Georg Trogemann, Ursula Damm

Algorithms rethought – on the fictional potential of software

Published in „Idee Inhalt Form. Beiträge zur Gestaltung der Gegenwart", by Prof. Dr. Winfried Speitkamp and Claudia Weinreich on behalf of the Bauhaus University Weimar on the occasion of the 100th anniversary of the Bauhaus 2019

Fictional narratives create unique worlds with their own facts, incidents, objects and laws. But fictional worlds remain capable of connection and expansion too; they ensure continuity, so the fiction can form as a whole in the imagination of the viewer and the reader, so that they can expand on it and even answer questions that are not explicitly resolved in the fictional representation itself. In the following we would like to describe a different fictional area, beyond the well-known genres of film, literature and theatre: algorithms and their runtime environments. It is peculiar to algorithms, as a successor to mathematics, that they open up spaces of possibility. They represent extreme densifications of innumerable individual cases, which the programmer creates through the programme structure and which unfold at the runtime of the algorithm and manifest at the interface of the machine as images, sounds and actions. Algorithmic areas of action not only increasingly determine our vision of reality, they create our reality, as concretisations within a self-created field of possibility. So algorithms are understood, in particular, as a cultural technology, whose aesthetic layers should be exposed. Using Membrane, a generative video installation, as an example, we will describe an artistic sphere within artificial intelligence. Through this we will question whether algorithms are capable of generating something new and unthought of that could attain value in the context of art.

The Philosophy of 'As If'

Hans Vaihinger's controversial magnum opus The Philosophy of 'As If' was published in 1911. In numerous examples, Vaihinger pursues the question of how one can achieve the right result despite starting with false assumptions. He already formulates his central question in the first few preliminary remarks to the introduction of the book: "How can it be that we achieve the right conclusion with consciously false assumptions? We operate with 'atoms,' although we know that out concept of the atom is arbitrary and wrong, and the curious thing is, we operate happily and successfully with this false concept: we wouldn't cope well without it, we might not even achieve our goals." Vaihinger's answer is surprisingly simple. The false assumptions represent practical, useful fictions. With that, all human knowledge and all findings can only ever be substantiated pragmatically, i.e. by the success that emerges in the use of fictions. It is not about freeing oneself from these fictions, but rather recognising their necessity, understanding them as tools which make thinking possible in the first place, and adapting to them. It is worth noting that Vaihinger had already formulated his philosophy of 'as if' in 1877 in his habilitation thesis Logische Untersuchungen. I. Teil: Die Lehre von der wissenschaftlichen Fiktion. American pragmatism did not yet exist when Vaihinger habilitated. For Vaihinger, fictions are ideas which we know are not durable, that we cannot find a representative for in reality. But semiotics and the philosophy of language also tell us that language does not create a mirror image of reality, rather thata we construct our vision of reality by means of our linguistic distinctions. Initially, all linguistic symbols refer solely to themselves and even if their link to the world can be well reasoned, it always remains at risk. With every inference, with every advance in imagination and language, the connection can be lost and must be reevaluated in reality. More can be described with language than we actually find in the world. Every form of fiction draws from these excesses of imagination and language.

Fiction and Algorithm

It would appear the starting point is a completely different one when we talk about algorithms. These are technical processes with mathematical roots. Algorithms are not formulated in natural languages, but rather in formal ones, they are clear, successive procedures towards the solution of precisely formulated problems. At every point in time the operation to be executed next is clearly defined. At first glance there would appear to be no room for fictions here: these are solutions to problems. It is precisely these deterministic formulas of action that now shape our world to an unprecedented extent, as structural elements of our everyday life: autonomous driving, automated credit granting, algorithmic share trading, predictive policing, partner searches: an automated agency emerges whose social impacts are only just beginning to become discernible. Alongside developments in informatics and technology and their specific problems, algorithms are currently mainly considered from the aspects of profitability, safety, responsibility, fairness, ethics, the protection of individual and human rights or the private sphere. If one looks at algorithms from the perspective of what Vaihinger calls fiction, another picture emerges. At least four different forms of fictional strategy involved in the implementation and execution of algorithms can be distinguished. The first two are, to some extent, tricks of the mind, which enable these technically complex systems to be realised and manageable in the first place. In the third and fourth forms, the algorithms themselves become fiction generators. Only here does their novel potential and effectiveness beyond familiar mathematical and formal fictions emerge.

1. Mathematical Fictions

The emergence of computer technologies can be understood historically as the merger of mathematics and engineering. Mathematical operations become mechanically resolvable through computers. The basic principles of mathematics include, among others, empty space, empty time, points without area, lines without width, surfaces without depth and much more. Consequently, all of these fictions of mathematics are found in our algorithms too. As operations of analytical geometry they form, for example, the basis of computer graphics and computer-generated imagery in general. That which works at the core of algorithms to this day, numbers, variables, functions etc., are all inventions of mathematics. Mathematics takes up a key position in what we now call digitalisation. It can be used as a means of knowledge, in order, for example, to describe and understand the makeup of materials in physics. But it is increasingly becoming a means of poiesis in computers, in the active construction, production and transformation of the world, for example in engineering, architecture, robotics, life sciences etc. Here a radical shift in the meaning of mathematics by digital technologies becomes apparent. A tool of analysis and discovery increasingly becomes a tool of synthesis. Ideas, which have proven themselves practical in mathematics, no longer remain pure fictions; we use them as if they existed. The Cartesian space, that Vaihinger still describes as a fiction of mathematics, becomes a fact through the robot arm; points in space that were once only imagined become controllable, as if they existed.

2. Levels of software

Computer hardware can only process so-called machine language. Programming at this level would not only be very time consuming, but also very error-prone. It would be almost impossible to realise very large applications, as one must work directly with the binary representation of numbers and commands on this lowest level, closest to the hardware. The user is therefore offered various levels of software, for interacting with and programming the computer, which sit on top of the machine code. The lowest level is assembly language, in which no abstraction of the processor architecture takes place. The higher levels, especially in higher programming languages, are completely abstracted from the hardware. The software can run on all computers that offer an interpreter or compiler for the respective language. The different programmer paradigms developed in computer sciences are also fictions by Vaihinger's definition. Every programming language must offer certain functionalities so that solutions can be efficiently described. But the various paradigms should, in particular, guide the thinking of the programmer and help them to structure their complex tasks and avoid reasoning errors in programming that would lead to failure of the programme at runtime. In this way, for example, object-oriented programming claims to adjust the software architecture to the specific area of reality of its use. But these are mainly modelings that create a new reality rather than depicting a given one, like how the trash can on a desktop computer interface only represents a very restricted, functional abstraction of a real trash can. I can ignite a real trash can, turn it upside down and stand on it, in order to reach the top book on a shelf. The programming language object remains a fiction, only accessible via its predetermined functionality. Ontologies fulfil a similar role in computer science, putting the relevant number of terms for a certain subject area and the relationships between them into words and presenting them in a formally ordered way. This is how artificial worlds are described, abstracted and demarcated. That which is not part of the ontology, does not exist.

3. Runtime environments

In order for tasks and problem areas to become computable, they must first be formalised. Formalisation is essentially the abstraction of the fullness of real conditions. We reduce the complex entity of the given to a few characteristics, which we assume completely or at least sufficiently cover the problem. The seemingly insignificant is removed, while the essential is named and brought to the forefront. This abstraction also takes place in the sensors and interfaces that are read and handled by algorithms. The entirety of an external occurrence is reduced to quantifiable proportions in the sensors. So abstraction identifies what, in terms of the objectively existent, must be subtracted and what ideas, so what is not existent, must be added, so that phenomena of our world can acquire form. Yet this only describes the path of phenomena into the machine. With the execution of codes, formalisms now become operative and create, for their part, phenomena. The abstraction process now runs backwards, so to speak, and becomes concretisation. While strict formal operations are carried out on the level of code, the images, sounds, actions at the interface of the machine are augmented again with unforeseeable artefacts, with side effects and unintentional consequences, which open up new interpretational scope for the viewer. The materiality of the machine and its environment create phenomena that are unforeseeable. If the visual, tactile, acoustic experiences generated by the programme are at the forefront of an application, as is mostly the case in the art context, then all that remains is to move towards the satisfactory image, the convincing movement or the moving sound through iterative testing and modification. The question of whether it "feels" right can not be decided at the level of formal codes, even though this is the source of the effect.

4. Ranges of possibility and reflexivity

Algorithms are general, which means they do not solve a specific task, for example the question "what is 3 multiplied by 7?" but rather a class of problems, for example multiplication. This class of problems can be made up of a very large, in principle infinite, number of concrete tasks. An algorithm that generates a parameter-controlled image can create other, different images if we vary its parameters. The individual image is still only a representative of the image range laid out by the algorithm. If we read the programme code as text, we can therefore also understand it as a compression of a range of possibility. If we no longer understand programme codes as solutions to problems, but rather as compact descriptions of ranges of possibility, another perspective of the algorithm arises. We are no longer interested in the correctness of the process, but rather the fictive and aesthetic potential they offer us. If we act speculatively with algorithms, without a fixed task, then other questions are posed of software development. The challenge is to develop programmes that open up interesting areas of possibility, and to move inquisitively through them. Reflexive strategies prove especially helpful in this. By reflexive we mean programme structures in which parts of the outcome are used to modify the code or at least its parameters. In this way, movement through the respective range of images, sounds, movements, forms etc. is given direction. Reflexive operations that imitate principles of nature—Vaihinger would also justifiably call these abstractions and transmissions fictions—include, among others, genetic algorithms and neuronal networks.

Membrane or How to Produce Algorithmic Fiction

Membrane is an art installation which has been first exhibited in Berlin in spring 2019. It builds on a series of generative video installations with real time video input. Membrane allows the viewer to interact directly with the generation of the image. In doing this it is possible to experience the 'imagination' of the computer, guiding the process according to curiosity and personal preferences.
An image on a computer is represented by a matrix of RGB values along an x- and a y-axis. A digital image consists of a grid of pixels which take their values from a spectrum of hues (usually 256 per channel). In the case of Membrane, these images are derived from a static video camera observing the Turmstrasse, a bustling shopping street in before the windows of the exhibition space. In this respect we can view the image as a snapshot of the place within a certain time frame and thus we are able to interfere with the temporal alterations of the image on an algorithmic level via software.
Generally, self-organised neural networks make reductions to the complexity of a set of data which, at first glance, looks like a video filter. The possible interpretations of an image are systematically reduced until you can make a statement about the content of the image (for example: yes, it's a dog or no, there is no dog). Neural Networks, a form of machine learning, work with interconnected sensors which can store information in addition to having perceptive capabilities. These so-called neurons measure and judge packets of data which are channeled through them whilst simultaneously adapting to the processed data. A reciprocal feedback is established which operates without the use of external categories. A neuron is defined solely by the architecture of the network (i.e the connections between the neurons). The set of neurons generates an abstract, netlike learning environment which is able to (re)produce data sets with specific properties and make evaluations for resemblances or distinctions. Humans perceive these procedures as 'subsymbolic', working without labels, symbols or metaphors.

In my earlier artistic experiments within this context (http://ursuladamm.de/transits-2012/) we considered each pixel of a video data stream as an operational unit. One pixel learns from colour fragments during the running time of the program and delivers a colour which can be considered as the sum of all colours during the running time of the camera. This simple method of memory creates something fundamentally new: a recording of patterns of movement at a certain location (http://ursuladamm.de/598/). Here, it becomes obvious how the arrangement of input devices, processing and output is able to create new worlds and contexts.

On a technical level, Membrane not only controls pixels or clear cut details of an image, but image 'features' which are learnt, remembered and reassembled. With regards to the example of colour: we choose features but their characteristics are delegated to an algorithm. TGANs (Temporal Generative Adversarial Nets, https://pfnet-research.github.io/tgan/, https://arxiv.org/abs/1611.06624) implement 'unsupervised learning' through the opposing feedback effect of two subnetworks: a generator produces short sequences of images and a discriminator evaluates the artificially produced footage. The algorithm has been specifically designed to produce representations of uncategorised video data and - with the help of it - to produce new image sequences.
We extend the TGAN algorithm by adding a wavelet analysis which allows us to interact with image features as opposed to only pixels from the start. Thus, our algorithm allows us to 'invent' images in a more radical manner than classical machine learning would allow. In practical terms, the algorithm speculates on the basis of its learning and develops its own, self organised temporality. However, this does not happen without an element of control: feature classes from a selected data set of videos are chosen as target values. In our case, the dataset consists of footage from street views of other cities from around the world, taken while travelling.

The concept behind this strategy is not to adapt our visual experience in Berlin to global urban aesthetics but rather to fathom the specificity and to invent by associations. These associations can be localised, varied and manipulated within the reference dataset. Furthermore, our modified TGAN algorithm will generate numerous possibilities to perform dynamic learning on both short and long timescales and ultimately to be controlled by the user/ visitor. The installation itself allows the manipulation of video footage from an unchanged street view to a purely abstract images, based on the found features of the footage. The artwork wants to answer the question of how we want to alter realistic depictions. What are the distortions of 'reality' we are drawn to? Which fictions are lying behind these 'aberrations'? Which aspects of the seen do we neglect? Where do we go with such shifts in image content and what will be the perceived experience at the centre of artistic expression?

The fictional potential of machine learning has become popular through Google's deep-dream algorithms. Trained networks synthesise images on the basis of words and terminologies. They are reconfiguring symbols which they allegedly deem to recognise. From an aesthetic perspective, these images look paranoid; instead of presenting a consistent approach, they tail off in formal details and reproduced previously found artefacts (through searching the internet). From an artistic point of view, the question now arises, how can something original and new be created with algorithms? This is the question behind the software design of Membrane. Unlike Google's deep-dream algorithms and images, we don't want to identify something specific within the video footage (like people or cars) but rather we are interested in how people perceive the scenes. That is why our machines look at smaller, formal image elements and features whose intrinsic values we want to reveal and strengthen. We want to expose the visitors to intentionally vague features: edges, lines, colours, geometrical primitives, movement. Here, instead of imitating a human way of seeing and understanding, we reveal the machine's way to capture, interpret and manipulate visual input. Interestingly, the resulting images resemble pictorial developments of classical modernism (progressing abstraction on the basis of formal aspects) and repeat artistic styles like Pointilism, Cubism and Tachism in a uniquely unintentional way. These styles fragmented the perceived as part of the pictorial transformation into individual sensory impressions. Motifs are now becoming features of previously processed items and are successively losing their relation to reality. At the same time, we question whether these fragmentations of cognition are proceeding in an arbitrary way or whether there may be other concepts of abstraction and imagery ahead of us?

If we look at Membrane in more detail, taking into consideration the attempted typology of fictional strategies in relation to algorithms established above, then all four strategies appear. We find both mathematical fictions as well as software fictions and fictions of runtime environments in the installation Membrane. In conclusion we would like to look at the fourth area, the ranges of possibilities and reflexivity, in a little more detail. It is important to state that the algorithms developed in computer sciences are not applied as tools of analysis here, but rather as tools of synthesis. It is not about the recognition of an image content, but rather production, in particular, in the present case, of moving image patterns. The software should be able to react to viewer preferences in a, per se, open field of development possibilities. How can we achieve this?

In the book A New Kind of Science, published in 2002, Stephen Wolfram lays out how very simple rules and programmes can create great complexity using the example of simple cellular automata. By means of one-dimensional automata he shows that qualitatively different classes of behaviours can be found and very simple rules already possess the power of universal calculus. Wolfram examined all 256 possible one-dimensional, binary automata. Such analysis can already no longer be comprehensively carried out for two-dimensional automata. If we view the individual image of the present installation, for example, as two-dimensional automata, in which the state of each point in the image consists of three 256-figure numbers and simultaneously remains open to processing rules that depend on the nearest neighbouring points, this would result in a range of possibilities for image generation that is no longer analytically manageable. Feedback adds a further level of complexity. So the question is: how do we deal with such extremely large ranges of possibility? The answer should no longer be surprising: We let ourselves be guided by fictions in programme development. We know that our brains work differently to artificial neuronal networks. Even the form of coding differs fundamentally: frequency coding for the human brain versus amplitude coding for artificial neuronal networks. Even if we speak of the perceptual capacity of networks, of learning, neurons, memory and similarity, these are little more than metaphors. Essentially, the operations can also be described completely in the language of mathematics. As although Wolfram's cellular automata develop without any reflexivity to an exterior and in our example reflexive strategies should ensure a connection between the status of the machine and a status of reality, our programme operations also remain within the scope of mathematical possibility. But the software-side construct of neuronal networks offers a visualisation that clearly helps us to tackle the unmanageable scope of possibility of our (digital) images not only more vividly, but also more precisely and to interpret the results with regard to the model, even though the reality we are talking about remains a constructed one. As soon as we stop identifying and analysing, but rather put the potential of algorithms to synthesise centre stage, operational classes such as neuronal networks, genetic algorithms and many other methods are little more than helpful metaphors that enable us to deal with the surplus of possibilities in rule-based action.

From a cultural perspective, there are two questions remaining: How can one take decisions within those aesthetic areas of action (parameter spaces)? Can the shift of the perspective from analysis to fiction help to asses our analytical procedures in a different way - understanding them as normative examples of our societal fictions serving predominantly as a self-reinforcement of present structures? Thus unbiased artistic navigation within the excess/surplus of normative options of actions might become a warrantor for novelty and the unseen.

The original german edition can be found here.