diversity of behavior as a man, do something really new."[8] Once again Turing seems to refuse a normative understanding of intelligence in favor of a deeply embodied and often gendered understanding of human behavior as performative, relational, and contextual. Turing's list is at once beautiful in the way it evokes a particular notion of human experience and intelligence and significant in that he does not discount the possibility that a machine indeed might do each of these things, particularly if we expand our notions of what computation is capable of and what both human and artificial intelligence might be.

Examining Turing's provocation, it is striking how directly it maps onto the work that he was undertaking at precisely this moment alongside Christopher Strachey. Known as "the man who wrote perfect programs" at a time when programming was an exceedingly difficult and error-prone process, Strachey's had a far from conventional road to computation. As nephew to the critic and biographer Lytton Strachey, Christopher was raised at 51 Gordon Square in proximity to Virginia Woolf, Clive Bell, and the other members of the Bloomsbury Group of writers, intellectuals, and philosophers. Despite this privileged background, Strachey did not meet with academic

5 For a discussion of the Turing Test and the theory of passing, see Jeremy Douglass, "Machine Writing and the Turing Test" (presentation, Alan Liu's Hyperliterature seminar, University of California, Santa Barbara, 1999), https://web.archive.org/web/20010525032059/http://www.english.ucsb.edu/grad/student-pages/jdouglass/coursework/hyperliterature/turing/#_Toc510202769/.

6 See Mar Hicks, *Programmed Inequality: How Britain Discarded Women Technologists and Lost Its Edge in Computing* (Cambridge, MA: MIT Press, 2017); and Jennifer S. Light, "When Computers Were Women," *Technology and Culture* 40, no. 3 (1999): 455–483.

7 Patricia Fancher, "Embodying Turing's Machine: Queer, Embodied Rhetorics in the History of Digital Computation," *Rhetoric Review* 37, no. 1 (2018): 98.

8 Turing, "Computing Machinery and Intelligence," 453.

success as a young child, and he suffered a breakdown in his second year at university while coming to terms with his homosexuality. While Strachey had hoped for a career in academia, he had neither the grades nor the disposition for a prominent fellowship, and so following graduation he spent over a decade as a teacher and later schoolmaster of young children at a number of lower-ranking institutions. Beginning in the late 1940s, Strachey learned of several computing machines being developed by Turing and others at the University of Manchester. Strachey had met Turing socially several years prior at King's College when Turing was a junior research fellow there and so reached out to Turing directly and was granted access to the Manchester Mark 1—one of the first stored-program digital computers. While the majority of research applications using the Mark 1 were purely mathematical, Strachey developed a number of surprising creative applications that remain the most noteworthy uses of the computer's comparatively limited capabilities. These include some of the earliest computer music, one of the earliest computer games, and arguably the first work of computational art: a love letter–generating algorithm developed alongside Turing.

Strachey is a fascinating figure in the history of computing, not only for his field-defining work within computer science but also for how he exemplifies the complexity of this early moment in computational research, when much of what would become the field of computer science was still unfixed. As an outsider, Strachey did not necessarily share the investments of other researchers working alongside him at the time; for instance, he believed in a clear distinction between the role of computational design and the engineering of computational systems.[9] Indeed many of the applications Strachey developed in the 1950s frustrated normative assumptions about how to balance computational speed and capacity with the elegance of a program's design or the efficiency with which it could be coded. This is especially apparent in the creative applications he developed when awaiting further assignment at Manchester in 1951. Strachey's computer games and music are playful applications that suggest not only that computational machines are vehicles for creative expression but also that such applications might be among their principal uses. The significance of this work is less in their supposed primacy—indeed there are several competing examples for the earliest music and games programmed for a computer—than in their function as the first and principal applications Strachey developed when given access to one of the earliest programmable digital machines. Much as with Turing's thinking on the performativity of human and machine intelligence (published one year prior to Strachey's appointment to Manchester in 1951), Strachey seemed to be testing for the very outliers in what we might consider the hallmarks of our humanity.

The love letter generator is most exemplary in this regard. Taking advantage of the random number generator built into the Mark 1, the program runs through a database of terms to generate formulaic yet evocative purple

9 Martin Campbell-Kelly, "Strachey: The Bloomsbury Years" (presentation, Strachey at 100: An Oxford Computing Pioneer, Oxford University, June 26, 2017), https://podcasts.ox.ac.uk/strachey-bloomsbury-years.

prose. In an article published the same year as Turing's death (1954), Strachey describes the love letter generator's function and gives one of the few surviving examples of the machine's original output:

> Darling Sweetheart,
>
> You are my avid fellow feeling. My affection curiously clings to your passionate wish. My liking yearns for your heart. You are my wistful sympathy: my tender liking.
>
> Yours beautifully
> M. U. C.[10]

Titled "The 'Thinking' Machine," the article explicitly addresses the ways these early experiments served as provocations for Turing's own work on artificial intelligence. Strachey notes that "[o]ne of the most interesting facts brought out by the attempts to make computers imitate human methods of thought is that a great deal of what is usually known as thinking can in fact be reduced to a relatively simple set of rules of the type which can be incorporated into a program."[11] Indeed the queerness of these letters is their disclosure that what seems rich and specific—the sincerity of romantic love—is perhaps entirely generic. This queerness exposes the thinness of normative romantic expression, pointing out the impersonality of affect, attachment, and relation itself. Rather than hold out romantic love as something inherently human and outside of simulation, Strachey's program follows Turing's own provocation in pointing out the largely impersonal nature of what it means to fall in love, suggesting that the Turing Test itself may be viewed as an exercise in the impersonality of humanness, flattening the distinction between man and machine as inhabiting genres of interaction and depersonalization.

From nearly all the writing on the love letter program, it seems clear that neither Strachey nor Turing saw this work as innovative or important. Instead, it seems their disposition toward these experiments was a playful appreciation for the performativity of love and the possibility that a machine might be made to approximate the emotional register of normative affection. Put simply, this exposure of the false veneer lying at the heart of that most deeply human emotion is pure camp: an exultant love of the artificial. In tasking a computer with the camp performance of romantic attachment, Turing and Strachey ultimately lay bare its inability to attain the true expres-

10 Christopher Strachey, "The 'Thinking' Machine," *Encounter* 3, no. 4 (1954): 26. "M. U. C." stands for Manchester University Computer. While this is one of only a few surviving examples of the original love letter generator, several artists and researchers have since reconstructed the program from Strachey's original archival notes. See David Link, "There Must Be an Angel: On the Beginnings of the Arithmetics of Rays," in *Variantology 2: On Deep Time Relations of Arts, Sciences and Technologies*, ed. Siegfried Zielinski and David Link (Cologne: König, 2006), 15–42; and Noah Wardrip-Fruin, "Digital Media Archaeology: Interpreting Computational Processes," in *Media Archaeology: Approaches, Applications, and Implications*, ed. Erkki Huhtamo and Jussi Parikka (Berkeley: University of California Press, 2011).

11 Strachey, "'Thinking' Machine," 26.

sion of romantic feeling. Indeed, the comedic quality of the program is found precisely in this gap between what the program promises to do and its output. Thinking with the history of computing, we might approach this gap as a failure or lack to be repaired through the progressive development of artificial intelligence over the subsequent seventy years. As with the historical claim for nearly all computational systems, we might presume that given enough time and computing power we might one day close this gap such that a machine will convincingly perform the register of romantic love. And yet, read through the affects of Turing and Strachey, the love letter program suggests just the opposite: that the lack implicit in the future-oriented teleology of computation need never be repaired if we learn to love the lack, inhabiting that space in a way that does not feel shattering, dwelling in the gap between love and the letter.

Jacob Gaboury is an associate professor of film and media at the University of California, Berkeley, specializing in the seventy-year history of digital image technologies alongside queer and feminist approaches to the study of science and technology.