

« **Poésie computationnelle** » in Pauline Escande-Gauquié & Bertrand Naivin (éd.), *Comprendre la culture numérique*, Paris, Dunod, 2019, p. 132-143.

Yves Citton

Poésie computationnelle

Que pourrait-on dire de la poésie computationnelle sans parler de la poésie computationnelle ? Cette énigme pourrait être la prémisse de cet article. Le silence assourdissant sur la poésie informatique dans les pages suivantes doit être lu comme un symptôme révélateur. Bien qu'il ait passé plus de trois décennies dans les milieux universitaires littéraires des deux côtés de l'Atlantique, l'auteur n'a jamais rencontré de poésie informatique, et encore moins étudié, au cours de sa carrière. Il savait (ou du moins soupçonnait) que certaines personnes utilisaient des ordinateurs pour produire des textes littéraires. Il aurait certainement été curieux d'en savoir plus sur eux et sur leur travail. Mais aucune autre motivation, plus forte, que cette curiosité fondamentale ne l'a poussé à faire un effort dans cette direction. En conséquence, le type de création littéraire connu sous le nom de « littérature numérique » est resté une *terra incognita*, tant pour lui que pour un grand nombre de ses collègues.

La littérature informatique aveuglante

Bien sûr, cela est moins dû à une conspiration du silence contre la poésie informatique qu'aux frontières disciplinaires étroites établies entre les siècles (l'auteur appartenait à la tribu des « spécialistes du XVIII^e siècle ») autant qu'entre les genres (l'auteur s'est concentré sur les relations entre littérature et philosophie politique). Le fait fondamental (et triste) demeure : on peut passer toute une carrière dans les études littéraires sans être sérieusement exposé à l'une des formes les plus novatrices et les plus passionnantes d'écriture littéraire (la poésie computationnelle).

Cette absence mérite certainement d'être mise en doute, déplorée et, espérons-le, corrigée. Mais, bien que triste, cet angle mort peut également avoir des effets heuristiques. Masquer la constellation la plus lumineuse au cours d'une nuit étoilée aide nos yeux à comprendre les corps environnants moins brillants. De même, l'absence de poésie computationnelle dans le paysage mental de cet auteur peut aider à développer une sensibilité plus fine à d'autres formes de présence computationnelle plus diffuses dans la poésie contemporaine. Cet article tentera donc de cartographier la présence moins évidente de l'informatique dans la littérature – en marge de l'écriture informatique en tant que telle.

Il y parviendra en identifiant quatre énigmes, selon lesquelles les arts de la programmation brouillent les frontières traditionnelles séparant « les deux cultures », les sciences humaines et les sciences. Il considérera les arts de la programmation – conçus littéralement comme des arts qui tentent de définir des comportements automatisés, que ce soit dans des machines ou des humains – comme un terrain de rencontre potentiel entre programmeurs, poètes, conteurs, mais aussi enseignants, concepteurs et législateurs. Chaque casse-tête soulèvera une question qu'une approche créative de la programmation pourrait relever d'un défi artistique et un tremplin politique pour reconfigurer la partition traditionnelle du sensible entre l'homme et la machine, les arts et les sciences, la liberté et la nécessité. L'enjeu est un mouvement de créolisation qui pourrait mettre en parallèle, dans les relations entre humains et non-humains, le type de créolisation le mieux illustré par Édouard Glissant (1997) dans les relations entre les cultures humaines.

Quatre puzzles pour les arts de la programmation

Art technogénétique. On peut d'abord suivre Thierry Bardini (2000) ou N. Katherine Hayles (2012) – parmi d'autres, bien sûr – lorsqu'ils analysent la relation entre pratiques humaines et évolutions technologiques en termes de « coévolution », « Technogénèse ». La leçon générale est

qu'il est de plus en plus difficile de maintenir une distinction étanche entre ce que les humains font avec des machines et ce que les machines font pour les humains, car nous sommes ce que nos appareils (et les institutions dans lesquelles ils sont utilisés) nous conditionnent à être. En même temps, nos appareils sont développés dans les directions qui découlent de la manière dont nous les utilisons.

Ce phénomène très général de la coévolution nous invite à observer comment chaque écrivain singulier choisit d'utiliser, d'ignorer, de rejeter, d'enlever, de subvertir les avantages que lui procure la disponibilité des machines informatiques. Cette observation nourrira le cœur de cet article. Avant de lancer cette exploration, cependant, il peut être intéressant d'inverser brièvement la question : nous pouvons en apprendre beaucoup sur la programmation actuelle telle qu'elle est réalisée – dans une perspective latourienne – si nous essayons de mapper en termes d'écriture poétique la façon dont les programmeurs et les pirates informatiques choisissent d'utiliser, d'ignorer, de rejeter, d'enlever, de corrompre les avantages offerts par leurs machines informatiques. Selon l'intuition de Pierre Lévy (1992), on peut considérer la « programmation comme un art ». La poétique computationnelle n'est pas le créneau privilégié de quelques joyeux poètes utilisant l'ordinateur pour générer des textes et des visualisations : elle imprègne un large éventail de pratiques créatives dans les vastes domaines de la conception de logiciels. Toute la programmation d'histoire (même si elle est encore jeune) peut facilement apparaître comme une aventure fondamentalement poétique, dans la mesure où sa tâche quotidienne a été d'inventer de nouvelles langues au cours des cinquante dernières années. Ce premier casse-tête peut donc être exprimé comme suit : *comment reconnaître le flou de la séparation entre pratiques artistiques et non artistiques dans les activités de programmation, ou inversement, sur la base de quels critères rétablissons-nous cette séparation ?*

Subjectivités computationnelles. Deuxièmement, comme conséquence directe de la coévolution et de la technogenèse, nous pouvons suivre David M. Berry (2013) quand il la définit comme une tâche cruciale pour que les humanités (numériques) puissent mieux comprendre la montée et les enjeux de nos « subjectivités informatiques de plus en plus nombreuses ». Dans la mesure où elles conditionnent notre façon de percevoir, de penser et de communiquer, la machine informatique n'est plus simplement un outil externe : elle a tendance à restructurer nos subjectivités de l'intérieur. Des chercheurs quantitatifs en sciences sociales, des neuroscientifiques, des psychologues, des mathématiciens et des épistémologues tentent déjà de comprendre certaines des nombreuses façons dont nos subjectivités ont tendance à « se transformer en informatique », alors que nous vivons dans des environnements de plus en plus algorithmiques. La notion de « plasticité » a suscité de nombreux débats stimulants entre les locataires d'une vision inquiète, qui dénonce la numérisation (commerciale) comme une tendance à réduire la portée de notre subjectivité (Carr 2011, Stiegler 2004), et les locataires d'une perspective plus optimiste, qui insiste sur le fait que notre agence (collective et individuelle) réagit aux pressions émanant de l'informatique omniprésente (Hayles 2010 & 2012, Malabou 2011). Bien que des recherches quantitatives soient indispensables pour cerner la complexité et l'ambivalence de telles évolutions – souvent sacrifiées à des conclusions idéologiques hâtives –, l'apport des poètes, romanciers, auteurs dramatiques, érudits et philosophes est certainement crucial pour rendre compte de l'expérience subjective de nos subjectivités computationnelles. Dans ce cas, le puzzle est le suivant : *comment intérioriser (à la fois incarner et spiritualiser) les arts de la programmation ? Que peut nous dire la littérature (informatique et traditionnelle) sur le développement de nos subjectivités informatiques ?*

Fiction automatisée. Troisièmement, nous pouvons suivre les penseurs provocateurs qui ont brisé nos hypothèses humanistes confortables en embrassant sans restriction le calcul. Friedrich Kittler (1993) a scandaleusement proclamé que « le dernier acte d'écriture historique pourrait bien avoir été le moment où, au début des années soixante-dix, les ingénieurs d'Intel ont disposé une douzaine de mètres carrés de papier à motif (64 mètres carrés, dans le cas du 8086) afin de concevoir l'architecture matérielle de leur premier microprocesseur intégré ». Jusque-là, il était crédible (et certainement rassurant) de nous voir, humains, utilisant divers types de « machines »

(du fusain aux stylos à bille, en passant par la presse écrite et le traitement de texte) pour écrire nos pensées, sentiments, commandes et projets sur divers types de matériaux, dans le but d'imposer notre ordre au monde chaotique qui nous entoure. Après les années 1970, nous devons faire face au fait que nos appareils d'écriture acquièrent un pouvoir croissant d'écrire à travers, sur et au-delà de nos volontés subjectives. Le GAFAM utilise son algorithme pour poursuivre l'extension de son effort d'écriture et de programmation de toutes ses actions et pensées. Ces approches conspirationnistes et transhumanistes ne sont pas les seuls jeux jouables en ville.

Il était certainement raisonnable, précis et réconfortant de disqualifier l'affirmation de Chris Anderson, dans son célèbre article de 2008 sur « The End of Theory », selon lequel nous n'avons plus besoin des humains (philosophes, théoriciens, scientifiques) pour générer des hypothèses, à un âge où l'apprentissage en profondeur non supervisé assemblé dans des réseaux de neurones peut le faire grâce à la vertu mécanique qui consiste à coupler d'énormes bases de données à une puissance de calcul énorme. Il peut toutefois être beaucoup plus intéressant de prendre cette affirmation (hâtive) comme un tremplin vers une redéfinition du type d'activité exercée par un philosophe, un théoricien, un scientifique, un romancier, un poète ou un dramaturge quand ils noircissent un homme. Feuille de papier vierge ou écran : hypothèse du doigt – malgré Newton. Certains parmi nous (sous l'égide des sciences) formulent des hypothèses, les interrogent, les raffinent, les modifient et enfin les remplacent, afin de rendre compte des données réellement renvoyées par l'observation. D'autres parmi nous (sous l'égide des arts) forgent des déclarations fictives sous-jacentes aux mondes possibles (hypothèses) afin de rendre compte de la manière troublante dont nous faisons l'expérience de notre être dans le monde.

Tant les scientifiques que les artistes essaient de nous faire percevoir et penser au-delà de la portée étroite de notre sens commun : tous deux « forgent des déclarations subversives » (hypothèses à la base) remettant en cause les limites préexistantes de ce qui devrait être considéré comme « pertinent ». En jouant avec les produits de leur imagination, ils contribuent tous deux à l'émergence de nouvelles dimensions pertinentes. En tant que tels, ils devraient tous deux se féliciter du fait qu'une puissance de calcul intensive, reliée à des bases de données volumineuses, peut générer (automatiquement) des hypothèses. Même si l'objectif à court terme, dans le cadre de la structuration capitaliste de l'économie mondialisée, est de remplacer ou de déqualifier le travail humain (coûteux) par la circulation (meilleur marché) de l'électricité dans les microprocesseurs, la perspective plus large dégagée par des algorithmes d'autoapprentissage est : pour nous aider les humains à surmonter l'inertie induite de nos définitions préexistantes de ce qui est (ou non) pertinent. L'exploration aveugle, automatique, systématique et exhaustive des corrélations possibles effectuées par des machines informatiques, si elle est utilisée intelligemment, est très susceptible de contribuer à l'émergence de formes de pertinence inattendues.

Exemple : sur la base de préjugés communément acceptés, nous avons collectivement tendance à conférer le pouvoir explicatif de la causalité à une corrélation entre les tournages de masse aveugles et la pratique intensive des jeux vidéo à la première personne. Des algorithmes d'autoapprentissage sans préjugés explorant les données volumineuses de nos enregistrements de carte de crédit pourraient nous surprendre en montrant une corrélation statistique plus forte entre les tirs en masse aveugles et, par exemple, l'utilisation quotidienne d'une certaine marque de soie dentaire (qui, en examinant de plus près comprend un certain produit chimique doté de propriétés neurotransmetteurs anormales). Une telle hypothèse hétérodoxe serait-elle du ressort de la littérature ou de la science ? Pourquoi avons-nous besoin de fiction ou d'explication scientifique, sinon pour dépasser les limites de notre sens commun et surprendre nos attentes (trop) raisonnables ? Le véritable casse-tête auquel sont confrontés les programmeurs, les scientifiques et les artistes est donc : *comment désamorcer la systématique aveugle des arts de la programmation afin de contourner les aveugles idéologiques de nos attentes imaginaires ?*

Calcul non calculable. Quatrièmement, nous pourrions être inspirés par l'écoute de ces rares voix qui, généralement dans le domaine des études de logiciels (Fuller 2008, Manovich 2014), nous invitent à remettre en question nos hypothèses communes sur les limites ontologiques des machines

informatiques. Nous avons tendance à considérer leur comportement comme totalement prévisible, car ils ont été programmés mécaniquement. Qu'est-ce qu'un ordinateur, sinon une machine dont le comportement a été « écrit à l'avance » (programmé) par son créateur (humain) ? Wendy Hui Kyong Chun (2011) explore l'histoire, la phénoménologie et la politique (sexospécifique) des nombreuses et diverses couches de gestes et attitudes de programmation qui composent notre logiciel actuel. Poursuivant les intuitions de Vilém Flusser (1985) et de Friedrich Kittler (1986), elle met en lumière un lien paradoxal et inquiétant entre la convivialité croissante de nos dispositifs de programmation et l'opacité croissante de leur fonctionnement interne. Les réalisations récentes de la programmation en apprentissage en profondeur dans les domaines de la traduction automatique, de la reconnaissance des formes et de la prédiction du comportement vont dans le même sens : dans notre nouveau monde plein d'algorithmes d'auto-apprentissage, les plus puissants ont tendance à être les moins compréhensibles. Nos dernières réalisations consistent à fabriquer des machines non plus intelligentes, mais autrement intelligentes – et nous ne parvenons souvent pas à comprendre la nature spécifique de cette différence, même si nous sommes tenus de reconnaître son efficacité.

Déplacer la question, cependant, et il peut devenir intrigant familier. Les écrivains littéraires ont souvent vécu l'expérience de la puissance et de l'intelligence de leur fiction. Aussi intelligents que soient Dante, Shakespeare, Cervantes, Jonathan Sterne, Jane Austen, Marcel Proust, Lewis Carroll ou Édouard Glissant, leurs œuvres ont porté des significations bien plus grandes que leurs intentions, leurs projets et leurs projets. Leurs écrits astucieux ont déclenché des cascades d'autres écrits, qui étaient assez imprévisibles au moment où ils ont publié leur travail. Bien que cela puisse également être vrai de scientifiques comme Galileo, Newton, Maxwell, Einstein et Bohr, cela est encore plus directement vrai des auteurs de code actuels. Leurs programmes (c'est-à-dire ce qu'ils ont écrit dans le code afin « d'écrire à l'avance » ce qui sera écrit avec leurs appareils) vont certainement être dépassés par les utilisations futures. Cette caractéristique, cependant, n'est pas simplement la conséquence du fait commun que les humains continuent à se réapproprier et réinvestir les anciennes langues avec de nouvelles significations. Cette excédence pourrait également provenir d'une propriété intrinsèque du calcul.

Luciana Parisi a suggéré qu'une lecture rigoureuse du « nombre Omega » du théoricien de l'information du théoricien de Gregory Chaitin bouleverse radicalement notre assimilation communément admise du calcul avec un résultat prédéterminé :

Pour Chaitin, quelque chose se passe dans le traitement informatique des données, quelque chose qui remet en cause l'équivalence entre entrée et sortie, et donc même l'idée que le traitement conduit toujours à un résultat déjà préprogrammé. Ce quelque chose est, selon Chaitin, un hasard algorithmique. La notion de hasard algorithmique implique que les informations ne peuvent pas être compressées dans un programme plus petit, dans la mesure où une transformation entropique de données se produit en entrée et en sortie, ce qui entraîne une tendance à une augmentation de la taille de ces données. De ce point de vue, la sortie du traitement ne correspond pas aux instructions saisies et son volume tend en fait à devenir plus important qu'au début du calcul. La découverte du caractère aléatoire de l'algorithme dans le traitement informatique a été expliquée par Chaitin en termes d'incomputable : des quantités de données croissantes mais inconnues qui caractérisent le traitement à base de règles. Chaitin appelle ce hasard algorithmique Omega. (Parisi, 2016, 133)

Le calcul génère un incomputable qui lui est propre. L'art de la programmation n'est pas seulement un art de prédéterminer mécaniquement ce qui va être écrit : son traitement mécanique écrit quelque chose à sa manière, qui échappe à la maîtrise de l'écrivain. Le casse-tête et le défi soulevés en parallèle par l'opacité du logiciel multicouche (Chun) et de l'incomparable né du calcul (Parisi) peuvent être exprimés à travers la question suivante : *comment repenser les arts de la programmation au-delà de la camisole de force de sortie entièrement déterministe ?*

Poésie traductionnelle

Certaines formulations de cet article auront peut-être paru *bizarres* au lecteur attentif. Qu'est-ce donc que les « tournages de masses aveugles » ? Pourquoi donc conjuguer GAFAM au singulier (« le GAFAM ») ? Et que diable pouvait donc signifier la phrase évoquant « un poète ou un dramaturge quand ils noircissent un homme. Feuille de papier vierge ou écran : hypothèse du doigt – malgré Newton » ? L'auteur, indiscipliné, avait-il bu ou fumé des substances trop égayantes ? Le correcteur, sous-payé, s'était-il endormi sur son clavier ? Ni l'un, ni l'autre. Cet article a simplement choisi de *mettre à l'œuvre* cette poésie computationnelle dont l'auteur ne sait rien, mais dont lui, l'article, nous dit pourtant quelque chose.

Les réflexions (humaines) qui précèdent ont en effet été originellement rédigées en anglais. Pour s'inscrire dans cet ouvrage sur *La vie numérique*, le texte anglais a été passé dans Google Traduction au cours des premiers mois de 2019. Ce que vous avez lu jusqu'à cette ultime section était le résultat brut de cette traduction automatique. L'auteur se rappelle clairement ses années de jeunesse, assistant de linguistique générale à l'université de Genève vers 1985, où lui et ses collègues se gaussaient des prédictions faites alors de pouvoir bientôt déléguer le travail de traduction à des ordinateurs. Jamais la computation basée sur la grammaire générative ne parviendrait à saisir les subtilités (contextuelles) du sens humain communiquées entre humains par des intelligences humaines. Ces jeunes gens avaient raison : ce n'est pas la grammaire chomskyenne qui pouvait faire un travail satisfaisant de traduction. Ils avaient tort : sur des bases distributionnelles appliquant une énorme puissance de computation à d'énormes bases de données, la force brute des réseaux de neurones parvient aujourd'hui à nous donner une bonne idée générale de ce que dit un texte mécaniquement importé d'une langue étrangère.

Restent bien entendu quelques « imperfections » : la « corrélation entre les tournages de masse aveugles et la pratique intensive des jeux vidéo à la première personne » a confondu le *shooting* d'une mitraillette avec le *shooting* d'un film dans « *a correlation between indiscriminate mass shootings and the intensive practice of first-person-shooter video games* ». La phrase incompréhensible butant sur Newton évoquait en fait « *a poet or a playwright when they blacken a blank sheet of paper or a screen: hypotheses fingunt—Newton notwithstanding* ». Le latin et la référence allusive à l'histoire des sciences auront effrayé la machine, qui aura inséré « un homme » et un point au mauvais endroit, allez savoir pourquoi.

Errare humanum est, dit-on (encore en latin). Mais, semble-t-il, la machine se trompe aussi. Serait-elle donc aussi humaine ? Elle sait en tous cas très bien renvoyer l'auteur à son humanité errante, lorsqu'elle parle des « locataires d'une vision inquiète », là où le texte anglais parlait de « *tenants of a worried view* ». Mais en y regardant de plus près, l'auteur, de langue maternelle française, se demande si ce n'est pas sa connaissance imparfaite de l'anglais qui lui a fait croire (à tort) que *tenant* pouvait désigner le « partisan » d'une certaine idéologie en plus du locataire d'un appartement...

Une certaine forme de « poésie » – à savoir, une certaine forme de dérangement des habitudes langagières, de bougé stylistique, cf Jenny 1991) s'est bien opérée dans les pages qui précèdent, du fait des erratismes croisés de la computation machinique effectuée par Google Translate et de la computation nerveuse effectuée par mon cerveau toujours-en-train-d'apprendre-l'anglais. Sans doute les points de frictions qui auront fait buter la lecture en disent-ils plus que mes prétentieuses réflexions sur la vie numérique et sur la poésie computationnelle.

Références

- Anderson, Chris. 2007. "The End of Theory: The Data Deluge Makes the Scientific Method Obsolete". *Wired*, June 23, 2008. <https://www.wired.com/2008/06/pb-theory/>
- Berry, David M. 2013. "Understanding the Digital Humanities" in *Understanding the Digital Humanities*, London, Palgrave MacMillan.
- Carr, Nicholas. 2010. *The Shallows: What the Internet Is Doing to Our Brain*. London, Norton.
- Chun, Wendy Hui Kyong. 2011. *Programmed Visions. Software and Memory*. Cambridge, MA, MIT Press.

- Citton, Yves. 2017. *The Ecology of Attention*. Cambridge, Polity Press.
- Flusser, Vilém. 1985. *Into the Universe of Technical Images*. Minneapolis. University of Minnesota Press. 2011.
- Flusser, Vilém. 1987. *Does Writing Have a Future?* Minneapolis, University of Minnesota Press. 2011.
- Fuller, Matthew (ed). 2008. *Software Studies. A Lexicon*. Cambridge, MIT Press.
- Glissant, Edouard. 1997. *Traité du Tout-Monde*. Paris. Gallimard.
- Hayles, N. Katherine. 2012. *Lire et penser en milieu numériques*. Grenoble, UGA Editions, 2016.
- Jenny, Laurent. 1990. *La parole singulière*. Paris, Belin.
- Kittler, Friedrich. 1986. *Gramophone, Film, Typewriter*. Stanford University Press. 1999.
- Kittler, Friedrich. 1993. *Mode protégé*. Dijon, Presses du réel, 2015.. *C-Theory*.
- Lévy, Pierre. 1992. *De la programmation considérée comme un des beaux-arts*. Paris. La Découverte.
- Malabou, Catherine. 2004. *Que faire de notre cerveau ?* Paris, Bayard.
- Manovich, Lev. 2013. *Software Takes Command*. New York. Bloomsbury.
- Nova, Nicolas & Vacheron, Joël. 2015. *Dadabot. An Introduction to Machinic Creolization*. Morges. IdPure Editions.
- O’Neil, Cathy. 2016. *Weapons of Math Destruction*. New York, Crown.
- Parisi, Luciana. 2013. *Contagious Architecture. Computation, Aesthetics, and Space*. Cambridge MA, MIT Press.
- Stiegler, Bernard. 2015. *La Société automatique*. Paris, Fayard.