along it
around it
against it

For an algorithmic *counter-design*

Francesco Tacchini
MA Information Experience Design
Royal College of Art

Contents ii
Plates iii
Abstract & Acknowledgment iv
Introduction 5

PART ONE THE NOTION OF THE ‘ALGORITHM’ 9

1.1 Algorithmic multiplicity 10
1.2 Music as applied maths 13

PART TWO THE ALGORITHM AS A MEDIUM 17

2.1 A ~~past~~ future vision 18
2.2 Towards algorithmic thinking 21
2.3 Towards a medium, beyond technology 24

PART THREE THE DESIGNER’S ROLE 27

3.1 Design *along* 28
3.2 Design *around* 32
3.3 Design *against* 36
3.4 For an algorithmic *counter-design* 41

Conclusion 43
Appendix 46
Selected bibliography 48

Euclid	9	*The Elements,* 300 BCE
Unknown	10	*Al-Khwarizmi* stamp, released in Soviet Union, 1983
Escher M.	13	*Symmetry Watercolor 117 Crab*, date unknown
Tinney R.	17	*Smalltalk*, cover of Byte Magazine, 1981
Verostko R.	23	*Black Elk Speaks*, plotter drawing, 2006
Vasulka W.	27	Frame from *Tableau IV,* a didactic video demonstrating the Rutt/Etra Scan Processor, 1975
Vasulka W.	30	Frame from *Tableau IV*, a didactic video demonstrating the Rutt/Etra Scan Processor, 1975
Dalí S.	31	*The Visage of War*, oil painting, 1940
Superflux	34	*NHI cost-benefit Algorithm*, 2013
Harvey A.	37	*CV Dazzle Look N^{o} 1*, NYU ITP Thesis Presentation, 2010
Oliver J.	40	*Transparency Grenade*, an handheld network intervention device, an exploit enactor, 2012-2014

ABSTRACT

The study aims to contribute to an analysis of the central role of computation in visual culture by conducting an enquiry on the notion of the *algorithm* and its emergence as a medium, and to further advance a design methodology to critically approach it. It is proposed that algorithms are today the fundamental unit for the production of visual work, whether consciously used or not. Their ubiquitous appearance and ambivalent use in the visual arts pose new questions: how does computation assists the proliferation and transmission of visual culture? What does it mean to accept the algorithm as both medium and end product, having acknowledged the techno-political implications of its ambivalent role? In other words, how can the designer critically approach the algorithm?

ACKNOWLEDGEMENT

I am very thankful to those who provided support during the research stage of this study. I am especially grateful to Gareth Polmeer, my thesis supervisor, for his help and guidance. To Frieder Nake for taking time to answer my questions.
To my family for the precious feedback on every aspect of this paper. Finally, thanks to Anab Jain, founder and director of *Superflux* for providing high quality images of their work.

INTRODUCTION

What is an algorithm? Upon hearing the term today, one might think of complex operations and cloudy processes submersed into commodity hardware, such as smart phones or personal computers. Put simply, an algorithm is a — highly fragmented — procedure for solving a problem; a set of instructions in fact. While the Oxford English Dictionary[1] describes it as 'a precisely defined set of mathematical or logical operations for the performance of a particular task,' one could metaphorically think of an algorithm as a cooking recipe where ingredients and instructions are listed in a precise order.

A somehow better question could enquire on how the perception and work of an algorithm mutate across different fields. Algorithms and their extension, software, permeate in fact many different aspects of our life, having become present in numerous facets of contemporary society. For the purpose of this paper, however, the study of an algorithmic notion and the enquiry on their pervasive appearance is restricted to the field of the visual arts. From such perspective, a reading of the OED definition might imply that the 'task' designers or artists are faced with is producing an artwork or a design; and the 'operations' for the creative process are 'precisely defined', suggesting that intervention on the part of the visual practitioner is superseded once the compositional process is set into motion.

Having advanced a brief definition of the algorithm, one might ask *how* it originates. While a historical framing of the notion could prove itself useful for grasping the origin of the concept, it is important to distinguish between

1 OED Online, Oxford University Press. www.oed.com/view/Entry/4959?redirectedFrom=algorithm (accessed September 3, 2014).

what is meant by *algorithm* in contexts different from the one where it originated, namely Computer Science.[2] Grounding the enquiry in the discipline that deals with the principles and use of computers — in other words, discussing the evolution of computing media to explain the development of algorithms over time — may induce to fall into the trap of a grand linear narrative. The risk can arise particularly when discussing the role of software in contemporary visual culture, since the cultural discourse surrounding computing media and new media, with particular regards to the latter, has close ties with technology, the progress of which is often benchmarked against that of human civilisation.[3]

The present research is therefore conducted with methods that seek to avoid traditional readings of cultural practices, such as those excessively subjugated to the rhetoric of constant evolution — which might obscure alternative or marginal knowledges. Borrowing a term from the French philosopher Michel Foucault,[4] the study attempts to conduct a micro-*archeology*[5] of the algorithm, rather than tracing an even unfolding of events: it is not the search for a beginning, but rather a methodology for digging into the background reasons behind the emergence and use of the concept, given certain conditions. The emphasis is initially placed on the notion of the algorithm as a procedure for executing a task, analysed under the light of disciplines which diverge from strict Computer Science. Special attention is devoted to the construction of musical experience across different cultures by way of a *canon* — which serves as a metaphor for the algorithm throughout the paper.

2 Defined as the scientific and practical approach to computation and its applications. Computer Science is the systematic study of the feasibility, structure, expression and mechanisation of the algorithms that underlie the acquisition, processing, representation, storage, and communication of information. To expand see: http://www.cs.bu.edu/AboutCS/WhatIsCS.pdf (Accessed September 6, 2014).

3 On this note, see the work of Stephen Zielinski, particularly the notion of *psychopathia medialis*.

4 Foucault M. (1972) *The archaeology of knowledge*, p. 42

5 Foucault writes that a discourse emerges and exists according to a complex set of discursive and institutional relationships which are strongly defined by gaps, breaks and ruptures. These conditions of existence are to be enquired.

The second chapter leaves music behind, but not the conception of a canon, to address the idea of a controlled medium. Algorithms are here intended in the formalised, modern sense and are studied as an all-encompassing means for artistic creation — an idea that permeates through the various levels of computational media. First advanced half a century ago, the fundamental concept of a dynamic artistic medium gained momentum during the 2000s with the invention and introduction of intuitive artistic programming languages. Having explored the relationship between software and medium in the realm of visual culture, the analysis focuses on early computer art, in an attempt to prove that the algorithm, as a controlled canon, has intrinsically been a critical medium for long, however one difficult to access in ways that produce meaningful visual work[6] for the majority of artists and designers. The chapter concludes with an indication to look beyond the apparent dichotomy *technology versus art*, by letting the latter shine through the former.

If, on the one hand, algorithms have become fundamental to the various processes of global capitalism such as market, finance, surveillance but also media production and consumption, on the other hand, they are increasingly central in anti-capitalist movements and counter-alternatives.[7] From a design perspective, considering the internal relations of capitalist society helps to reflect upon a possible strategy to approach the algorithm. In other words, it is essential to ponder upon the political reach of technological advancements, if one is to fully grasp the affect of computation on (visual) culture. As the contemporary designer learns to fulfil this new role, a methodology is advanced: designing *along*, *around* or *against* the algorithm.

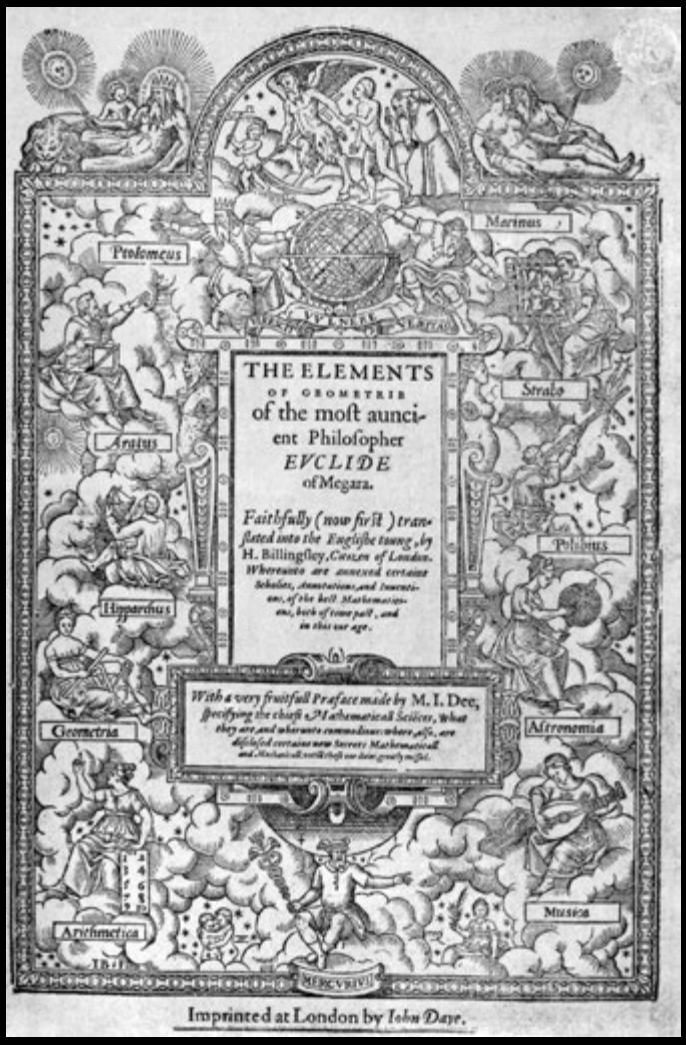
6 Kitchin R., Dodge M. (2011) *Code/space: Software and Everyday Life*, p. 255

7 An example of these practices, specific to the field of economic, is the rise of cryptocurrencies, a medium for exchange that makes use of cryptography to secure the transactions and to control the creation of new units. The first cryptocurrency to be created was *Bitcoin*, in 2009.

The dichotomy between its deliberate and unconscious use in visual practices introduces the final chapter. While most designers un/knowingly design *along* algorithms, others, wary of the techno-political implication of the algorithmic omnipresence, actively work *around* or *against* them in order to expose or exploit them. Relevant examples are advanced to support the hypothesis that critically understanding and accessing the notion of the algorithm — in ways similar, for instance, to those observed in automated musical composition — has lasting consequences on visual culture. The paper concludes with a challenge to the designer: can an emerging *counter-design* practice empower the people?

PART ONE

The notion of the ‘algorithm’



Euclid, *The Elements*, written in 300 BCE. In *Book VII* of the corpus appears – possibly for the first time in history – a non-trivial algorithm: a method for computing the greatest common divisor of two integers. Albeit being known as Euclid’s algorithm, it may even predate Euclid. To expand, see Van der Waerden, (1954) *Science Awakening*.

1.1 ALGORITHMIC MULTIPLICITY

> *Algorism is the art by which at present we use those Indian figures, which number two times five.*[8]

Algorithms are ancient. The term derives from a Latin adaptation of the name of Muhammad ibn Musa al-Khwarizmi,[9] a Persian mathematician, astronomer and scholar living in Ninth-century Mesopotamia. The Latin acceptation of the term, *al-gorism*, evolved from a translation of Khwarizmi's name and referred solely to the rules of performing arithmetic using Hindu-Arabic numerals.

8 De Villedieu A. (1240), *Carmen de Algorismo*

9 Hogendijk J. P. (1998) 'al-Khwarzimi' in *Pythagoras* 38, pp. 4–5

Today algorithms are at the core of computational logic. Most importantly they are pervasive and effectively omnipresent: both embedded in the electronic computing devices that populate our everyday life — in the form of software — or acting at high-frequencies over huge networks such as the financial, communicational, transportation and surveillance ones. In the words of Katherine Hayles[10] code has become 'an important actor in the contemporary world, because it has the power to change the behaviour of digital computers, which in turn permeate nearly every kind of advanced technology.' Software is very present, 'it set off missiles or regulate air traffic; control medical equipment or generate PET scans; model turbulent flows or help design innovative architecture.' For the vast majority of time, however, the multiplicity of forms in which algorithms are experienced is unrecognised; similarly, the extent to which behaviours are watched, analysed, influenced or even governed by the surrounding software is unknown.

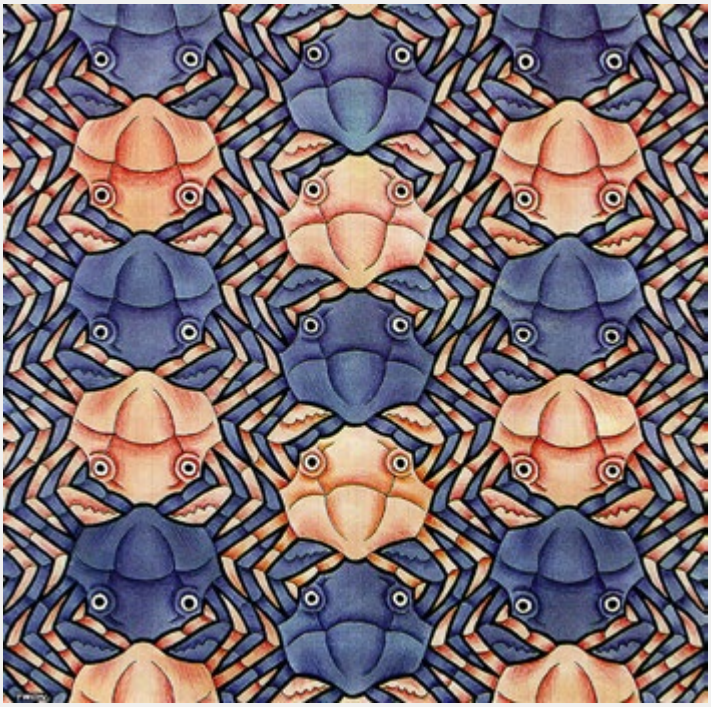
Such lack of engagement is likely a consequence of the complex pragmatic dimension within which algorithms and generally computer programmes run: understandably, a specific language is required to describe and compute these steps, the likes of which the majority of untrained users are not accustomed to. Much effort in traditional algorithmic theory has in fact gone towards the formalisation of syntax, whose main aim is to divorce [formal] expression from [material] content. However, a purely formal conception of the algorithm renders the leap from the practical to the theoretical world a difficult one to accomplish.

10 Hayles K. (2010), *My Mother Was a Computer: Digital Subjects and Literary Texts*, p. 48.

In an effort to bridge this gap, Andrew Goffey suggests a link in-between code and action. In *Software Studies*, he proposes that the algorithm is a sum of *logic* and *control*.[11] In the realm of programming, algorithms perform actions because their logical syntax embodies a command structure that enables to do so. In other words, they are a precisely controlled series of rules and steps in the accomplishment of a task. As aforementioned, an attempt to grasp what an algorithm entails will require to step away from its logically consistent form: if there is more than a purely systematic conception, a notion of the algorithm is to be sought in a broader set of processes across humans cultural practices. Music, for instance, can help demystify it.

Algorithms can be thought of as canons. Under this light, an analysis of their implementation can be extended to musical composition. Throughout human history, arithmetic rulesets have in fact been adopted for constructing and refining musical modes, in a variety of manners. Rhythm and mathematics are deeply intertwined, for example, in the rational tuning system constructed by Pythagoras as the basis of the Western musical scale, or in the exotic variations on the crab-canons of classical Indian music, used by the French composer Olivier Messiaen. Let us briefly dwell upon these observations, to further test the possibility of a transversal algorithmic notion.

11 Kowalski R. (1978) *Algorithm = Logic + Control*, p. 1



1.2 MUSIC AS APPLIED MATHS

> *You cannot evade quantity… You may fly to poetry and music, and quantity and number will face you in your rhythms and your octaves.*[12]

At the basis of Western musical mode lies Pythagoras' scale [A, B, C, D, E, F, G]. The set of familiar notes is the result of a series of precise calculations known as *Pythagorean tuning*: a system which, albeit bearing the Greek mathematician's name, had likely been formalised in Mesopotamia during the Middle Bronze Age and travelled to Hellas through oriental musical instruments.[13] In Sixth century BCE, Pythagoras perfected the scale by altering some of the Mesopotamian tones with the application of a mathematical music theory. He believed — just like Ptolemy, who wrote so after four centuries[14] — that mathematical laws 'underlie the systems of both musical

12 Whitehead A. N. quoted in Newman J. R. (1956), *The World of Mathematics*.

13 Dalley S., Reyes A. T. 'Mesopotamian Contact and Influence in the Greek World.' In: Dalley S. (1998) *The Legacy of Mesopotamia* pp. 85–124.

14 Grout D. J., Palisca C. V. (1996), *A History of Western Music*.

intervals and of the heavenly bodies', and calculated the famous scale using a stack of intervals called perfect fifths, with all its tones based on a 3:2 ratio.[15] The apparent simplicity of the Western scale withholds the complexity of the Pythagorean musical tuning. However, the arithmetic formulas which Pythagoras used to elaborate the system form a plain, precise, procedure. Here lies the notion of an algorithm as a controlled canon; one which is applied [control] in a precise and regular manner [logic] to build a musical scale. Without delving into the details of Pythagoras' theory, it is sufficient to acknowledge that by means of a serial procedure a task is executed and a complex system arises. From a ruleset to a method, to, finally, a long-lasting musical mode.

Building on the metaphor of the algorithm as a canon to construct musical experiences, the compositions of Olivier Messiaen, one of the most important French composer of the Twentieth century, provide for an even more significant example. Messiaen is known for his rhythmically complex music, explored through mathematical formulas.[16] The composer's intellectual curiosity led him to investigate foreign civilisations, incorporating formal aspects of Indian culture into his own symphonies.

During the 1940s Messiaen studied the *Vedas* — the foundations of Hindu spiritualism — to gain a better understanding of traditional classical music. He especially focused on the fundamental part of the Indian melody, the *decitalas*: ascending and descending rhythmic cycles constituted of varying time-units, called *matras*. Among Messiaen works, the rhythmic patterns found in *Cinq rechans*[17] reflect particular mathematical complexity, result of the

15 Sethares W. A. (2005), *Tuning, Timbre, Spectrum, Scale*, p. 163

16 'A musical charm, at once voluptuous and contemplative, that resides in certain mathematical impossibilities of the modal and rhythmic domains.' Messiaen O., (1956). *The technique of my musical language.*

17 Composed in 1948 for twelve a-capella voices, the music was written for a poem created by Messiaen himself in a fictional language similar to French.

application of a precise set of compositional rules. In this oeuvre, Messiaen used and altered the Hindu rhythmic element he had discovered[18] by performing repetitive tasks; his techniques include the addition or subtraction of matras, the splitting of patterns into components and the symmetrical repetition: simple mathematical operations culminating in a canon to be cyclically applied throughout the piece.[19] As in the case of Pythagorean tuning, despite the simplicity of the underlying formulas, *Cinq rechans* appears and sounds complex upon close inspection.

Even though Messiaen was the first to investigate Indian rhythmic patterns and use them consciously, working with augmentation, diminution and symmetries, canonic composition was already in vogue in the Seventeenth century: W. A. Mozart automated score-writing techniques in his *Dice Music*[20] by assembling and combining small musical fragments by chance. J. S. Bach used geometry to construct mirrored and non-retrogradable rhythms, famously used in the *Goldberg Variations*.[21] For these composers, music is intended as a gradual process: once a pattern is constructed and set in motion by means of a rigorous canon, further intervention is little, if not unnecessary. Mozart, Bach, Messiaen and others[22] are programmers writing algorithms. They prefer, as English musician Brian Eno wrote of himself, 'making rather than executing plans' and hence 'gravitate towards situations and systems that, once set into operation, create music with little or no intervention on [their] part'.[23] In 1968, Steve Reich, the father of processual music, theorised that a composer is not an inventor of personal codes for an audience to decipher, but rather a discoverer of impersonal patterns determined through the application of a canon.[24]

18 Wiest L. J. (1990), *Olivier Messiane's Cinqu Rechants: the importance of rhythm as a structural element*, p. 26

19 Ibid, p. 25

20 Alpern, A. (1995), *Techniques for algorithmic composition of music*

21 Hofstadter D. (1979), *Gödel, Escher, Bach: An Eternal Golden Braid*

22 The list comprises: Lejaren Hiller, Robert Baker, Iannis Xenakis, Karlheinz Stockhausen, John Cage… and many more.

23 Alpern, A. (1995), *Techniques for algorithmic composition of music*

24 Steve R. (1968) 'Music as a Gradual Process.' In: Hillier P. (1974), *Writings about Music.*

Reich's thoughts further demystify the notion of algorithm that is being advanced: given a procedure is logically constructed, the execution of a task can reach beyond solving mathematical problems and dynamically generate a complex system, whether musical, computational or visual. In view of this, the canon directing a machine on how to draw a form can be paralleled to that of a theorist building a music tuning system or of a composer writing an oeuvre: both are, in the broad sense, algorithms. In spite of the visual outcome of the former, the similarities are many. The only difference lies in the augmented capacity of a computer to perform certain operations. A quick computing machine provides in fact for a more efficient framework to apply a canon.

In the second half of the Twentieth century, the advent of computers leveraged an enormous calculation power, spreading the application of algorithms far beyond the dreams of its earliest practitioners. During the 1960s and 1970s, in the novel field of computer-driven research, digitally-informed artistic practice was fertile.[25] Programmers begun approaching image-making as a compositional form in which the computer generated the image: the notion of a *controlled canon* for artistic creation was therefore very present. In this context the algorithm gradually escaped its function of computational unit employed for strictly mathematical or scientific purposes and became a capacious medium of outmost importance to the discourse of visual media.

25 Verostko R. (1994) 'Notes on Algorithmic Art.' In: *ISEA '94 Catalogue, The Fifth International Symposium on Electronic Art*

PART TWO

The algorithm as a medium

The August 1981 issue of Byte magazine was entirely focused on a new computer programming language called *Smalltalk*. It introduced *Smalltalk* to many of those who went on to commercialize it. The cover (by Robert Tinney) showed *Smalltalk* as a balloon casting off from a remote island, heading towards the mainland of computer users.

2.1 A ~~PAST~~ FUTURE VISION

How have algorithms contributed to the proliferation of visual culture? To approach such question it is worth looking at the theorisation of Lev Manovich, a key thinker in the field of new media. Manovich has carried out a great deal of research in the recent academic field of software studies. In *Software takes command*[26] — an enquiry on contemporary media and digital culture — he attempts to tackle the important question of what media are after software and whether it is still relevant to discuss them as multiple entities, given the progressively crucial role of code in the visual arts. Manovich delves into the conditions behind the emergence of software as an artistic medium, addressing in particular its affect on the notion and practice of visual media. Among others, a remarkably significant idea is advanced by 'the book's key protagonist: Alan Kay', a notable computer pioneer, who during the 1970s introduced the concept of a computer as a dynamic, governable medium.

Manovich begins by outlining a brief history of computers as the prime media for cultural representation, expression and communication. He interestingly remarks that while in the 1960s the first 'ideas and techniques were developed… [to turn] computers into cultural machines', truly groundbreaking was the work of Alan Kay, who introduced the *Graphical User Interface* [GUI] to the end-user and polished those computing techniques unevenly implemented during the previous decade. In the words of Kay,[27] the digital computer, given its own language, would have become the ultimate 'medium of expression', allowing the user to experiment with 'drawing, painting, animating pictures, and composing and generating music.'

26 The title pays homage to the work of Sigfried Giedion *Mechanization Takes Command: a contribution to anonymous history*, an enquiry on the machines of industrial Western society and the mechanisation of a number of its domains and infrastructures.

27 Manovich L. (2002), *The Language of New Media*, p. 393

28 It is relevant to note that Kay recognised the importance of a computer's *language*, and acknowledged that it superseded the hardware and physical materials used to build the machine. However, in his proposal, these aspects were somehow unified under the paradigm of a *universal media machine*. Not algorithms, nor software – as an extension of them – were the metamedium: the whole computer was. This, in my opinion, might derive from the fact that the state of computing technology in the 1970s and 1980s could not afford software to 'escape' and from the physical black-box within which it was programmed and it operated. It was not possible to have wide access to software because access to the computer itself was very limited. Te little availability of computing devices at the time meant code was one of the machine's important components, but not the fundamental, trivial and omnipresent one it is today, hence the different perception of its role.

Alan Kay had a clear vision. The boldness of the notion of a media machine that would uplift people by providing an active, dynamic kind of human-computer interaction was striking. A computing platform which would not merely coexist alongside the artist's tools, but also be an 'umbrella' for creative means of expression and communication habitual of visual practices. Such a unique cluster of media capable of effectively simulate the details and functions of other and future ones was unheard of. The universal media machine was so qualitatively different from what existed that it needed its own name: *metamedium*.

While this paper does not aim to outline the development of computing platforms, it is relevant to introduce the key notion of *metamedium* to gather a specific, contextualised vocabulary. In 1984 the *Scientific American* published an article in which the computer pioneer further unfolds the suggestion of an all-encompassing machine:

> *It can act like many tools. It is the first metamedium, and as such it has degrees of freedom for representation and expression never before encountered and as yet barely investigated.*[29]

Kay defined this kind of computing paradigm 'active', since it involved the user 'in a two-way conversation' by 'responding to queries', therefore sketching the technical and conceptual basis of the modern-day laptop. The ability to both read and write in a medium is a crucial aspect of Kay's understanding of how to interface with the computer. One is literate only if allowed to perform both actions.[30] Consequently the programming language developed by Kay and his team, namely *SmallTalk*, was consistently used to write the programmes running on the platform. Insuring uniformity

29 Kay A., 'Computer Software.' In: *Scientific American*, September 1984

30 Kay A. (1989), 'User Interface: A Personal View.' In: Laurel B. (1989) *The Art of Human-Computer Interface Design*

would allow one to identify, understand and code their own software: 'all media editing applications… were to serve as examples' for the end-user to draw from.[31]

Three decades later, important advances have been made in the aforementioned direction. Edmund Carpenter, a colleague of Marshall McLuhan, wrote that 'Electricity has made angels of us all', referring to the wonder of our representations being everywhere at once, by virtue of electronic information.[32] The general availability and widespread usage of high-performance, affordable computing devices witnesses the extent to which not the sole electricity of analogue media but that of 'universal media machines' affect global processes of production and consumption of visual culture. In a saturate media context the emergence of software and its use is very manifest: virtually everybody who owns a computer can act freely within the platform and creatively work with applications of varied types. Although apparently stepping in the right direction, some of these developments compromised Kay's project.

The increasingly claimed accessibility and novelty of contemporary media — reinforced by marketing mantras such as 'new upgrade' and 'constant update'[33] or by metaphors such as the 'Cloud' — smoothen out people's perception of computing platforms,[34] while paralleling the paradoxes of closed systems, intellectual property and planned obsolescence[35] which render devices inaccessible on multiple levels. In conclusion, despite the emergence of many kinds of media hub — allowing to edit and consume personal content through proprietary programmes[36] — users are still constrained by pre-determined laws of what can and cannot be done with software; consequentially, such machines do not facilitate the 'not-yet-invented' that Alan Kay longed for.

31 Manovich L. (2014), *Software takes command*

32 Carpenter E. S. (1974) 'Electricity has made angels of us all.' In: *Oh, what a blow that phantom gave me!*, p. ii

33 *Constant Update* is the name of an audio piece by Fatima Al Qadiri and Dalton Caldwell dedicated to the exploration of data-related anxiety. The artists use notification sounds against a repetitive music background to meditate on constantly updated spaces: youtube.com/watch?v=fxxzZrSpc5Q (Accessed August 5, 2014).

34 To paraphrase media theorist Jussi Parikka: 'These are the fantasy object of capitalist economy.'. See Parikka J. (2012), *What is Media Archaeology?*

2.2 TOWARDS ALGORITHMIC THINKING

To fully grasp the essence of polymorph media today, to understand what a medium signifies for the field of visual arts and, finally, to articulate a discourse on the heterogeneous nature of the cultural visual landscape, a 'part is crucial… the part in question is software.' Manovich's claim (*there is only software*) seem to repurpose and resonate with Kay's proposal (*there is one metamedium)*. In spite of the increased use and availability of media-editing software, the recent advances in computing did not necessarily drive a relevant change in the way users utilise it. Nor the idea of a *metamedium* was achieved, as envisioned by Alan Kay. There might only be software; but is it a true medium? Is code facilitating invention, pushing forward innovation in the field of visual arts, empowering people? The answer is yes. Software does permeate visual culture and its current means of production. It can be used as a powerful, transversal, controllable medium. However, it genuinely is one only if algorithmic thinking is applied: a conscious application of code entails that the programmer, artist or designer acknowledge and interact with the algorithms that lie at the processual level of media, understanding what is behind, and beyond, the shiny surface of the GUI.

Manovich states that the concept of a unique medium, comparable to that envisioned by Kay, became more concrete during the 2000s. To create images, the computer artist who rejects the GUI should comprehend part of the computational processes happening and how they can be deployed. In this sense, the recent introduction of intuitive artistic programming languages has crucially contributed:

35 Planned obsolescence was a solution to the Great Depression, hypothesized by Bernard London in 1932. He proposed to label commercial products with an expiry date and tax those using them past the determined lifespan, in an attempt to force consumers to purchase, hence reversing the economic downturn. The idea, unimplemented at the time, gained momentum in the second half of the century, becoming an underlying constant of modern media: separately-sold accessories; dismissed charging cables; non-replaceable batteries, memories, units; discontinued parts… Such technological commodities designed to be unfixable are now the norm.

36 One need only think of the Adobe Creative suite to realise that a lot is possible, while similarly, under the surface, a lot is restricted.

Action Sript, Processing, Python among others have encouraged more users to write software. Zach Sim, one of the founders of *CodeAcademy* — an online school which interactively teaches programming — sustains that since people 'have a genuine desire to understand the world,' they do not merely use a machine, but rather 'want to understand how it works.'[37] The divide between those who are literate and those who are not is slowly narrowing; but it exists nonetheless, even among designers, whose practices regularly depend upon software. In spite of these developments, to familiarise with computer languages is still a daunting task for many: one which, albeit being facilitated, is of outmost importance in the Twenty-first century.

The hypothesis advanced herein is that appreciating the algorithmic notion and accessing[38] code by being able to read it — on a more abstract level, applying algorithmic thinking to art and design — can have a critical impact on the work of visual practitioners. To being constructing a methodology to approach the algorithm as a transversal medium for creative purposes, it is worth looking back to analyse the efforts of a loose group of groundbreaking American artists from the 1970s: the *Algorists*.

Despite essentially being a group of like-minded artists, rather than a defined school, the Algorists share a similar approach to creating Computer artwork. The first and most important aspect of their art is akin to what was observed in musical composition: the Algorists employ a *controlled canon* to describe the actions to be executed (by a machine) to produce work (of art). The use of a computational machine facilitates and enriches this practice. The artist and the computer have

37 Wortham J., 'A Surge in Learning the Language of the Internet' in The New York Times, March 27, 2012. nytimes.com/2012/03/28/technology/for-an-edge-on-the-internet-computer-code-gains-a-following.html?pagewanted=all (Accessed 8 September, 2012).

38 The notion of *accessing code* was advanced by Manovich in *Software Takes Command*, p. 348. Accessing code stands for not only reading but also writing it. Similarly, accessing an algorithm means being literate enough to understand how it operates and intervene on it.

in fact a real symbiotic relationship:[39] beyond the procedure to create art through instructions that define the image-making parameters, a serendipitous, computer-generated aspect is introduced to the process, namely randomness. The Algorist Jean-Pierre Hèbert considers chance fruitful for the artwork, in that it introduces a 'visual order completely independent of either imitation of nature or the limitation and bias of his own invention'.[40] The third typical element of the Algorist's work is artistic sensibility. Roman Verostko warns 'not [to] confuse the procedure by which the artist creates algorithms with

39 Correspondence between Jean-Pierre Hèbert and Nick Lambert, November 2002. computer-arts-society.com/static/cas/computerartsthesis/index.html%3Fpage_id=320.html#_ftn3 (Accessed September 6, 2014)

40 Rickey G. (1995) *Constructivism: Origins and Evolution*, p. 157

the procedures by which the algorithms execute the work'.[41] The Algorist takes decisions about tonalities, character, flow and length, identifying 'the conditions behind acceptable and unacceptable combinations of shape, scale, form and colour'[42] in order to reach an ensemble amenable to become a programme. In this sense, the algorithm is a real *metamedium*, used by the artist who applies it through his conception of coding to create an image, which might differ from that of another artist.

The last aspect is fundamental to algorithmic thinking. Applying a canon to produce design or art results in work which partially detaches from the processes behind it, the canon being controlled *and* serendipitous. Working at the level of code, the algorist effectively defines the structure of and impose the limitations to his own medium, allowing however an element of chance. The algorithm can act like many tools, recalling the words of Alan Kay, and 'it has degrees of freedom for representation and expression never before encountered.' It then is, and has been since the advent of computers, a transversal *canon* for artistic creation.

2.3 TOWARDS A MEDIUM, BEYOND TECHNOLOGY

Before articulating an analysis of the methods to deploy for approaching the algorithmic notion in the visual arts today, a distinction is required between the idea of an artistically mature (meta)medium and that of mere technology. The reception of computational work of art has, at times, suffered from a lack of emotional charge, particularly when the procedure for creating said art involved non-human processes, such as those performed by a machine. Since in the semiotic character of their

41 Verostko R. (1994) 'Notes on Algorithmic Art.' In: *ISEA '94 Catalogue, The Fifth International Symposium on Electronic Art*

42 Verostko R. (1988) *Epigenetic Painting Software As Genotype, A New Dimension of Art*

appearance art and design share, today, many interfaces with software, it is worth discussing the controversy in support of the idea of an algorithmic medium.

The reception of algorithmic art during the 1960s was not overwhelming nor enthusiastic. Most critics did not understand it and there was some small-bourgeois suspicion, as it happens with most new artistic paradigms. However, it was immediately accepted by the art world, not suffering much from the *technology-awing* short-sightedness that is sometimes reserved to recent computational work of art.[43] In *Novelty waves*, Matt Pearson, a coder and self-defined algorist, argues that paralleling software and art to then welcome their intersection as a novel outcome of the fruitful interfacing of two otherwise separate entities has little meaning.[44] In the wittily-titled essay 'Nobody ever cried at a website,' he challenges the idea of a dichotomy between technology and emotion — the contrast arising when the attention awarded to the mechanisms behind a computational work of art supersedes the emotional response to it. Pearson conceptualises that the majority of artistic media have been experienced and labelled as technology in the early stage of their deployment, to only gain an emotional maturity when the audience could see beyond their working apparatus, beyond the awe of implementation. In other words, accepting that 'technology is but a means' helps demystifying the relationship between a medium and its end product — in this case, the relation the algorithm has with digital, dynamic or interactive artworks — letting the latter shine through the medium that defines it. Pearson's analysis is fundamentally correct. Computing platforms, hence algorithms, are not just a technology but rather a *metamedium* which — however

43 Correspondence with Frieder Nake, September 2014. See appendix *(a)* for the full text.

44 Pearson M. (2011) *Novelty Waves*, independently published as an e-book.

problematic to discuss outside a technical frame, given its nature — allows to think and most importantly act freely. Rather than the purely improved computational power or the technological advancements, it is the knowledge gained while developing a canon that grants programmers a higher level of artistic freedom, together with a better conceptual picture of the capabilities of their work. Algorists, as a loose group of like-minded practitioner, realised this. Starting from 1965, their impressive efforts, distilled through a new medium, gave birth to a generative aesthetic[45] which strongly influenced the production of computer art for decades after their appearance.

Having analysed the background reasons behind the transversal application of an algorithmic notion in diverse cultural fields, and having partially established which conditions are behind the emergence of the algorithm as a medium in visual culture, let us examine the position of a designer approaching it in the current socio-political context. What is the contemporary designer's role in a society affected by computation on multiple levels, given the presence of algorithms which reach far beyond their application in a creative context? What kind of algorithmic design is possible? Which tactics and overall strategy can be employed to approach them in technically, visually and politically meaningful ways?

45 Frieder Nake explains that generative is an old principle. The term itself was first pronounced in 1965 in form of 'generative aesthetics'. The occasion was the very first exhibition of computer art, which was opened in Stuttgart, Germany, on the 5th February, 1965. To expand see Frieder Nake's lecture at Eyeo Festival 2014, minute 00:18 vimeo.com/104315361 (Accessed September 12, 2014).

PART THREE

The designer’s role

3.1 DESIGN *ALONG*

The last decades have witnessed a proliferation of designers whose primary medium is the algorithm. They design *along* and with it rather than relying on Twentieth-century commercial media, and are consequently freed from the passive position of a mere responder caged in the pre-built walls of a commercial software's offering and aesthetic. The design or art of those who employ algorithmic thinking reflect their working approach more precisely than proprietary software could ever do: the output is likely to be the elegant result of a carefully built canon.[46] By contrast, those who unconsciously use algorithms produce a less peculiar and distinctive style. As discussed, computer code is at the fundament of the production and transmission of visual culture, but only a conscious application of algorithmic thinking can empower those using it. The unconscious encounter of the two — a designer operating software without questioning its GUI — entails that the commercial application's limitations prevail: the parameters and tools which are in place when a new work is developed witness the strong influence software has already exercised on it.

The work of the Algorists stands as a precedent of such divide. The aesthetic roots of their art-forms differ from those arising from direct manipulation. Nick Lambert writes that 'the algoristic forms are underpinned by visually obvious mathematical processes, manifested in the intricate geometries of their art'.[47] By contrast, the images created with — and mediate by — the GUI result from previous computer inexperience: they are underpinned by a data structure that is not manifest in visual form. An example

46 Lambert N., *Computer Art Thesis*, computer-arts-society.com/static/cas/computer-artsthesis/index.html (Accessed September 10, 2014).

47 Ibid.

of the algorithmic notion happily applied to the visual arts is found in the work of Steina and Woody Vasulka.

The Vasulkas are early adopters of computer-assisted video art, having practiced in the genre beginning from the 1960s. Of particular interest is a machine they helped improving, called *Rutt/Etra Scan Processor*. The analogue electronic tool which manipulates the scanning of the raster in a cathode ray tube, produces effects of image compression and expansion. During 1972 in New York, Steve Rutt teamed up with Bill Etra to build the first prototype. Their 'dream was to create a compositional tool' that would allow to 'prepare visuals like a composer composes music.'[48] Steina and Woody Vasulka used the *Rutt/Etra Scan Processor* to expand the scope of their experiments with images, like other artists would do with similar signal-processing devices in the early seventies. Scan processors, in fact, already existed at the time.[49] However, the Vasulkas did not only play with the machine, but improved it.

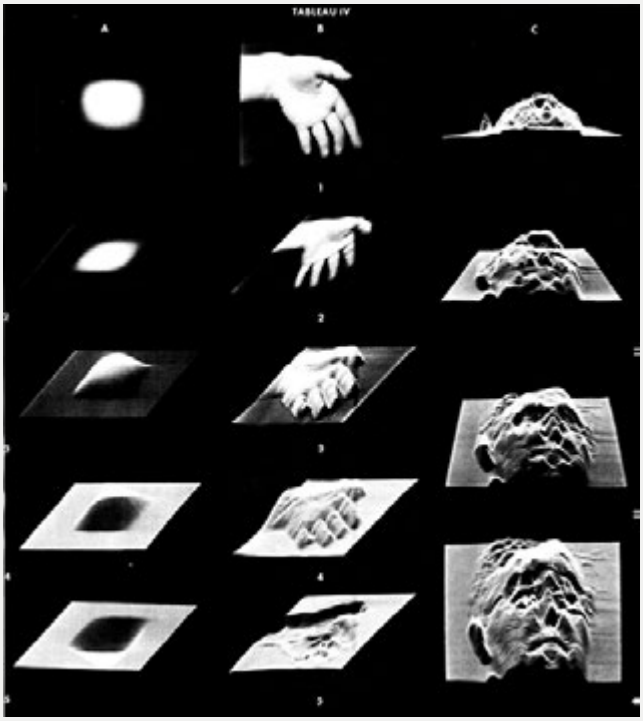
The *Rutt/Etra* fundamentally understood video signals, facilitating the patching of video sync-rate oscillators to control and offset. In 1974, the Vasulkas added an impressive video effect, namely 'z-displace', in which the lighter areas of the frame pull the lines of the raster vertically and give the illusion of three-dimensional shapes.[50] Given the input-signal is an image captured by a camera feed, these lines seem to adopt the contours of objects, creating aleatory landscapes with the quality of being three-dimensional. The resulting imagery brought forward by the Vasulka was technical,[51] abstract and, most importantly, never before seen. Albeit bearing a relative experimental dryness, simple shapes and volumes — such

48 Etra, B. (2008), Blog post: v002.info/plugins/v002-rutt-etra/ (Accessed September 15, 2014)

49 Other relevant examples are Lee Harrison's *Scani-Mate*, Nam June Paik and Shuya Abe's *Paik/Abe Scan Modulator,* Bill Hearn's *Vidium*. To expand, see Jeffrey Siedler's article on video synthesizers: audiovisualizers.com/toolshak/vidsynth/vsynth_intro.htm (Accessed August 15, 2014).

50 Spielmann Y. (2004) *Video and Computer, The Aesthetics of Steina and Woody Vasulka*, Fondation Langlois. See fondation-langlois.org/html/e/page.php?NumPage=456 (Accessed 10 August, 2014)

51 They often made demonstrations, for example, on the process of building up a digital image by layers. See the plate at page 30.



as a sphere, or a hand — functioned as powerful driving forces for creativity.[52]

In their investigation of the electronic nature of video signals, the duo often morphed image and sound, creating unsettling hybrids. One technique employed by the Vasulkas drew from the notion of feedback, formalised in the 1948 by Norbert Wiener in his postulation of *cybernetics*.[53] The use of feedback loop in particular, 'a method in which the artist points his camera directly at the monitor screen, deriving various images through camera manipulation,'[54] exemplifies algorithmic recursion applied to art. Recursion is the process of repeating items in a self-similar way. Its power 'lies in the possibility of defining an infinite set of objects by a finite statement.' Analogously, by

52 The beautiful imagery of the Rutt/Etra synthesizer has been digitally repurposed by the artist Anton Marini in the form of a plug-in for Macintosh OS environment. Marini writes that '*v002 Rutt/Etra* is an attempt to capture some of the beauty of the original hardware,' which is hard to replicate since 'modern graphics and computer systems make fundamentally different assumptions from the analog video systems' in use at the time. The live add-on is available at: http://v002.info/plugins/v002-rutt-etra/ (Accessed August, 24 2014). Furthermore, an in-browser version was developed by Steve Belovarich, and it makes use of live webcam feed and audio analysis to distort the z-displace effect: kineticvideo.co/ (Accessed September 3, 2014).

53 Wiener, N. (1948) *Cybernetics: or Control and Communication in the Animal and the Machine*

54 Shirey, D. (1972) 'Video Art Turns to Abstract Imagery.' In: *The New York Time,* July 4.

To grasp the notion of recursion in the arts it is useful to think of a technique known as *mise en abyme*, which indicates the effect of a picture appearing within itself, in a place where a similar picture would realistically be expected to appear. The appearance, in this case, is called recursive. A famous example of *mise en abyme*, or recursive painting is Salvador Dalí's *The Visage of War* (left).

reversing the process, 'an infinite number of computations can be described by a finite recursive program, even if this program contains no explicit repetitions.'[55]

That of Woody and Steina Vasulka is an alchemical practice, with hands-on circuit bending, software-writing and algorithmic thinking transversally applied to generate new kind of dynamic image systems. Through the use of complex tools they 'explored the relationship of sound to image from a variety of angles; creating pieces in which the audio is controlled by the visual, the visual is controlled by the audio, or both[56] are controlled by an outside source.'[57] The logical construct of their canon resulted in art which was both staged and unexpected, but often *programmed*. They designed, or better produced art, *along* the algorithm. The contemporary designer can, similarly, work alongside it, provided that s/he appreciates the notion of algorithm as a transversal canon and that s/he strives for the unexplored, pushing beyond the immediate surface of the tools s/he already masters.

55 Wirth, N. (1976) *Algorithms + Data Structures = Programs*, p. 126.

56 Steina's *Violin Power* performance (1970 to 1978) further exemplifies the ability to interconnect fields such as sound and image. Yvonne Spielmann writes: 'Besides being the performer, Steina plays the violin and the video so that in intermediary ways the observer and the observed converge. The movements of the bow on the violin's strings deflected the image position of exactly this gesture. Not only does the sound spread the scan lines so that they become horizontally visible thereby exploring temporal dimensionality, but Steina also uses the Scan Processor to modulate the soundwaves until they build up spatial sculptural pattern.' See: fondation-langlois.org/html/e/page.php?NumPage=485 (Accessed August, 18 2014).

57 Carpenter E. S. (1974) 'Electricity has made angels of us all' in Oh, what a blow that phantom gave me!, p. ii

3.2 DESIGN *AROUND*

Algorithms act, but they do so as part of an ill-defined network of actions upon actions, part of a complex of power-knowledge relations, in which unintended consequences, like the side effects of a program's behaviour, can become critically important.[58]

Akin to Kay's computer, the algorithm is a means for 'drawing, painting, animating pictures, and composing and generating music'. Furthermore, the *metamedium* is not just essential to visual culture, but has become an important agent in the global processes of contemporary capitalism. Computing technology has taken on media-quality, it has become ubiquitous, and at the same time fundamental in terms of finance, trade, transport, communication, State and corporate surveillance: a somehow different society has arisen as a consequence of these developments. However, only in the last two decades have the masses of humans been confronted with the complex, widespread phenomenon of computing technology. To frame the algorithmic omnipresence therefore requires that the designer also explore its current societal implications and internal relations.[59]

The ambivalent use of the algorithm in different fields challenges the techno-political literacy of people. Software is not just inside their pockets, submersed in commodity hardware, but also around, above, behind and before them. 'The greater the dependence on a technology the greater the need to study and expose its inner workings', state the *Critical Engineers*.[60] The designer should translate this statement into an approach that transcends the simple application of an algorithm for work of generative aesthetic. Designing *along*, as proposed before, might not be

58 Goffey A., 'Algorithm' in Fuller M. (editor), Software Studies. On the crucial role of side effects in software see Agre P. (1997), Computation and Human Experience, Cambridge University Press

59 Marx states that 'technology reveals the active relation of man to nature, the direct process of the production of his life, and thereby it also lays bare the process of the production of the social relations of his life, and of the mental conceptions that flow from these relations'. See the chapter 'Machinery and Large Scale Industry' in the *Capital* (1867) to expand on the range of relations that technology mediates.

60 From the *Critical Engineering Manifesto*, point #2. See appendix, *(b)* for the full manifesto, or: criticalengineering.org/ (Accessed September 20, 2014).

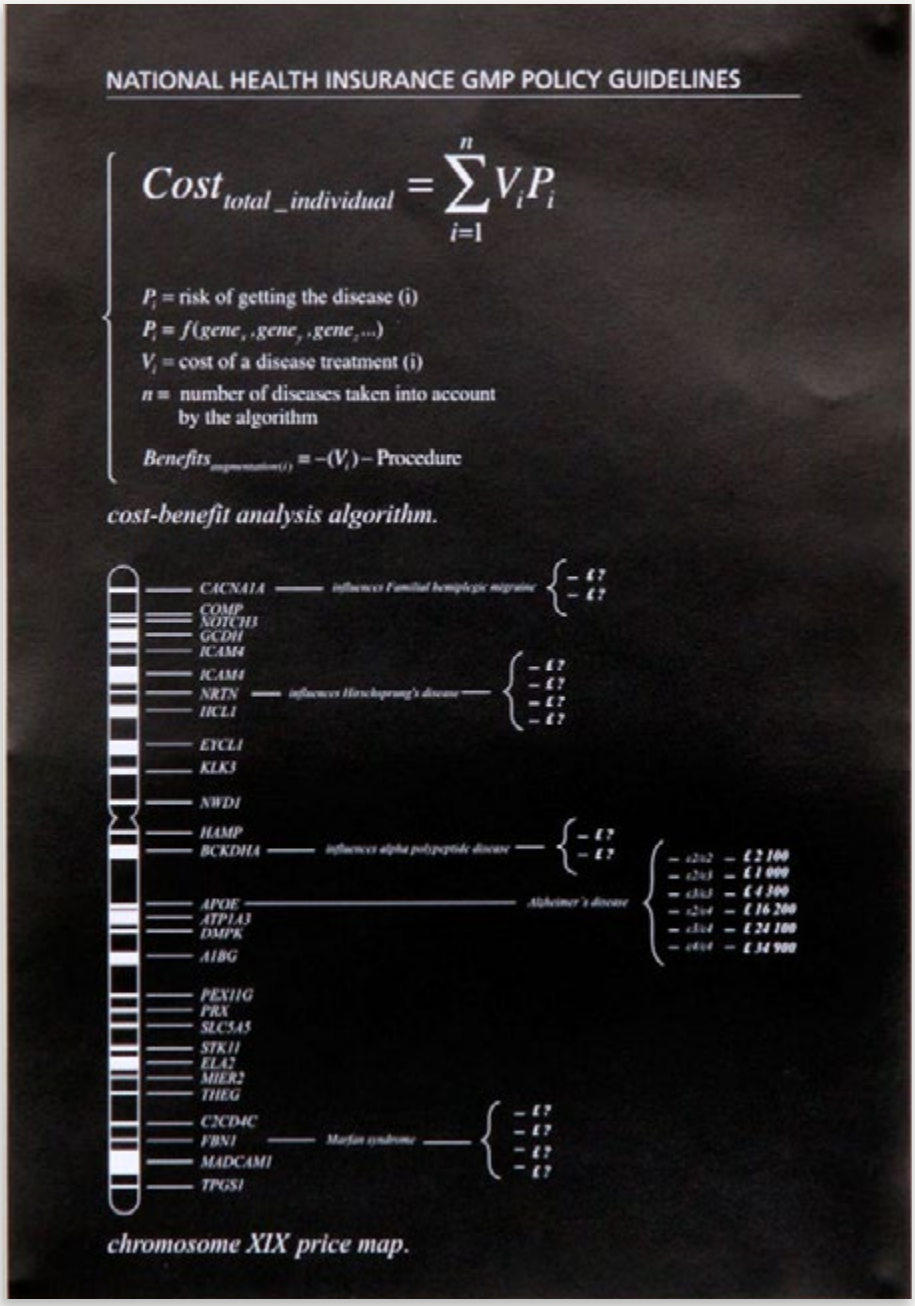
appropriate in this case: designing *around* the algorithm is a sharper method to reflect upon its fluctuating role(s).

To help unpacking the idea one might look at the theorisation of Anthony Dunne and Fiona Raby. In *Speculative Everything*,[61] the duo advocates for a critical approach to design that utilises fictions and speculative narratives to stimulate debate and reveal the choices beyond the constraints of social, technological, economic apparatuses. They juxtapose a design which focuses on rendering technology attractive and consumable to a provocative one which sets to explore the implications of advancement in science and technology, in the service of society rather than industry. Citizen, rather than consumer; problem finding, rather than solving; ethics rather than user-friendliness.[62] The designer necessitates to appreciate the notion of an algorithm to be able to analyse the societal segments and global processes that software penetrates. Only then can he further purpose, or speculate, by proposing alternative narratives which raise awareness, ask questions or spark conversations.

In 2013, *Dynamic Genetics*, a major corporation in the biosynthetic therapy industry, accused Arnold Mann of possessing proprietary DNA. Extensive evidence, obtained by a private security agency, testify that Mann was carrying patent-protected genetic code, obtained by modifying his own DNA in a private clinic. In 2012, Mann's insurance contributions increased, after a health-check from the National Health Insurance (NHI) revealed an elevated risks of chronic health conditions in his genetic profile. The NHI's *DNA Mapping Program* had, in fact, recently developed a cost-benefit algorithm which

61 Dunne A., Raby F. (2013) *Speculative Everything*.

62 *A/B* (2009), Dunne & Raby. See appendix, *(c)* for the full list.


NATIONAL HEALTH INSURANCE GMP POLICY GUIDELINES
$Cost_{total\_individual} = \sum_{i=1}^{n} V_i P_i$
P_i = risk of getting the disease (i)
$P_i = f(gene_1, gene_2, gene_3 \ldots)$
V_i = cost of a disease treatment (i)
n = number of diseases taken into account by the algorithm
$Benefits_{augmentation(i)} \equiv -(V_i)$ – Procedure
cost-benefit analysis algorithm.
CACNA1A
influences Familial hemiplegic migraine
– £?
– £?
COMP
NOTCH3
GCDH
ICAM4
ICAM1
NRTN
influences Hirschsprung's disease
– £?
– £?
– £?
– £?
IICL1
EYCL1
KLK3
NWD1
HAMP
BCKDHA
influences alpha polypeptide disease
– £?
– £?
APOE
Alzheimer's disease
ε2ε2 – £ 2 100
ε2ε3 – £ 1 000
ε3ε3 – £ 4 300
ε2ε4 – £ 16 200
ε3ε4 – £ 24 100
ε4ε4 – £ 34 900
ATP1A3
DMPK
A1BG
PEX11G
PRX
SLC5A5
STK11
ELA2
MIER2
THEG
C2CD4C
FBN1
Marfan syndrome
– £?
– £?
– £?
– £?
MADCAM1
TPGS1
chromosome XIX price map.

would cross-reference a citizen's existing profile — consisting of data-points relevant to fitness activities, daily routines and nutrition habits — with the genetic findings of a spit test. Caught between an inflated health levy and the staggering cost of private treatment, Arnold Mann turned to a black market clinic for an unlicensed gene upgrade, the therapeutical procedures of which are Dynamic Genetics's intellectual property.

The case of *Dynamic Genetics versus Mann* provides for an efficient example of how to design *around* the algorithm. It is a rich, but fictitious and speculative narrative: a thought experiment, albeit one which gives an informed foretaste of how a near-future society might be.[63] The super-fiction, orchestrated by the Anglo-Indian design practice *Superflux*, mirrors real-world organisations to reveal some of the intricate relationships and tensions between politics, economics and computation. To reflect upon wider complexities in order to challenge the deep assumptions about technological power and control is one of the role of a contemporary designer. S/he who is not afraid to delve into the substrata of computational machines and processes, to embrace complexity and to understand the transversality of the algorithmic notion, can advance provocative alternatives to aid deconstruct, demystify, expose and incites suspicion of the murky processes of global capitalism. To design *around* is to bypass the algorithm, circumventing it in order to bring forward a different perspective.

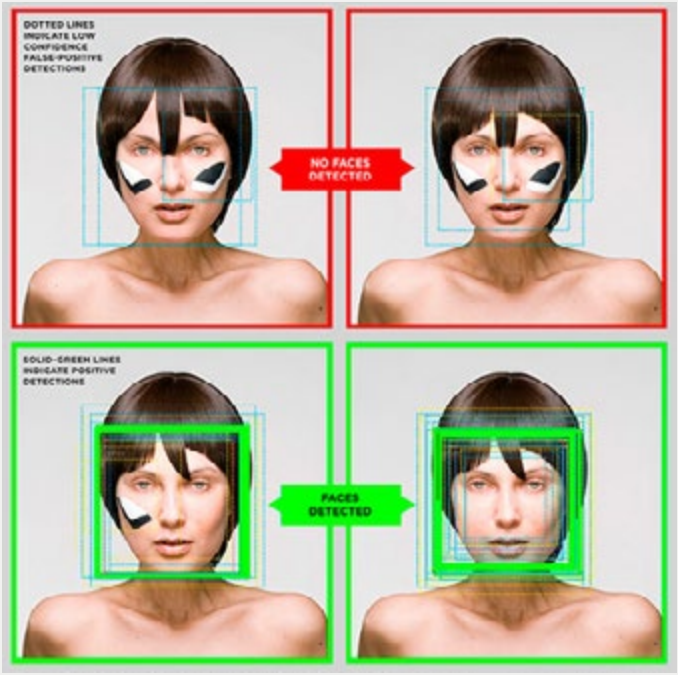
63 To expand see: superflux.in/work/dynamicgenetics (Accessed September 14, 2014).

3.3 DESIGN *AGAINST*

The software people make use of, depend upon and submit to does not simply represent a creative challenge. It can subtly enforce a limit, raise a digital fence, constrain. Those who interact with it often try to disrupt its hidden procedures, playing with the restrictions of a system to inhibit its ability to perform. Resistance to the algorithm has a heterogeneous nature: avoiding taxes by exploiting legal loophole;[64] cheating in a video-game; testing the limits of a responsive kinetic work of art. The algorithm is scrutinised by its human counterpart, in order to assess it, resist it, break it. However, those algorithms acting at a much faster pace than people can hide an unseen threat. A threat to privacy; a threat to freedom of speech; a threat to transparency. In a world of increasingly widespread and sophisticated technology, new radical procedures to culturally confront the computational system shall be brought forward. In view of this, the third method deployed to approach the algorithm should be wary of the political implications behind its emergence. Only then can the contemporary designer exercise a direct approach towards the algorithm to exploit and resist it. That is to say: design *against* it.

Adam Harvey's work represents a typical case of designing *against*. His activity deals with making software observable in meaningful ways by exposing the algorithm and fighting it back by similar means. Harvey's concern revolves around the cheap, unregulated, ubiquitous facial recognition technologies present in public spaces, pairing people lineaments to existing biometric profiles, in a dystopic constant fashion. He considers it a 'mainstay of commercial and government surveillance systems' that jeopardise

64 The artist Paolo Cirio created *Loophole 4 all*, an artwork which 'unveiled over 200000 Caymans Islands companies and combined with an aggressive business strategy, it reversed global finance machination for creative subversive agendas. The website Loophole4All.com promoted the sale of real identities of anonymous companies at low cost to democratize the privileges of offshore businesses. This performance generated international media attention, engaged an active audience and drew outrage from authorities on the Cayman Islands, international law and accounting firms, PayPal and real owners of the companies. Further, the artist interviewed major experts and produced a video documentary investigating offshore centers to expose their social costs and to envision solutions to global economic inequality. In the offline art installation, the paper trail of the project is displayed with prints of the counterfeited Certificates of Incorporation and the documents of the scheme set up for the operation.' See project at: loophole4all.com/


DOTTED LINES
INDICATE LOW
CONFIDENCE
FALSE-POSITIVE
DETECTIONS
NO FACES
DETECTED
SOLID-GREEN LINES
INDICATE POSITIVE
DETECTIONS
FACES
DETECTED

privacy. 'Sophisticated algorithms can already extract information about your gender, age and even mood from a single image, and then link those physical attributes to commercial or government databases.'[65]

A way to escape the threat is to achieve invisibility, the rationale being: if one is unseen, his actions are too. To fight the ubiquitous hazard, Harvey utilises fashion as a masquerade. In 2010, he developed *CV Dazzle*, a reverse engineering method of facial colouring which hinders detection of faces. *CV Dazzle* is a mask. A camouflage. A method of *crypsis*, or avoidance of observation, that allows an otherwise visible organism to remain indiscernible from the surrounding environment[66] through deception.[67] The technique has been used throughout history to confuse and Harvey committed to a new iteration of it, in an effort to provoke a discussion about privacy which sensitises audiences to advanced forms of surveillance.[68]

CV Dazzle is carefully assembled by means of hairstyling and makeup. These are intended to break apart the continuity of a visage that a machine can so efficiently scan. The smart algorithms[70] behind the science of face recognition rely in fact on 'the identification and spatial relationship of key facial features, like symmetry and tonal contours.'[71] However, one can halt machine detection by creating a disrupting *anti-face* that exploits the same techniques used by the algorithms. It is a matter of reverse engineering: by understanding how software works, the counter-process of designing an anti-face is more intuitive. Harvey focused on *OpenCV*, one of the most widely used face detectors, which uses the *Viola-Jones algorithm*. 'It performs best for frontal face imagery and excels at

65 Niedermeyer J., Thomas J. (2014) 'Face to Anti-Face.' In: *The New York Times,* December 14.

66 A. Anand, R. Bhandari, (2009) *Face detection in camouflage faces*, Dept. of Computer Science and Engineering, IIT Kanpur.

67 'From all appearances, deception has always been critical to daily survival – for human and non-human creatures alike – and, judging by its current ubiquity, there is no end in immediate sight.' Behrens R. (2009), *Camoupedia*

68 In a press release Harvey explained his amaze on why an intelligence agency requested to publish his art work in a classified document: 'In 2013 I received an inquiry from an intelligence agency to publish my artwork in a classified document. I obliged but was frustrated by the odd request. Was I a suspect or an inspiration? And what does it mean for an artwork to become classified?' See: untappedcities.com/2013/08/19/daily-what-new-museum-opens-new-counter-surveillance-pop-up-privacy-shop/ (Accessed August 8, 2014).

70 Otherwise known as ubiquitous-computing, these are a tool of Data Science, a discipline dealing with data-mining. See 'A Machine Learning View on Profiling' (2013). In: M. Hildebrandt, K. de Vries, *Privacy, Due Process and the Computational Turn.*

71 To expand, see: ahprojects.com/projects/cv-dazzle/#summary (Accessed September 3, 2014).

computational speed'[72] guarantee its creators: an ideal candidate. The object detection framework developed by Viola and Jones overlaps a series of black and white rectangles across an image to analyse the differences between the dark and light regions of a face. If no face is detected, it continues scanning; but upon spotting a suspicious area, it marks it with a red rectangle. In the post-processing stage these potential faces are checked for overlaps, requiring an average of two positives to confirm a face. Having exposed such software, Harvey teamed up with fashion designers to explore how hair styling and makeup could be used as a camouflage, resulting in an open-source series of 'style tips for reclaiming privacy.' Applying these would render an (anti)face unrecognisable to several state-of-art face detection algorithms,[73] as a series of experiments conducted by the designer with different subjects. A case of algorithmic fashion design versus the algorithm itself.

In conclusion, to design *against* is a straight-forward method for engaging with the algorithm, employed by the designer who consider the exploit to be the most desirable form of exposure.[74] Further to an understanding of software, it implies a frontal, militant involvement aimed at raising awareness about its toxic potential, while advancing concrete alternatives to resist invasive technologies. As Gilles Deleuze once wrote, 'there is no need to fear or hope, but only to look for new weapons.'[75]

72 The algorithm is exposed in a public video created by Harvey: vimeo.com/12774628 (Accessed August 8, 2014).

73 To expand, see: ahprojects.com/projects/cv-dazzle/#summary (Accessed September 3, 2014).

74 From the *Critical Engineering Manifesto*, point #10. See appendix, *(b)* for the full manifesto, or: criticalengineering.org/ (Accessed September 20, 2014).

75 Deleuze G. (1992) *Postscript on the Societies of Control*

Julian Oliver's *Transparency Grenade* (left) is an iconic cure for the lack of Corporate and Governmental transparency. It is a tool which renders 'the process of leaking information from closed meetings as easy as pulling a pin.'

It is equipped with a tiny computer, a microphone and a wireless antenna, 'to capture network traffic and audio, and anonymously stream it to a dedicated server where it is mined for information.'

See more at www.transparencygrenade.com

3.4 FOR AN ALGORITHMIC *COUNTER-DESIGN*

In a rapidly changing world, uncertainty coexists with technological advancements, both articulations of the same transformative language which shapes how people communicate, move and live. To be able to address the ways in which technologies interface with the environment and with everyday life, a *counter-design* strategy has been proposed, which should research and acknowledge, explore and apply, exploit and resist this language and its inherent social, political, economical implications. The contemporary designer can employ three tactics to face the algorithm. S/he may accept the algorithmic notion of a *metamedium*, work *along* it and apply it as a *controlled canon* for artistic invention, resulting in carefully crafted creations of renewed elegance. S/he may go further and confront the beliefs or the challenges posed by technology, finding a way around it so to advance a different perspective on what the work of an algorithm entails: a speculation. Finally, s/he may directly confront it, by not only unmasking the algorithm but also unmaking it, in an effort to oppose it: design *against*. S/he may use these singularly, in pairs or all together.

The study hypothesizes that applying this framework, that is to say implementing the design tactics advanced herein, can have a lasting impact on visual culture. As discussed, the algorithm acting as ubiquitous *minimum unit* across a range of fields of application demands an informed, politically aware, unpredictable, disrupting and visually innovative practice. Designing *along*, *around* and *against* gather under the umbrella of a working methodology that accepts the peculiar role(s) of the algorithm as both

a passive and active agent, a serendipitous and controlled procedure, a *metamedium* and an end-product. Together, these modes establish a transversal kind of critical practice which can be summed up in the expression *algorithmic counter-design*. Can it effectively empower people? Whether such strategy will contribute to a wider and sharper understanding of society's current condition is left for time to tell. Meanwhile, the challenge for the contemporary designer is to *counter-design* with the intent of surprising, uncovering, disclosing, enlightening an audience.

CONCLUSION

In 1977 Alan Kay imagined a *metamedium* that would encompass a wide range of media processes and facilitate the 'not-yet-invented'. The technological advancement of the last three decades put such idea into effect, but in ways different from what Kay had envisioned. The study carried out herein set to frame the idea of a *metamedium* within the broad context of the visual arts, proposing that the minimum and most relevant unit for the contemporary discourse of visual culture is not the computer, nor mere software, but rather the notion of the algorithm.

To assist in the thorough understanding of the idea, a micro-*archeology* of the algorithm was conducted. Such research methodology sought to excavate the background reasons behind a possible transversal use of the concept in fields of knowledge which diverged from strict Computer Science. Metaphorically intended as a *canon*, the algorithm was scrutinised as a procedure for constructing musical experience, with regards to the work and theorisation of Greek mathematicians and European composers.

Building on this transversal notion, the study sustained that an algorithmic canon had already been employed for the creation of art and design half a century ago, with the pioneering work of the Algorists. It was therefore hypothesised that algorithmic thinking, defined as an approach to design and art in which the creative appreciates the notion of an algorithm, has had critical impact on the work of visual practitioners, it continues to do so, and it will increasingly become relevant given the much larger and further-reaching presence of software.

Finally, it was insisted that significant technological, political, social and economical changes respond to the advancements in computational processes and machines, and to the pervasiveness of algorithms. As a result, the *metamedium* escapes its sheer role of creative agent and becomes the vector of transformations in wide segments of the society, hence requiring critical interventions on behalf of the contemporary designer who seeks to fully grasp its essence. A triad of possible approaches were proposed, which constitute a critical methodology for the designer who wishes to approach the algorithmic paradigm in his ambivalent guise: aligning to the algorithm by designing *along* it, circumventing it by designing *around* it or actively resisting it by designing *against* it. These three attitudes establish a practice identified as *algorithmic counter-design*.

To conclude, the paper aimed to give insights on the emerging role of the algorithm in the visual arts and establish a possible designer's practice to critically intervene. Alongside this small contribution, the emergent *counter-design* framework is likely to be explored, questioned, or refused by further research and practitioners, as the world accelerates towards an increasingly algorithmic society.

APPENDIX AND SELECTED BIBLIOGRAPHY

(a) Correspondence with Frieder Nake

Dear Francesco,

my reply to your friendly request is delayed so much because I am currently travelling in the USA and trying to keep communication down to a minimum in favor of my work. Besides, you will notice that I am sending from another account. Don't answer to it but rather to my address as given above.

The dichotomy between art and technology - wrong. Software interfacing with art, but wrongly believed to be without emotional content. These are your hypotheses, right? And these are your questions: **(1)** were early works of computer art received emotionally? **(2)** Were they received at all or were they regretted, perhaps seen suspiciously (computer being mainly used for scientific purposes)? **(3)** Was there the same tendency to look for what's behind and merely awe at the mechanisms of the machine?

I'll give you a few (superficial) comments, before I read your paragraph to which I will try to comment afterwards.

The dichotomy. Of course, it does exist, and it does not exist. It exists as a result of historical developments in modern Western societies where what is being called "art" and what is being called "technology" occupy quite different social domains. There are different schools, different education, different budgets, different institutions, tons of factual differences in the fabric of these societies. They all rest upon the difference, if you like, between the productive basis and the ideological superstructure of these societies (Perhaps read a bit in Tony Eagleton?). But, on the other hand, in the individual, particularly in a practicing this or that, mathematician or filmmaker, engineer or musician, the dichotomy may not exist at all. We can, of course, because of the deep organizational split, identify in most of individuals a clear separation. But each one of us and even institutions may deny it.

Clearly, software and art and design now share many interfaces. I don't have to tell you. It's all because of the semiotic character of software and of the arts in their algorithmic appearance. Then to claim that software, software generation as well as software experience, that software-generated artistic expressions and works were void of emotion is, absolutely, bullshit. You did, of course, find this in the 1960s, and you still find it. Isn't it, first of all, quite okay that a person may react without any emotion to something he or she perceives or does? Why should anything bring up emotions in me. And, secondly, there are many of us who are deeply or flatly moved emotionally when something happens that is of technological origin. This kind of claim is so stupid that you wonder why intelligent people formulate it. If they are not emotionally moved, so what?

Now to your questions. **(1)** Were early works received emotionally? I don't really know. If you ask me, did I receive this or that emotionally, that or this, however, not, my answer is YES, that was so. Was I the only person in the world? I guess not. I would safely assume that all those who did something, also were moved. If only by pride.

(2) Of course, there was suspicion. As always, when a new way of generating something artistic shows up. No, this is no art. What are you doing there? Etc. Petit-bourgeois reactions. The reception of algorithmic art was not overwhelming, it was not enthusiastic, critics didn't understand. But I sold immediately, I was invited immediately, I made friends with artists immediately. I am a living witness who cannot confirm any of the stupid reactions. Yes, the scientific importance of anything in computing, the fear of scientific ways of thinking, algorithmic thinking taking over domains that up to then had been wild and crazy, that happened and caused much justified reaction. However, this reaction has little to do with art! I am afraid of the robot that may some day be my nurse. I am afraid of the people who run NSA or the German secret service. I am afraid of all the computing power that is needed to run our entire society day by day. I am not afraid of computers interfacing art.

(3) There was in some, not many, an awe for technology. This has always been with us. The better we understand an area of technology, the more we get our fingers dirty by touching technology, the less we will stand in awe.

Now to your paragraph. Quite frankly, is this possible?, I do not know Matt Pearson, but I will look up his work. "Nobody ever cried at a website" is a nice statement. Poetic. Is it true? I don't know. How will he know? It's a claim, a projection. It is not without good reason. Chances are perhaps not that high that people cry at websites. So what? Do they cry at laws? Do they cry at mathematics or physics texts? I personally cry at a lot of things, novels, theater plays, poems, music, music, music (with and without text), movies. When I introduce a group of students to programming, I read a poem and discuss it. I then read and run a piece of program and discuss it. They are two forms of text, rather close to each other.

If nobody cries at websites, that's probably due to the specific websites we are looking at. It may well be that what is behind a website, or a piece of algorithmic art, or some functions of Photoshop is more interesting to people than anything else that could also call for their interest. But there are not that many that would like to study the software behind. Everything that exists on a computer exists doubly, in a duplicate mode: as surface and subface, inseparably. The surface is for us to perceive, the subface for the processor to compute. Everything in postmodern times is of this kind. We must live up to this new mode of reality, the artificial and virtual. Losing emotion? What a crazy idea!

Yes, the reaction, our human reaction, to what so bewilderingly and alienatingly appears in technological (computable) form is what we try to find and have been trying to find ever since computers started to appear. This is about 70 years now. Only since about 1994, twenty years, are the masses of humans confronted with a phenomenon, we still cannot make sense of. This may be so because computing technology is in everybody's hands, has really taken on media-quality, but is still not understood well.

My students don't watch TV any more. To boring, they say. Do they understand it? It's too slow for them. They have surrendered to increasing speed, to shallow emotion. They don't even program, but script a bit and believe it's programming. The sharp radical rational mind, merciless, and the grand overwhelming cry at the atrocities of capitalism, both must be our desire, our practice.

(b) Critical Engineering Manifesto

(0) The Critical Engineer considers Engineering to be the most transformative language of our time, shaping the way we move, communicate and think. It is the work of the Critical Engineer to study and exploit this language, exposing its influence.

(1) The Critical Engineer considers any technology depended upon to be both a challenge and a threat. The greater the dependence on a technology the greater the need to study and expose its inner workings, regardless of ownership or legal provision.

(2) The Critical Engineer raises awareness that with each technological advance our techno-political literacy is challenged.

(3) The Critical Engineer deconstructs and incites suspicion of rich user experiences.

(4) The Critical Engineer looks beyond the "awe of implementation" to determine methods of influence and their specific effects.

(5) The Critical Engineer recognises that each work of engineering engineers its user, proportional to that user's dependency upon it.

(6) The Critical Engineer expands "machine" to describe interrelationships encompassing devices, bodies, agents, forces and networks.

(7) The Critical Engineer observes the space between the production and consumption of technology. Acting rapidly to changes in this space, the Critical Engineer serves to expose moments of imbalance and deception.

(8) The Critical Engineer looks to the history of art, architecture, activism, philosophy and invention and finds exemplary works of Critical Engineering. Strategies, ideas and agendas from these disciplines will be adopted, re-purposed and deployed.

(9) The Critical Engineer notes that written code expands into social and psychological realms, regulating behaviour between people and the machines they interact with. By understanding this, the Critical Engineer seeks to reconstruct user-constraints and social action through means of digital excavation.

(10) The Critical Engineer considers the exploi to be the most desirable form of exposure.

(c) A/B (2009) Dunne & Raby

Affirmative / Critical
Problem solving / Problem finding
Provides answers / Asks questions
Design for production / Design for debate
Design as solution / Design as medium
In the service of industry / In the service of society
Fictional functions / Functional fictions
For how the world is / For how the world could be
Change the world to suit us / Change us to suit the world
Science fiction / Social fiction
Futures / Parallel worlds
The "real" real / The "unreal" real
Narratives of production / Narratives of consumption
Applications / Implications
Fun / Humour
Innovation / Provocation
Concept design / Conceptual design
Consumer / Citizen
Makes us buy / Makes us think
Ergonomics / Rhetoric
User-friendliness / Ethics
Process / Authorship

ACADEMIC PAPERS

Alpern, A. (1995). *Techniques for algorithmic composition of music*. Hampshire College.

Anand, A., Bhandari, B., (2009). F*ace detection in camouflage faces*. Dept. of Computer Science and Engineering, IIT Kanpur.

Kowalski, R. (1978). *Algorithm = Logic + Control*. Imperial College, London, England.

Verostko, R. (1988). *Epigenetic Painting Software As Genotype, A New Dimension of Art.* Presented at the First International Symposium on Electronic Art, 1988 (FISEA'88)

Wiest, L. J. (1990). *Olivier Messiane's Cinqu Rechants: the importance of rhythm as a structural element.* School of Music, University of Arizona

BOOKS

Agre, P. (1997). *Computation and Human Experience*. Cambridge, UK, Cambridge University Press.

Behrens, R. (2009). *Camoupedia: A Compendium of Research on Art, Architecture and Camouflage.* Bobolink Books

Carpenter, E. S. (1974). *Oh, what a blow that phantom gave me!*. 1st ed. New York: Holt, Rinehart and Winston.

Dalley, S. (1998). *The Legacy of Mesopotamia*. 1st ed. Oxford: Oxford University Press.

De Villedieu, A. (1240). *Carmen de Algorismo.*

Deleuze, G. (1992). *Postscript on the Societies of Control*. October #59, Cambridge, Mass, MIT Press.

Dunne, A., Raby, F. (2013). *Speculative Everything*. Cambridge, Mass, MIT Press.

Foucault, M. (1972). *The archaeology of knowledge*. New York, Pantheon Books.

Fuller, M., (2008). *Software Studies.* Cambridge, Mass, MIT Press.

Grout D. J., Palisca C. V. (1996). *A History of Western Music*. 1st ed. New York: W.W. Norton.

Hayles, K. (2005). *My Mother Was a Computer: Digital Subjects and Literary Texts*. 1st ed. Chicago: University of Chicago Press.

Hildebrandt, M., de Vries, K. (2013). *Privacy, Due Process and the Computational Turn*. Routledge.

Hillier, P. (1974). *Writings about Music*. 1st ed. Halifax: Press of Nova Scotia College of Art and Design.

Hofstadter, D. (1979). *Gödel, Escher, Bach: An Eternal Golden Braid*. 1st ed. New York: Basic Books.

Kitchin R., Dodge M. (2011). *Code/space: Software and Everyday Life*. Cambridge, Mass, MIT Press.

Laurel, B. (1989). *The Art of Human-Computer Interface Design*. 1st ed. Reading, Mass.: Addison-Wesley Pub. Co.

Manovic, L. (2002). The language of new media. Cambridge, Mass, MIT Press.

Manovich, L. (2014). *Software takes command*. London, UK, Bloomsbury publishing.

Marx, K., Engels, F. (1970). *Das Kapital*. Chicago: H. Regnery.

Messiaen, O. (1956). *The technique of my musical language*. 1st ed. Paris: A. Leduc.

Newman J. R. (1956). *The World of Mathematics.* 1st ed. New York: Simon and Schuster.

Parikka, J. (2012). *What is media archaeology?.* Cambridge, UK, Polity Press.

Rickey, G. (1995). *Constructivism: Origins and Evolution*. 1st ed. New York: G. Braziller.

Sethares, W. A. (2005). *Tuning, Timbre, Spectrum, Scale*. 1st ed. London: Springer.

Sigfried, G. (1948). *Mechanization takes command, a contribution to anonymous history*. 1st ed. New York: Oxford Univ. Press.

Wiener, N. (1948). *Cybernetics: or Control and Communication in the Animal and the Machine.* 1st ed. New York: M.I.T. Press.

Wirth, N. (1976). *Algorithms + Data Structures = Programs*. 1st ed. Englewood Cliffs, N.J.: Prentice-Hall.

Zielinski, S. (2006). *Deep time of the media: toward an archaeology of hearing and seeing by technical means*. Cambridge, Mass, MIT Press.

MAGAZINES

Kay A., 'Computer Software.' (1984) In: *Scientific American*, September issue.

Hoogendijk, J. P. (1998). *Pythagoras* #38.

Verostko, R. (1994). 'Notes on Algorithmic Art.' In: *ISEA Catalogue, The Fifth International Symposium on Electronic Art*

NEWSPAPER ARTICLES

Niedermeyer, J., Thomas, J. (2014). 'Face to Anti-Face.' In: *The New York Times*, December 14.

Shirey, D. (1972). 'Video Art Turns to Abstract Imagery.' In: *The New York Time*, July 4.

Wortham, J. (2012). 'A Surge in Learning the Language of the Internet.' In: *The New York Times*, March 27.

OTHERS

Lambert, N. (2000s) *Computer Art Thesis*. Ongoing online.

Pearson, M. (2011). *Novelty Waves*. Indie published e-book.

Spielmann, Y. (2004). *Video and Computer, The Aesthetics of Steina and Woody Vasulka.* Fondation Langlois.