# The Opacity of Critical Thinking's Software. On the Problem of a Critique *Through* Technology

GIACOMO PEZZANO
(Università di Torino)

**Abstract:** The paper addresses the critique of software's opacity by examining the opacity of the "software" used by those very same critics: writing. First, I raise the general problem of bringing software back into visibility by citing cases in which a certain degree of opacity in technology seems even to be required as a precondition for its use (§ 1.1), and by highlighting the specificity of the algorithmic black box, which seems to withdraw radically from its own programmer (§ 1.2). Then, I outline the premises for a renewed critical *ethos*: I take the possible solutions to the algorithmic black box as an illustration of the difference between a critique that *outs* technology (a critique *of* technology) and a critique that *comes out* on its own (a critique *by* technology), ultimately defending a third way, namely, a critique *through* technology (§ 2.1); I discuss what have traditionally been the most common forms of critique, arguing that the intellectual, academic one is based exclusively on the technology of writing and thus questions technology's implicit uniqueness in light of the ubiquity of code (§ 2.2). Finally, I examine the media theorist F.A. Kittler as a key-example of this type of renewed critical thinking: I expose his peculiar "mediological argument", according to which critics should realize the traditional technological conditions of their own activity and accept current changes brought about by computer technology (§ 3.1); I present the code written by Kittler as an example of a critique *through* software, describing him as a true philosopher-programmer (§ 3.2). To

*Trópos. Rivista di ermeneutica e critica filosofica* – vol. 15 (2023), n. 2
ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

conclude, I point towards a mediologically more inclusive future for critical thinking.

**Keywords**: Transparency; Software; Writing; Coding; Media Philosophy.

## 1. *From the Analog Streetcar to the Digital Streetcar: A Growing Opacity?*

### 1.1. *Not All Black Boxes Come to Harm*

That technology is characterized by a certain opacity is not in itself an extraordinarily new discovery: one need only recall the traditional story of the Golem, which represents the prototype of the automatic machine designed by human beings whose behaviour at a certain point or on a certain level becomes inscrutable, unpredictable, and uncontrollable – even for its programmer. But the existence of a veil of opacity in the behaviour of our machines does not necessarily predict an apocalyptic outcome; conversely, it can also have a positive meaning, offering the basis for a fruitful relationship with them. Remaining within the last century, consider Weber's quite famous observation (2004, 17-18):

> Unless we happen to be physicists, those of us who travel by streetcar have not the faintest idea how the streetcar works. Nor have we any need to know it. It is enough for us to know that we can "count on" the behavior of the streetcar. We can base our own behaviour on it. But we have no idea how to build a streetcar so that it will move. […] The growing process of intellectualization and rationalization does *not* imply a growing understanding of the conditions under which we live. It means something quite different. It is the knowledge or the conviction that if *only we wished* to understand them we *could* do so at any time. It means that in principle, then, we are not ruled by mysterious, unpredictable forces, but that, on the contrary, we can in principle *control everything by means of calculation*. That in turn means the disenchantment of the world.

If I am a passenger on a streetcar, I do not need to know its inner workings or what makes it move: I can just take advantage of its presence without any further need to understand what is happening or what I am dealing with. Since there seems to be no reason to pull back the curtain on what is happening during a ride, no critical thinking is required for a passenger. At the same time, if I ever feared that the streetcar might behave like Frankenstein's monster, I would be


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

comforted to know that there is at least someone who could perform this sort of "deep reading", viz., the physicist, who has mastered the laws of nature underlying the construction and the functioning of the vehicle. In other words, there is still someone for whom technology is not totally opaque, and this is enough to keep things under control: disenchantment awaits behind the curtain, if one really wants it, and all mysteries can be somehow dissipated, if it really becomes necessary.

Going even further, Flusser (2000, 27-28) states that human interactions with an apparatus – here specifically a camera – paradoxically function better when presented as a black box:

> The program of the camera has to be rich, otherwise the game would soon be over. The possibilities contained within it have to transcend the ability of the functionary to exhaust them, i.e. the competence of the camera has to be greater than that of its functionaries. No photographer, not even the totality of all photographers, can entirely get to the bottom of what a correctly programmed camera is up to. It is a black box. It is precisely the obscurity of the box which motivates photographers to take photographs. […] The camera does what the photographer wants it to do, even though the photographer does not know what is going on inside the camera. This is precisely what is characteristic of the functioning of apparatuses: the functionary controls the apparatus thanks to the control of its exterior (the input and output) and is controlled by it thanks to the impenetrability of its interior. To put it another way: functionaries control a game over which they have no competence. The world of Kafka, in fact.

If I am a photographer and I am interested in discovering all the affordances of my tool, hopefully in order to produce something particularly creative and original, then the presence of some degree of opacity is even required, lest there be no possibility of properly doing something with it: it is a Kafkian situation, because I am asked to control something which is structurally bigger than I and more complex than my own competence; nevertheless, the circle is in fact more virtuous than vicious, because it frames the condition of real possibility for my decisions and actions. In a similar vein, some media studies scholars stress the relevance of media's «banal deception» (Natale 2021), according to which technologies tend not to be fully transparent (even deceiving us), so that we take their presence for granted and embed them in everyday life, exactly in order to guarantee improved meaning and functionality – that is, to prompt effective actions rather than critical reflections.


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

To take a trivial example: if the steering wheel of my car were to present itself as a big shiny octahedral diamond or as a mere iron tube without any kind of handhold, and not in the form of an unremarkable circle that is easy to handle, it would be a distraction or even a true obstacle – not at all an aid and a facilitator. Thus, it is actually beneficial to be the victim of a "low level" deception which makes one assume, implicitly, that the usual design is the most natural and normal for a car and not one that was tailored to human exigencies.

After all, our ordinary relationship with technologies and devices is not focused on what lies behind the mask of their actual or possible presence, as if we were critical thinkers constantly at work; rather, we use them and count on their readiness-to-hand, to the point that if we find ourselves asking what a tool really is, then something has gone wrong and breaks the usual course of events:

> It is the nature of tool-being to recede from every view. In the strict sense, we can never know just *what* equipment is. Like the giant squids of the Marianas Trench, tool-beings are encountered only once they have washed up dead on the shore, no longer immersed in their withdrawn reality. […] Tool-being is that which *withdraws* (Harman 2002, 4-5; see also Heidegger 1962, 95-107).

To what extent does this phenomenon also apply to our currently predominant technologies, viz., the digital ones? Not so much, according to the many voices decrying the slippery slope of digital technologies. For them, the Golem's vengeance awaits just around the corner. Let me explain.

### 1.2. *Is There a Machine's Soul Reader in the (Dark) House?*

On the most general level, this trepidation exists due to the existence of an intrinsic opacity that is further obscured by an often obsessive insistence on the key-value of transparency, one symbolized by the Apple Store's glass cube design, that is, as a totally transparent structure lacking any dark side or room in which everything remains totally visible. This paves the way for a new form of ideology, according to which not sharing everything publicly and fully openly suggests that one has something to hide and refuses to take care of others: the consequence is a radical and constant self-exposition which perilously tends to overlook the fact that we do not know who is doing what with all our data (see e.g., Carbone, Lingua 2023; Han 2022; Lyon 2018; Zuboff 2019). An iconic example of such a condition is the problematic connection


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

between user-friendliness and interfaces on the screens of our devices: since the appearance of the first Macintosh computers, what Apple calls the "transparent interface", emphasizing its fluid and efficient performance, should more properly be understood as an "opaque" one, to the extent that a desktop screen populated by icons, i.e., representations of file folders, documents, trash, etc., offers nothing to reveal the underlying structure of the device, that is, the nature of what lies hidden under the interaction surface (cf. Turkle 2005, 7-12). The more we accord solid trust to the astonishing performances of the new, autonomous «E-Memory» represented by digital devices or «Cloud-Tech», the more it becomes «more transparent-in-use and at the same time more *opaque in its workings*»: the result is that «we may use it with felicity but increasingly have less idea of how it works» (Clowes 2015, 272).

Nevertheless, one could still follow Weber and say that users have no idea what lies behind their screens and who has access to it, but at least we have the equivalent of the old physicist somewhere: the computer scientist or programmer who knows what "ordinary mortals" cannot even imagine. I, as a simple user, am not supposed to master the peculiar language of the machine, but I can count on the fact that at least its designer is able to dialogue with it in a deeper way: any driverless streetcar always has a programmer who is able to read – and, before, to write – its code. *Unfortunately*, with the algorithmic black box this seems not to be the case (cf. e.g., Burrell 2016; Castelvecchi 2016; Pasquale 2015), particularly with the neural networks at the core of the so-called deep learning AI, capable of learning automatically something for which it had not previously been trained. For instance, we can have a convolutional neural network for self-driving cars which develops the capacity to detect useful road features after being trained with data on the steering angle generally used by humans but not with the outline of roads (Bojarski *et. al.* 2016). To put it simply, a Tesla makes decisions that neither the passenger nor the programmer are able to understand: even though there was at least some "Psychologist of the machine" with the analogue streetcar, this is not the case with the digital streetcar, as we cannot rely on a machine's soul reader or the like – apparently.

## 2. *The Essence of Foreground: Rewriting Our Critical Ethos*

### *2.1. Between Outing and Coming Out*

Not by chance, many trends in critical studies of digital society – such as, just to mention some, surveillance studies, sousveillance studies, critical data


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

studies, and critical algorithm studies – take as a starting point the fact that the main problem with new technologies is that «they exhibit a "deep opacity"» which should be tackled as soon as possible (Frabetti 2014, xiii), and, more specifically, that «as software increasingly structures the contemporary world, curiously, it also withdraws» becoming for us «harder and harder to focus on as it is embedded, hidden, off-shored or merely forgotten about»: hence, the true challenge would be «to bring software back into visibility», paying attention to «what it is (ontology), where it has come from (through media archaeology and genealogy) but also what it is doing (through a form of mechanology)» (Berry 2011, 4). Let's pretend that it is actually possible to bring software back into visibility, solely for the sake of preserving our mental health and not losing any hope for a better future: the crucial question of what this could mean, exactly, still remains open. Generally, the meaning of this transparency is sought through either a human-correction strategy or a machine-correction strategy. On the one hand, there are attempts to excavate AI (e.g., Crawford, Paglen 2021), unmasking the human bias which machines inherit from us – namely, the coded biases based on social, racial, and gender distinctions. For instance, if current AI is predominantly portrayed as white, it is because its programmers live in a predominantly white milieu and frame (e.g., Benjamin 2019; Buolamwini, Gebru 2018; Broussard 2019; Cave, Dihal 2020; Eubanks 2018; O'Neil 2016; Kantayaa 2020). In other words, we begin to acknowledge that automated systems are neither inherently good nor neutral because they reflect the priorities, preferences, and prejudices – the "coded gaze" – of human beings in general and of those who have the direct power to mould artificial intelligence more specifically: thus, we first and foremost need to correct the programming. One proposed solution might call upon a specific group of «algorithmic watchers» or «algorithmic guardian angels», able to monitor the design of AI in order to evaluate the potentially biased behaviour of algorithms during their evolution: human angels totally self-transparent or tendentially such, hence able to guarantee the generation of perfectly transparent algorithms (Jean 2019, 88-153). But if there is no way to really penetrate the deep learning processes of AI, how could this be effective? Hence, on the other hand, we find the idea that it should be AI itself that explains what it does, and ideally also how and why: this project of a so-called «explainable artificial intelligence» entails the introduction of further levels of explication between the "inner" training phase and the "external" communication phase with the user (see e.g., Kamath, Liu 2021; Phillips *et. al.* 2021).


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

Thus, while in the first case it is the human being who tries to open the black box of AI, revealing its hidden essence, in the second case it is the AI itself which is supposed to become more open, manifesting its own true nature: it is the difference between "outing" – the external and even violent disclosure of someone else's secret – and "coming out" – the act of deliberate self-disclosure. In other words, on one side we critique and correct the AI, while on the other side it critiques and corrects itself: in the first situation, we shed light on the software on order to take control of it; in the second situation, we let it run until it arrives on its own at the resolution of a "critical-ethical algorithm". Significantly, this possible bifurcation – which, it goes without saying, covers the opposite poles of a spectrum of possible nuances – seems to correspond with the two of the more traditional critical-ethical moves towards technologies: an ethics and critique *of* technology and an ethics and critique *by* technology – again, simplifying. On the one hand, the quest is for some kind of human intervention able to limit and lead technological evolution and trends, directing them towards the good (human-solutionism); on the other hand, the idea is that some kind of technological fix or shortcut will be able to solve the problem – created by an earlier technology – sooner or later (techno-solutionism). Clearly, "techno-solutionism" is not an option at all for a "human-solutionist", not even when the possibility of being autonomous, independent, and self-explainable seems to characterize the new digital technologies; conversely, the fact that a streetcar ultimately has no ability to explain itself even though we might expect it to do so is precisely the source of any possible problem: ethics and critique are a transcendent affaire, not an immanent one, that is, they cannot be intrinsic to technological performances. Self-transparency cannot be the solution: there is no chance for a coming out.

In such a perspective, bringing software back into visibility would mean precisely that we reject the dream that it will show itself by itself (as a Hegelian Absolute or a Neoplatonic divinity) and begin to act *on* it, first of all by critiquing it from an exterior point of view, namely, by somehow defining its limits and directing its behaviour: in such a case, outing is considered the only effective way. Traditionally, this involves intellectual acts such as gaining or regaining awareness, realizing one's own influences and conditionings, explaining reasons, reconstructing genesis, and/or various forms of political and pedagogical intervention able to rule the production (e.g., European GDPR, Italian "Garante della privacy", etc.) as the use (e.g., training on how to be a conscious and responsible user, education to digital awareness, etc.). I have nothing against these strategies, but we should ask: by which means are


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

they actually performed or could they actually be performed – particularly on the intellectual level? Is it possible both to reject an ethics or critique *by* technology and also to push a little bit further into an ethics or critique *of* technology, when it comes time to clean the software? I suggest that we can, taking general inspiration from the so-called "ethics by design" approach: there is also the opportunity to walk the third path of an ethics or critique *through* technology, in which "ethics" does not merely mean to be value-inspired or good-driven but implies first of all a new *ethos* in the sense of a new way of carrying oneself, of a different attitude, and "critique" does not solely engage in investigating technology in the traditional ways, viz., from the outside, but can also interrogate it in a more experimental way, viz., from the inside.

Exploring such a doing-within represents the premise for re-designing our own critical *habitus*.

### *2.2. Renewing the Toolbox of Intellectual Labour*

Let's start with Zerubavel's discussion (2018, 60-92; 2019) of the logic and mechanics of semiotic subversion. The Israeli sociologist, motivated by the idea that a social scalpel underlies the way we generate our deep mental entities and patterns, individuates two basic strategies for questioning what we take for granted: marking the hitherto unmarked or *foregrounding*, and unmarking the hitherto marked or *backgrounding*. I focus here on the first strategy, which revolves around the attempt to refrain from taking our default assumptions for granted by making the implicit explicit: it is what we generally call critical attitude – just think of the semantics of defamiliarization, estrangement, deautomatization, denaturalization, demystification, unmasking, debunking, and so on in critical theory. According to Zerubavel, such a gesture finds expression in four fundamental ways: foregrounding can be *political* (social activism), *academic* (intellectual labour), *artistic* (creative works), and *comic* (humoristic interplays). Clearly, there can be several overlaps between them, but the distinction seems quite intuitive, capturing the difference between – let's say – the actions of Martin Luther King Jr., Theodor L.W. Adorno, Spike Lee, and Ricky Gervais (but you can pick the examples you prefer, of course). Briefly, we would have four main options for exercising our ability to take a critical position towards our reality, and – more generally – our same fundamental tenets. With respect to digital technologies, these would correspond to the EU AI Act endorsed in 2024, to a refined philosophical study on the ethics of the AI such as Floridi (2023), to the installation *Toy Prototipe* by


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

the South-Korean performance artist Geumhyung Jeong, and to an interaction with ChatGPT that, when asked to produce a short aphorism about how chatbots are ruining human intellect, responds that «chatbots, shortcuts to answers, can diminish depth in thought» (again, one can choose the favourite examples).

Now, I want to stress that a similar articulation seems to allow no room for a peculiar *technological* foregrounding, that is, for a form of critical action which allows for the design of a technology whose aspect and use are different from the ones currently taken for granted – and are hopefully more ethically inspired, by embodying moral values and script (Akrich 1992; Klenk 2017; Latour 1994; Poel, Kroes 2014; Verbeek 2006), be it following the design for all approach, the participative design approach, the inclusive design approach, or the value sensitive design approach (Umbrello, Poel 2021), and be it – when digital technologies are concerned – intervening on software or on hardware (Striano 2023). To put it differently, the tool seems unable to bear and express any ethical and critical stance, as such: it is just a tool and lacks any kind of self-reflexivity – and here is why it needs to be governed and directed towards the good or the like. Maybe this reflects some still ingrained prejudice towards the different roles of philosophers, scientists, technicians, and craftsmen in society, as in general towards the different social and human value of intellectual and manual labour, and/or more plainly – giving that often the best solution is the simplest solution – a factual lack of technical skill. As it is, it is important to notice that such a reluctance towards a sort of technological critique of technology itself – whatever that means, for the moment – seems to concern mainly intellectual criticism, to be fair, that is, humanistic "foregrounders": one hardly finds – still being generous – academic humanistic intellectuals dealing with the ideation and construction of a technological device as an output of their reflexive labour.

Well, not so fast: this is true depending on what we are willing to consider as technology.

To get straight to the point, it would be inaccurate to say that intellectuals do not practice technological foregrounding: in fact, they do it all the time, but using a particular kind of technology, i.e., *writing*. Academic foregrounding is as such technological, insofar as it consists in *written* foregrounding. This may seem a truism (and actually it is) but this is precisely the question I want to raise, now with a specific focus on the software's opacity: if it is in fact true that «although the technology of code is now everywhere, the ability to read and write it is not» (Vee 2017, 15), then we should also accept that «we need to


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

become more adept at reading and writing computer code in order to fully understand the technical experience of the everyday world today» (Berry 2011, 9), namely, in order to fully articulate a critical and ethical stance towards digital technologies and artificial intelligence. Otherwise, we risk leaving coding in the exclusive hand of that «elite of scientists and technicians» whose members «use numbers (algorithms) to articulate and communicate information», labelling those «letter addicts» who still «attach some glimmer of reverence» to the act of writing/reading as «incapable of understanding the communications revolution going on around us» (Flusser 1991, 15). In sum, code literacy can hardly be ignored by those who fly the flag of human literacy, unless we want to exacerbate the divide between the technical approach of data scientists and the literary approach of critical theorists (Moats, Seaver 2019).

The precondition for such a switch in critical labour is posed by whomever – for instance – admits that the true challenge for humanistic studies is to shift its «attention to the *effects* of technology» to «a humanistically informed theory of the *making* of technology», i.e., to «a humanistic computing at the level of design, modeling of information architecture, data types, interface, and protocols»: this would engage in the design of «platforms that embody humanistic values», synthesizing «method and theory into ways of doing as thinking» (Drucker 2012, 87). To go beyond the singular focus on the effects of technology requires making something different *through* technology itself, including at the level of humanistic intellectual labour: a similar awareness nurtures the movement of digital humanities, where we find statements such as «humanists rarely acknowledge the material production of their textual objects» and «humanists do not have a name (other than "art" or "performance") for an interpretation or reading that is not written» (Staley 2017, 36). The result is also an explicit problematization of the traditional humanistic way of conceiving and practicing writing:

> Why must writing, especially writing that captures critical thinking, be composed of words? Why not images? Why not sound? Why not objects? The word text, after all, derives from the Latin *textus*, meaning "that which is woven", strands of different material intertwined together. Let the warp be words and the weft be something else entirely (Sample 2012, 404-405).

Leaving aside the many aspects linked with this reorientation (Pezzano 2024, 203-347), the main question now becomes: why cannot writing, especially writing engaged in critical thinking and thus in foregrounding, be


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

composed (at least also) of code? In other words, why cannot ethical-critical writing be expressed directly *through* software, rather than simply be a critique "about" it and its opacity? Taking this path seems unavoidable today when software takes command of cultural production: it is now impossible to deny that software has replaced a diverse array of physical, mechanical, and electronic technologies once used to create, store, distribute, and access cultural artifacts. Software becomes the universal language through which we speak and our world runs, that is, something which deeply re-adjusts and re-shapes culture, how we make culture – the way we engender and fix meanings (see the already classical Manovich 2013). With that, code has even begun to underwrite the technology of writing itself, layering itself «over and under the infrastructure of writing and literacy», so that – concretely – «much of our textual writing is done on computers, through software programs» (Vee 2017, 35). There is no need to look far: in expressing these thoughts, I am using software; in reading these thoughts, you are using software or have used one – even if you are nostalgic for the *printed* paper, you found the file online, downloaded it, launched the print-app, etc. Paradoxically, the same usual practice of writing a paper in order to discuss the opacity of software and all its related issues only manages to thicken its obscurity: in this exact moment, we are both partners in crime.

So, if we accept that technology in general and digital technology in particular is not only a new object of philosophical and more in general humanistic reflection but also something that can truly change the traditional way of practicing this reflection (Coeckelbergh 2017, 281-287; 2019, 230-268; Hoffmann 2015), then what would it mean to think critically *through* software (if it really is possible)?

## 3. *The Mediological Argument: Kittler's Critical Coding*

### 3.1. *The Critic's Blindness*

An important example of this direction is represented by the German media theorist Friedrich A. Kittler, whose peculiar figure is gaining more and more attention in media philosophy and in media studies as a whole. In this context, he is particularly interesting because he maintains an unwavering level of coherence by following what can be described as "the mediological argument". Its structure is as follows:


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

*Premise 1*. Traditional critique does not pay enough attention to the technological and mediological conditions of critique itself, that is, to the fact that critiquing means writing texts.
*Premise 2*. Today, in the digital era, writing no longer means writing texts.
*Premise 3*. The role performed by written language in the past is now performed by code.
➔
*Conclusion*. Critique requires an update that substitutes or integrates text-writing with code-writing.

With regard to the first premise, Kittler took tremendously seriously the idea that technology represents «the unthought» in Western thinking, which has been repressed or forgotten from its very origin up until now (Stiegler 1998, ix, 276). This is valid not only at the level of its contents, viz., technology has not traditionally been an object of thought, but also and primarily at the level of thinking's form, viz., thinking itself as technology or, at least, the thinking technologies have never been openly thematized and put into question. Indeed, he was fully explicit in highlighting the technological and mediological blindness affecting humanists and specifically philosophers, stressing on several occasions that «philosophical astonishment has never challenged its own preconditions: the techniques of questioning, the books and the institutions, which are philosophy too» (Kittler 1981, 90):

> More than any other theorists, philosophers forgot to ask which media support their very practice. […] The very concept of writing as philosophy's own (technical) medium is missing. […] Metaphysics always already forgets technical media, from writing itself up to the written book, its own precondition. […] Quite in contrast to illuminators, painters, scientists, historians, and poets, thinkers tend to forget their very medium. […] Philosophy, although it dealt from time to time with physical media or elements such as ether, light, and water, completely neglected its own technical media from the ancient volumes up to the modern bestsellers (Kittler 2009, 23-24, 26, 28).

With regard to the second premise, Kittler was quite drastic in diagnosing that «what will soon end in the monopoly of bits and fibre optics began with the monopoly of writing», stressing the transfer of «the age-old monopoly of writing» into «the omnipotence of integrated circuits» (Kittler 1999, 4, 18-19). In his opinion, we could have even stopped writing in the traditional


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

alphabetical sense, since we can count on microprocessors which store, elaborate, and transmit digital bits, thus subverting the technology of writing texts, once the most suitable instrument for recording and managing data flows made of images, sounds, and voices:

> The bulk of written texts – including the paper I am actually reading to you – no longer exist in perceivable time and space, but in a computer memory's transistor cell. And since, in the last three decades, the heroic deeds of Silicon Valley have managed to reduce the dimensions of transistor cells to the submicron level (that is, to less than a micrometer), our present-day scene of writing can only be described by way of fractal geometry […]. Now that all signs have been miniaturized to a molecular scale, the act of writing has vanished. As everyone knows, even if no one wants to say it, nobody writes anymore. Writing – that peculiar kind of software – […] occurs through inscriptions that […] – in contrast to all writing implements of history, are themselves able to read and write. The final act of writing in history may have occurred in the late 1970s, when a team of Intel engineers spread a few dozen square meters of blueprint paper on the floor of an empty garage in Santa Clara and drew up the hardware architecture of the first integrated microprocessor (Kittler 2013, 219-220).

With regard to the third premise, Kittler stated that with the coexistence of the twilight of writing and the dawn of programming, meaning has shrunk down to the sentence, sentences to words, and words to letters – that is, language is now truly made of nothing but strings of symbols which escape the trap of reference/sense, signified/signifier, and the like. In a word, *coding*. According to him, the ancient monopoly of everyday languages, which functioned «as their own metalanguages», has now collapsed, giving way to «a new hierarchy of programming languages», which is moreover fundamentally arranged to «undermine sensory perceptions», so that «we can simply no longer know what our writing is doing, and least of all when we are programming»: thus, «codes begin to proliferate and approach the opacity of everyday languages that, for millennia, has subjected people to these same languages» (Kittler 2013, 221, 218). As Kittler himself has repeatedly insisted, coding is a really weird language, given that it does exactly what it says, namely, it offers both an instruction and an operation, a statement and a procedure, or – more precisely – it represents simultaneously the description of an action and its same performance. If we really want to compare it with our ordinary natural human languages, we could say that code-linguistic acts are not locutionary without always also being illocutionary and perlocutionary – or even better: a code is


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

made of nothing but commands and imperatives, which moreover are immediately carried out. Briefly, in code language, everything is "said" and "written down" in order to be executed straight away. In other words, «programming has a complex relationship with writing»: on the one hand it is writing, while on the other hand it is a digital and technological writing, distinguishing itself from writing in human languages. Programming is writing because it is «symbols inscribed on a surface and designed to be read»; but it is also not writing, or rather, it is something even «more than writing», because «the symbols that constitute computer code are designed not only to be read but also to be executed, or run, by the computer» (Vee 2017, 31).

But what about the conclusion, then? If we do not write anymore, in the sense of our traditional everyday human language, because – like it or not – it is code (and its hardware) all the way down, then what room still remains for critical thinking – if any?

### 3.2. *A Philosopher-Programmer*

The answer, or – at least – Kittler's one, lies in what follows (Figs. 1-3):

```
/*
v. 3.57
31.07.11
PTRACE.PAS (c't 1 / 93,167ff.) Extended
Super ellipsoid, rotational body and procedural textures from povray3
DOS version no longer supported

compile:
Normal: xgraf xsuptrace ray.s matrices.s
// Option -DNEWFRESNEL: simplified Fresnel light from povray3
Option -DJITTER: fuzzy (and time-consuming) shadows
Option -DGAMMA: gamma correction
SVGALIB or DGA: picture.i ".equ XWINDOW, 1" change!

RUNTIME:
<xsuptrace 1>: reproducible noise for runtime tests

FEATURES:
All reflective surfaces can be switched to ReflectionMapping: what
then appears on its surface is a picture to be loaded. This feature
But users have to demand standard interfaces first.

Prompts (':') for scalars and vectors can be defined either with <w [eiter]>
or <n [on]> acknowledge or answer with a new input

3 constant objects: sky, hell, ground (which only allows plein-air images)
Any number of variable objects (limited only by memory)
As variable objects by default 2 balls and 1 other object each predefined
ned. But standard objects can be deleted again.
If more than the standard number is required by an object, inquire
Prompt the new coordinates.
All objects editable (constant only by reassignment of color to normal)
procedure, variable also by location and size).
Some exotic objects are scalable and rotatable. This will be expanded.
Any number of lamps, the first 2 predefined.
Surfaces 1. global, 2. individually assignable after assignment to objects.
LINUX: any size * .24f images can be loaded as 2D textures.

Objects as a ring list for faster intershade () (see Foley, p.
Lamps as a simply linked list.
```


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

```
Left-hand coordinate system: left <x <right, front <y <behind,
below <z <above.

NEW:
01.04.97: AttCol () - acceleration away - transparencies would be worse
07.04.98: individual Fresnel coefficients with individual dullness for
opaque, but metallic surfaces (so these coefficients TransC and
to overwrite the dullness transparently). That's physically correct
and, mirabile dictu, better than POVRAY3. Color charts (ColTabT) global
edierbar
Nov. 22, 1998: ReflV () after Glassner, Image Synthesis, p. 726, again in
Light () calculated per lamp
24.12.98: Object slice selectable; Unzipped DOS - *. 24f files are loadable
08.03.99: Experimental support for Ohta / Maekawa algorithm
01.09.01: DOS no longer supported
13.08.04: Init_Ohta () different, untested
24.10.08: Steel new to /usr/ich/laptop/xsuptrace.c

BUGS:
Spheres of stei, sup, sor determined very empirically
2DMapping on Steiner, Agnesi, SuperQuadrik and Drehkoerper only as a reflection
Map implements: the transformation (x, y, z) -> (u, v) would be hard
MapProc () and Init_Fresnel () are not prepared for multiple calls from xsuptrace
Editing boxes and pyramids still uneven
lambda and thin globally, not variable per surface
No transparency shading as in CALCNEW.C
For floating objects, NULL pointer errors are inevitable; you
Change a midpoint coordinate to small amounts
SOR orb is detected in Edit_Sor () instead of Gravity ()
Refractive indices do not depend on wavelengths of light
08.04.99: intershade () returns L-> Shad correctly, but the object cohesion
breaks again and again, slows down more
30.12.00: xgraf (gcc with optimization) can falsify the SOR curve if
in SorInput the difference quotient Dy / Dx (ie the slope between
two x fixed points) becomes too large.
09.09.03: Changes to transformation matrices only apply to gcc -g. dark
31.07.11: QuColProc () dare

HINTS:
Between internal data structures and user displays, complex conversion
```


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

```
take place; So do not patch global data!
*/

#define  SUPTRACE
#define  COLTABSIZE   127       // unter DOS ggf. kleiner
#define  PIII                  // bei schlechterer CPU dringend aendern!

// CompileTime: Globale Zaehler, bei neuen Objekten oder Oberflaechen erhoehen

#define SurfCnt       27
#define FormCnt       16

#include <time.h>
#define  SPALTEN 640
#define  ZEILEN  480
#define  RAY
#include "xdefines.h"   // SVGALIB: defines.h
#include "ray.h"        // SVGALIB: #include bild16m.c || lock16m.c

extern float Infinit,halb,Epsilon;
VCT3  lambda = { 52.36, 56.6, 68.313 };
float duenne = 0.35, LampDiv = 1.;

// RunTime: Zaehler je Sitzung
int ObjCnt  = 0;
int RingCnt = 0;
int LampCnt = 0;
int FormCnts[FormCnt] = {0};
int TestFormCnt = FormCnt-1;

#ifdef JITTER
float circle[19][2] = {
  {0.75, 0.433}, { 0.0,0.866},   {-0.75,0.433},{-0.75,-0.433 },{0.0,-0.866 },
  {0.75,-0.433}, { 0.0,0.0  },   { 0.75,0.0  },{0.375, 0.65  },{-0.375,0.65},
  {-0.75,0.0  }, {-0.375,-0.65}, {0.375,-0.65},{0.375,0.216  },{0.0,0.433  },
  {-0.375,0.217},{-0.375,-0.216},{0.0,-0.433 },{0.375,-0.217}};
float jitter_offset = 0.75/((float)RAND_MAX);
#endif
```

**Figures 1-3** *An excerpt from the Kittler's code for the program "xsubtrace". Hardware: Pentium IV, x86 family of processors, x87 family of floating point coprocessors; Languages: C and Assembly*


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

These are some of the 5257 lines of computer code written by Kittler himself between early 1980s and 2000s for the graphics program *xsubtrace*, which calculates images according to optical laws following the principle of raytracing, which simulates the propagation of light emitted by one or more light sources[1]. This software was tasked with creating images which do not simply imitate the visible, but directly manipulate the real and use its same laws, transposing physical laws into the algebraically pure logic of computation; this would be a way to instantiate one of Kittler's main ideas: optical media– already since photography and then radically with digitalization – have led to the end of art, that is, the end of the representation of an object for a subject as well as the end of a subject's representation to itself, i.e., the reproduction of something, which is substituted with the self-reproduction of the real itself, i.e., with the direct production of something (cf. Rieber 2017). For my argument, it does not matter if such a claim is actually true and if Kittler wanted to stress this very thing through his program; rather, what counts is the theoretical function assigned to the code by Kittler: he was realizing an epistemological programming, that is, he was truly thinking *through* media, doing critical thinking by coding.

Indeed, as it has been sharply pointed out, Kittler used code as a means to explore technology to inform his own theoretical, traditional writing, with the consequence that a similar exploration in and through code reshaped the way he theorized. For him, to understand a medium meant to understand its most intricate workings – a perfect application of the *verum ipsum factum* principle: hence, programming became effectively «a kind of theorizing, an activity of philosophical labor», which no longer involves the traditional practice of interfacing with a page and tracing ideas by expressing them in words, but «interfacing with a machine and tracing ideas by expressing them in code». In other words, «critical making informs his theory» in the radical sense that it represents for all intents and purposes an act of theorization in itself:

> If, in Kittler's formulation, the workings of technology transform our understanding of the world, by developing a more intimate understanding of the software and hardware by writing and assembling code, Kittler was bathing in the waters of our technological moment while testing, to extend the metaphor, buoyancy and the effects of submersion. […] Just as we read the drafts, journals, and letters of philosophers to understand their writings, so too can we read their code, which is another expression of

[1] For the full code and more technical info, see https://criticalcodestudies.com/kittler.html.


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

> their thought and critical practice, even if it is an assemblage of code derived and adapted from other sources. For what is critical writing but an assemblage built from our own thoughts and the thoughts of others? This is a long way of saying that through this example, we can see how code becomes a channel of discourse, a means of critical thinking and exploration, a writing process of drafting and revision, and a mode of theoretical practice that is theorizing through making (Marino 2020, 166-167).

Hence, Kittler does not simply *say* that media forms shape our minds, that is, he does not simply *write* in defence of the notion of media-technological a priori: rather, he puts it in practice through the exercise of code-writing, which «precedes, expresses, and extends his theory», shaping in this way «his philosophy and his thinking on computational media beyond merely giving him the bona fides to write about it». Thus, Kittler becomes the emblem of «the philosopher-programmer who creates a system for simulating a process to achieve a deeper understanding», making code itself «a means of symbolic thinking and reflection — not merely an illustration of ideas, but an expression of them» (Marino 2020, 170, 176, 197). As a result, this operation helps reduce the potential opacity of the software and of digital technologies in general: it represents a way of achieving effective agency, penetrating the backstage of computation and developing an intimate understanding of its functioning – finally gaining access to a renewed form of understanding. In the end, Kittler incarnates the comprehension of the fact that «if reading and writing are fundamental to subject formation in the age of letters, so too is the reading and writing of code in the age of digital machines» (Marino 2020, 196).

*Conclusion. Towards a Major Mediological Inclusivity*

In this paper, I tackled the question of the critique of software's opacity by proposing to examine the opacity of the "software" used primarily by those very same critical thinkers: writing. *First*, I raised the general problem of bringing software back into visibility by discussing some cases in which a certain degree of opacity in technology seems even to be required as a precondition for its use, and by highlighting the specificity of algorithmic black box, which seems to withdraw radically from its own programmer. *Secondly*, I outlined the premises for a renewed critical *ethos*: I took the possible solutions to the algorithmic black box as an illustration of the difference between a critique that *outs* technology and a critique that *comes out* on its own, defending


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

a third way of critique *through* technology; I distinguished the traditional forms of critique, claiming that the intellectual, academic one is exclusively based on the technology of writing and questioning such an implicit technological uniqueness in the light of the omnipresence of code. *Thirdly*, I presented the figure of the media theorist F.A. Kittler as a key-example of such a renewal of critical thinking: I exposed his peculiar "mediological argument" according to which critics have to realize the traditional mediological conditions of their own activity and face their current change due to the computer technology; I presented a code written by him as an example of critique *through* software, describing him as a true philosopher-programmer.

With that, my aim was not to suggest that a critique enacted through technology in general and coding in particular can make us immune to the opacity of software and algorithms, especially if it is so effectively engrained into their deep structure; nor was my aim to defend the idea that we should substitute the inherited language-written critique with a code-written one, even taking for granted that it would be somehow possible. Rather, my intention was to indicate a possible new alternative for the exercise of critical thinking in our digital age, which points toward major mediological inclusivity in comparison to a tradition so strictly indebted to the written word: certainly, if our critical intellectual labour has really been committed to a sort of monogamous longstanding relationship with writing until now, then we should not abandon the former exclusive partner to hastily engage in another relation of the same kind with coding (or whatever else). Instead, it could be more interesting and fruitful to be open to experiment with a polyamorous engagement: critical thinking can pass through more than one single technology, and even multiple at the same time – *tertium datur*. Undoubtedly, this does not provide a panacea to guarantee the end of digital opacity; but, at the very least, it adds more arrows in the quiver of the critical thinker: this may not only increase our critical awareness, but also promote a new critical attitude based on editing and manipulating rather than on analysing and commenting – a critique which is more in the style of video game modding than in the style of text interpretation.

giacomo.pezzano@unito.it


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

Bibliography

Akrich, M., 1992, "The De-scription of Technical Objects", in W.E. Bijker, J. Law (eds.), *Shaping Technology/Building Society: Studies in Socio-technical Change*, MIT Press, Cambridge (MA), 205-224.

Berry, D.M., 2011, *The Philosophy of Software: Code and Mediation in the Digital Age*, Palgrave, London.

Benjamin, R., 2019, *Race After Technology: Abolitionist Tools for the New Jim Code*, Polity Press, Cambridge.

Bojarski, M. *et. al.,* 2016, "End to End Learning for Self-Driving Cars", in *arXiv*, n. 1604.07316v1, pp. 1-9.

Buolamwini, J., Gebru, T., 2018, *Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification*, Proceedings of Machine Learning Research 81, 1-15.

Broussard, M., 2019, *Artificial Unintelligence: How Computers Misunderstand the World*, MIT Press, Cambridge (MA).

Burrell, J., 2016, *How the Machine 'Thinks': Understanding Opacity in Machine Learning Algorithms*, Big Data & Society 3, 1, 1-12.

Carbone, M., Lingua, G., 2023, *Toward an Anthropology of Screens: Showing and Hiding, Exposing and Protecting*, Palgrave, London.

Castelvecchi, D., 2016, *Can we Open the Black Box of the AI?*, Nature 538, 7623, 20-23.

Cave, S., Dihal, K., 2020, *The Whiteness of AI*, Philosophy & Technology 33, 4, 685-703.

Clowes, R., 2015, *Thinking in the Cloud: The Cognitive Incorporation of Cloud-Based Technology*, Philosophy & Technology 28, 2, 261-296.

Coeckelbergh, M., 2017, *Using Words and Things: Language and Philosophy of Technology*, Routledge, London-New York.

Coeckelbergh, M., 2019, *Introduction to Philosophy of Technology*, Oxford University Press, Oxford.

Crawford, K., Paglen, T., 2021, *Excavating AI: the politics of images in machine learning training sets*, AI & Society: Knowledge, Culture and Communication 36, 4, 1105-1116.

Drucker, J., 2012, *Humanistic Theory and Digital Scholarship*, in M.K. Gold (ed.), *Debates in the Digital Humanities*, University of Minnesota Press, Minneapolis, 85-95.

Eubanks, V., 2018, *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor*, St. Martin's Press, New York.


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

Floridi, L., 2023, *The Ethics of Artificial Intelligence: Principles, Challenges, and Opportunities*, Oxford University Press, Oxford.
Flusser, V., 1991, *On Books*, Artforum 30, 3, 14-15.
Flusser, V., 2000, *Towards a Philosophy of Photography*, Reaktion Books, London.
Frabetti, F., 2014, *Software Theory. A Cultural and Philosophical Study,* Rowman & Littlefield, London.
Han, B.-c., 2022, *Infocracy: Digitization and the Crisis of Democracy*, Polity Press, Cambridge.
Harman, G., 2002, *Tool-Being: Heidegger and the Metaphysics of Objects*, Open Court, Chicago-La Salle.
Heidegger, M., 1962, *Being and Time*, tr. by J. Macquarrie, E. Robinson, Harper & Row, New York.
Hoffmann, M.H.G., 2015, *Changing Philosophy Through Technology: Complexity and Computer-Supported Collaborative Argument Mapping*, in Philosophy & Technology 28, 2, 167-188.
Jean, A., 2019, *De l'autre côté de la machine. Voyage d'une scientifique au pays des algorithms*, Éd. de l'Observatoire, Paris.
Kamath, U., Liu, J., 2021, *Explainable Artificial Intelligence: An Introduction to Interpretable Machine Learning*, Springer, Cham.
Kantayaa, S., 2020, *Coded Bias*, documentary, U.S.A.
Kittler, F.A., 1981, *Forgetting*, in Discourse 3, 88-121.
Kittler, F.A., 1999, *Gramophone, Film, Typewriter*, Stanford University Press, Stanford.
Kittler, F.A., 2009, *Towards an Ontology of the Media*, in Theory, Culture & Society 26, 2-3, 23-31.
Kittler, F.A., 2013, *The Truth of the Technological World: Essays on the Genealogy of Presence*, Stanford University Press, Stanford.
Klenk, M., 2017, *How Do Technological Artefacts Embody Moral Values?*, Philosophy & Technology 34, 3, 525-544.
Latour, B., 1994, *On Technical Mediation*, Common Knowledge 3, 29-64.
Lyon, D., 2018, *The Culture of Surveillance: Watching as a Way of Life*, John Wiley & Sons, Hoboken.
Manovich, L., 2013, *Software Takes Command*, Bloomsbury, London.
Marino, M.C., 2020, *Critical Code Studies*, MIT Press, Cambridge (MA).
Moats, D., Seaver, N., 2019, *'You Social Scientists Love Mind Games': Experimenting in the 'Divide' between Data Science and Critical Algorithm Studies*, Big Data & Society 6, 1, 1-19.


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

Natale, S., 2021, *Deceitful Media: Artificial Intelligence and Social Life after the Turing Test*, Oxford University Press, Oxford.

O'Neil, C., 2016, *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*, Crown, New York.

Pasquale, F., 2015, *The Black Box Society: The Secret Algorithms that Control Money and Information*, Harvard University Press, Cambridge (MA).

Pezzano, G., 2024, *D1git4l-m3nte. Antropologia filosofica e umanità digitale*, FrancoAngeli, Milano.

Phillips, P.J. *et. al.*, 2021, *Four Principles of Explainable Artificial Intelligence*, National Institute of Standards and Technology Interagency or Internal Report 8312, 1-36.

Poel, I.V.D., Kroes, P., 2014, "Can Technology Embody Values?", in P. Kroes, P.P. Verbeek (eds.), *The Moral Status of Technical Artefacts*, Springer, Dordrecht, 89-101.

Rieber, A., 2017, *La fin de l'art selon Friedrich Kittler*, Appareil 19, 1-22.

Sample, M.L., 2012, "What's Wrong with Writing Essays", in M.K. Gold (ed.), *Debates in the Digital Humanities*, University of Minnesota Press, Minneapolis, 404-405.

Staley, D., 2017, "On the "Maker Turn" in the Humanities", in J. Sayers (ed.), *Making Things and Drawing Boundaries: Experiments in the Digital Humanities*, University of Minnesota Press, Minneapolis, 32-41.

Stiegler, B., 1998, *Technics and Time: 1. The Fault of Epimetheus*, Stanford University Press, Stanford.

Striano, F., 2023, *Alert! Ideological Interfaces, TikTok, and the Meme Teleology*, Techné: Research in Philosophy and Technology 27, 2, 183-200.

Turkle, S., 2005, *The Second Self: Computers and the Human Spirit*, MIT Press, Cambridge (MA).

Umbrello, S., van de Poel, I., 2021, *Mapping Value Sensitive Design onto AI for Social Good Principles*, AI and Ethics 1, 3, 283-296.

Vee, A., 2017, *Coding Literacy: How Computer Programming Is Changing Writing*, MIT Press, Cambridge (MA).

Verbeek, P.P., 2006, *Materializing Morality: Design Ethics and Technological Mediation*, Science, Technoloy, & Human Values 31, 3, 361-380.

Zerubavel, E., 2018, *Taken for Granted: The Remarkable Power of the Unremarkable*, Princeton University Press, Princeton.

Zerubavel, E., 2019, "Foregrounding and Backgrounding: The Logic and Mechanics of Semiotic Subversion", in W.H. Brekhus, G. Ignatow (eds.),


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440

*The Oxford Handbook of Cognitive Sociology*, Oxford University Press, Oxford, 567-584.

Zuboff, S., 2019, *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*, PublicAffairs, New York.

Weber, M., 2004, *The Vocation Lectures*, tr. by R. Livingstone, Hackett, Indianapolis.

**Giacomo Pezzano** (PhD), fixed-term Junior Assistant Professor (RtdA) at the University of Turin (Dept. of Philosophy and Education Sciences). He is currently PI of two research projects which investigate how media and digital technologies can reconfigure our habits of thought, and whose activities also include the production of a philosophical graphic essay in collaboration with the International School of Comics of Turin, entitled *How to Do Concepts with Images*. His latest book is *D1git4l-m3nte. Antropologia filosofica e umanità digitale* (2024).


ISSN: 2036-542X
DOI: 10.13135/2036-542X/10440