

crash

The first attempt to model a crash paper involved writing software with a modular or nodal design and using communicating processes and frozen states of computation which can be embedded within each other, or run backwards and forwards; processes exchanging functionalities of interpretation and poetics subjected to changes in a timeline. We could consider a crash program which writes the crash paper or a paper which writes the program or is itself the program, given the definition of software as a descriptive means. Early conceptual code for crash acted as an interpreter for textual software, leading to the obvious self-application of the code itself which created problems such as being unable to do anything other than read itself and interpret itself. Work on the translation of such conceptual code into runnable Lisp code continues. Another attempt (the tape played during the symposium) involved the rough use of Burrough's tape cut-up technique as amateur shamanic ritual, relation to time and time reversal and simultaneity and reproduction of a Turing machine tape head.

As some background for the symposium and this paper it may well be worth summarising the roots of crash within the ap project fm01: film machine. fm01 is not conceived as an editing engine for the manipulation of generic clips (in an expanding database of all possible scenes categorised according to a huge number of elements and relations) but rather offers an enmeshing within script, data streams and environment; a total software environment for the semi-automated production, scripting and editing of endless cinema, transposing non-metaphoric systems and grammar theory (of computer languages, abstraction and data containers) to the realm of expanded cinema. and within this context a coding/scripting environment constructed by the operator. fm01 is concerned with viable/non-viable distributions of data across intensive nodes; environmentally situated event contingencies (decodings and recordings) within multiplexing structural couplings. fm01 investigates how data can radically be represented for the machinic (a question of interface) and critiques the implications of current meta-data approaches to the representation of data. As complex stratas of self-organising interactions present themselves, human-machine polyvalency overcomes traditional ontology and opens up the notion of an expanding environmental OS - where code is an environment of subsystems in search of form or pattern, orphaned bits without superstructure, and where the user as programmer is simultaneously inserted into this fugal, solaric codebase.

This paper thus traces a major, fake or feigned trajectory through basic components (seeking post-assembly in code) such as software as descriptive means, ontology of naked crash, Welt as simulation arriving at time reversal with a minor, real embedded trajectory describing trajectoriality; trajectoriality itself commencing with De Quincey's The English Mail Coach and journeying through other trajectorial means such as steam-train referenced vis a vis entropy and Los Alamos compression. Our subject is space/time trajectoriality itself in relation to crash and computation.

1] Descriptive means:

Software as a descriptive means or notation. Within Structure and Interpretation of Computer Programs [Sussman and Abelson]: "a subject organised around the notion that a computer language is primarily a formal medium for expressing ideas about methodology, rather than just a way to get a computer to perform operations." And describing there the emergence of "procedural epistemology": "the study of the structure of knowledge from an imperative point of view". That this paper could be itself written as software, as functionalities such as node or compression or as mechanisms including that of chunking itself and that we should make no differentiation between paper and (a piece of) software. What we're interested in here is abstraction within such descriptive means. Such a sub-component can be made obvious with reference to Richard Feynman's Lectures On Computation in which 13 levels to any operating system are described, from electronic circuitry to the shell.

2] Godel's theorem:

At its most basic: this statement is false. Godel's theorem of incompleteness is all about notation therefore Godel's theorem could be/has been re-stated in software and is software. Indeed such a notation could be seen as creating both software and the field of computation.

3] Halting problem:

The halting problem can simply be stated as "Given a description of an algorithm and a description of its initial arguments, determine whether the algorithm, when executed with these arguments, ever halts (the alternative is that it runs forever without halting)." Again, centring on the problems of self-application and multiplying meta-algorithms to determine if this one halts and so on. Our conclusion is that any expansive system cannot be termed either incomplete or complete. However, there's nothing new here (see the hacker classic Godel, Escher, Bach by Hofstadter) and the relation of incompleteness to simulation theory is another hacker staple. It's equally banal to state that software can and does change the world. We're not concerned with this field but rather the relation and question of computability and of provability following Bruno Marchal.

4] Noise and information theory fulcrum point:

From Friedrich Kittler "There is no software"

"Silicon hardware obeys many of the requisites for such highly connected, non-programmable systems. Between its millions of transistor cells, some million to the power of two interactions take place already; there is electronic diffusion, there is quantum mechanical tunnelling all over the chip. Yet, technically, these interactions are still treated in terms of system limitations, physical side-effects, and so on. To minimise all the noise that it would be possible to eliminate is the prize paid for structurally programmable machines. The inverse strategy of maximising noise would not only find the way back from IBM to Shannon, it may well be the only way to enter that body of real numbers originally known as chaos."

Further sub-components here:

A solution to the halting problem would state that there is decidability or computability applied to always abstraction of 1 and 0 no logic built from a logic gate undecidable switch. Within quantum computing a qubit can exist not only in a state corresponding to the logical state 0 or 1 as in a classical bit, but also in states corresponding to a blend or superposition of these classical states. In other words, a qubit can exist as a zero, a one, or simultaneously as both 0 and 1, with a numerical coefficient representing the probability for each state. Quantum mechanics predicts that the photon can travel both two simultaneously, collapsing down to one path only upon measurement.

Repetition of data, or in other forms a repetition which ensures the safe arrival of that data (within a layered protocol which is layered in terms of abstraction). The question arises as to how long to wait for such a repetition to become evident, to distinguish a meaningful message from noise (protocol and halting problem).

Decoherence, or the tendency of a quantum computer to decay from a given quantum state into an incoherent state as it interacts, or entangles, with the state of the environment. These interactions between the environment and qubits are unavoidable, and induce the breakdown of information stored in the quantum computer, and thus errors in computation. Again a question of error correction and the necessary repetition to discover, within a context of information theory and entropy, what was meant by a message (the measurable information content).

5] An ontology of crash:

Crash is always in relation to a (necessarily programmable) machine. A machine does not crash unless it is a Turing machine. Indeed, crash can readily be seen as the exposure of the programmable and machinic in that what perhaps was not necessarily viewed as machine/coded is now revealed as such. Crash exists in relation to an expanded interface or operating system and in relation to exposure and revelation. Types of crash can be enumerated and graded, ranging from a crash in GUI, or an application failure (due to a segmentation fault - the operating system halts a promiscuous application) to a total system freeze (experienced always at the level of the interface). Crash returns the user to a lower

state of abstraction; a crash within a GUI can be resolved within the shell; we need to find some way to hook in to the frozen machine. Broken software can be debugged at a hardware level. Crash as interruption within OS or application functionality, a fatal exception, a segmentation fault - doing or accessing what is forbidden for a process, what is outside the system of notation or bounds of an application. For example, a segmentation fault occurs when a program tries to access memory locations that haven't been allocated for the program's use. Under a Turing machine, if the transition function is not defined for the current state and current symbol, then the Turing machine crashes or halts. Crash implies both exposure and cynical delight in that naive revelation: witness magazines and sites devoting pages to images of crashes on location, in the real world - in railway stations, on ATMs, at airports, and in shop displays. What we're interested in here is crash in relation to a generalised OS (in-turn of exposure of machinic OS and of OS considered as non-machinic or under larger embrace of the machinic). The OS manages all system resources and provides for a stable and consistent way for all applications to access lower levels of abstraction such as hardware. The OS provides an interface to the real world and for the user. The OS is an interface to resources. Given the contemporary field of equivalence of hardware and software, an OS can now only be considered as managing and securing economies and resources in any field begging the obvious question of the consequences of this for a theory of crash.

Crash is extrusion, revelation and in-turn on the fissure of decidability and of noise and information (see h-bomb information component; in the words of Seth Lloyd: "A one-kilogram chunk of matter converted completely to energy - this is a working definition of a 20-megaton hydrogen bomb. An exploding nuclear weapon is processing a huge amount of information, its input given by its initial configuration (data encoded as particle positions, velocities and spins - MH) and its output given by the radiation it emits"). Within a realm of perfect abstractions or supreme black boxing there can be no crashing, in that crashing can be viewed as leakage, for example directly illustrated by the case of a buffer overflow under which data is written beyond its predefined bounds into the private space of another process.

Finally, within such a component we could consider a crash into certainty within the context of a crash revelation or resolution of quantum state.

6] World/universe as simulation or Welt am Draht:

Welt am Draht, or World on a Wire is a two part television series from the mid 1970s directed by Rainer Werner Fassbinder and based heavily on Daniel Galouye's Simulacron-3, also known as Counterfeit World.

"Simulacron-3 (1964) suggests a virtual world within the "real" world, which in turn was another virtual world within the real world. When the creator of the virtual world, stumbles upon a secret, he is killed, and his successor becomes the primary suspect. Thanks to a giant computer that allows people to "download" into the virtual world, the successor journeys to the virtual world to clear himself; but he quickly

discovers that the world he lives in is itself a simulation, and that he is nothing more than a computer program within someone else's simulation. Simulacron-3 was made into the highly underrated "The Thirteenth Floor," which came out just a few months after "The Matrix" in 1999. Daniel Galouye's story anticipates the layers of deception that make up the world of the Matrix, more than thirty-five years earlier."

" I sat there for an eternity, staring incredulously ahead.

The road ended a hundred feet away.

On each side of the strip, the very earth itself dropped off into an impenetrable barrier of stygian blackness.

Out there were no stars, no moonlight -- only the nothingness within nothingness that might be found beyond the darkest infinity."

In contrast to this banal revelation of dully hitting edge of a simulation (a soft crash exposure) Fassbinder's essential work posits embeddings of simulations/overlap and ill-decision between realities and further embeddings of non-deterministic post-Turing machines simulating and spawning and continuing the processes of other simulated post-Turing machines. Fassbinder suggests both insertion and overlap of individuals/universes. It's worth noting that such works are as themselves filmic and within the filmic, and thus the context of fm01 project, suggest the possibility of simulation in the non-observer/all-observer flatness of film.

Kopf as a reflexive simulation machine within the relation of the notion of reality as simulation to paranoia. The entire oeuvre of Philip K Dick essentially outlines this knit, and his *Through A Glass Darkly*, with an overlap of observer and observed subject, is central to this literature which can also include Fowle's *The Magus* and through direct citation there, Burrough's *Cities of the Red Night*.

As Jonathan Kemp observes with reference to La Mettrie's *L'Homme Machine*, from 1750, humanity is readily viewed in the image or model of whatever machine dominates contemporary life. To quote Kemp, "in 1750 with La Mettrie "*L'Homme Machine*" it was the clock - and in the 60's a watered down vision of man machine symbiosis was put forward by Licklider as a means for controlling complexity - typically nowadays we call a computer's storage of data a "memory", and HCl's impose functionalist models on the user as an idealism, fine tuned and executed (and conceptually stalled) as the user is effectively incorporated into its models."

But it goes much further than seeing humanity in terms of computation, rather computation reflexively impacts on ontology and notions of the universe. There may well be nothing new here, following a course outlined by La Mettrie, but what is novel is this notion of crashing, which embeds "of itself and not itself" as outlined in *Welt am Draht* and less so in the rest of works such as *The Matrix* (ignored for evident

conceptual and aesthetic reasons), which share with *The 13th Floor* (another descendant of Galouye's *Simulacron-3*) the moment of bland revelation, hitting an edge of the simulation with a truth beyond. *Mise en scene* of nested illusions. In contrast *Welt am Draht* suggests an overlapping of interior *Welt* simulation with making simulation (the extruded machinery of simulation), and being outside; a mapped interior/exterior *kopfschmerz* of two holdings, constructing and constructed overlap which is crash; overmapping of the trajectorial at/on multiple simultaneous sites. One last note relating to *Simulacron-3*: a drawing found amongst the research notes of a previously disappeared scientist (who is aware of the simulated nature of this world) illustrates Zeno's paradox.

7] Crash of a reality-bound OS

8] Universe as computer

Juergen Schmidhuber identifies Conrad Zuse's 1967 *Calculating Space* (*Rechnender Raum*) paper as "the first to propose that physics is just computation, suggesting that the history of our universe is being computed on, say, a cellular automaton." It's interesting to note that Zuse is regarded as building the first programmable computers (1935-1941). In the book *Calculating Space* (1969), Zuse addresses questions of digital measurement and modelling of physics; if it can be simulated or absolutely measured mathematically it can/is simulating itself/be simulated. The digital is defined in relation to the lack of a continuum, stressing rather the discreet states of automata. The classical model of physics is analogue in nature, however quantum physics is hybrid, being occupied with both continuous and discrete values.

Schmidhuber further elaborates such notions as "digital physics", and that "the simplest explanation of our universe is the simplest program that computes it. In 1997 Schmidhuber pointed out that the simplest such program actually computes all possible universes with all types of physical constants and laws, not just ours."

Within the essay *Black Hole Computing*, Seth Lloyd and Jack Y. NG argue that rather than offering an exception to the rule that everything computes, black holes rather form an extreme and compressed example for this model of the universe as quantum computer computing itself (as its own code and data).

"the real question is whether Hawking radiation (emitted by the black hole) returns the answer of the computation or merely gibberish. The issue remains contentious, but most physicists, including Hawking, now think that the radiation is a highly processed version of the information that went into the hole during its formation. Although matter cannot leave the hole, its information content can. Understanding precisely how is one of the liveliest questions in physics right now.

Last year Gary Horowitz of the University of California at Santa Barbara and Juan Maldacena of the Institute for Advanced Study in Princeton, N.J., outlined one possible mechanism. The escape hatch is entanglement, a quantum phenomenon in which the properties of two or more systems remain correlated across the reaches of space and time. Entanglement enables teleportation, in which information is transferred from one particle to another with such fidelity that the particle has effectively been beamed from one location to another at up to the speed of light."

"What is the universe computing? As far as we can tell, it is not producing a single answer to a single question, like the giant Deep Thought computer in the science-fiction classic *The Hitchhiker's Guide to the Galaxy*. Instead the universe is computing itself. Powered by Standard Model software, the universe computes quantum fields, chemicals, bacteria, human beings, stars and galaxies. As it computes, it maps out its own spacetime geometry to the ultimate precision allowed by the laws of physics. Computation is existence."

9] Los Alamos compression beginning with De Quincey expansion - a time-line or trajectory:

When I first started relating De Quincey's essay "The English Mail Coach" to the concerns of crash in terms of a technique of stylistic or linguistic compression (as admitted by De Quincey in an explanatory notice addressing critics who seemed unable to grasp his combinatorial logic), this particular juxtaposition came without the realisation that the complete essay compresses 70 odd seconds of potential impact or crash, a crash/collision always yet to come or operating retroactively in description and unravelling/overlapping/multiplying in the dreams described. Thus J G Ballard can also be brought back into the equation - again considered as a further compression, an out-turn in some sense.

De Quincey was introduced into the crash equation primarily as a pre-extension for a trajectory describing the passage from the steam engine in relation to entropy to information theory in relation to entropy and the birth of such theory at Los Alamos (birth place of hydrogen bomb as supposedly hyper-entropic in heat death or hyper-computational according to Seth Lloyd).

Within a history of the universe as computation a Los Alamos compression could provide the birth of a model of computation applied to the universe which embeds a subsequent theory of the universe as computer (Seth Lloyd) with a 10 megaton H bomb as the ultimate contemporary computer.

10] Time-reversal

Post-simulation and thus pre-simulation time operates in reverse. The simulation is retro-active and writes/works backwards.

We return to paranoia as a central thematic but now outfitted within the ubiquitous mockup, dress rehearsal, and stage scenery of Pynchon's Gravity's Rainbow (a supreme investigation of time and trajectory) in relation to the timing of rocket soundwave apparition, Pavlov and erection; backwards and forwards directions in time engendered by a yet-to-be-reached yet-to-be-scripted recursive simulation manufacture which operates post the event and re-writes events.

Thus, forming a suitably melodramatic backdrop, the iconic ur-crash of Lady Diana can be seen as the first stage of a process leading to the future murders of Jack the Ripper. Crash involves traversal of space and traversal of time (Zeno again) in this unfolded dream sequence which parallels De Quincey.

martin howse february/march 2005