



4.02

Algorithms rethought – zum fiktionalen Potenzial von Software

Ursula Damm
Georg Trogemann

Fiktive Erzählungen erschaffen andersartige Welten mit eigenen Tatsachen, Begebenheiten, Dingen und Gesetzen. Doch auch fiktive Welten bleiben anschlussfähig; sie gewährleisten Kontinuitäten, damit die Fiktion als Ganzheit in der Vorstellung der Betrachterin und des Betrachters, der Leserin und des Lesers entstehen kann, damit diese weiterdenken und sogar Fragen beantworten können, die in der fiktionalen Darstellung selbst nicht explizit aufgelöst werden. Jenseits bekannter fiktiver Genres wie Film, Literatur, Theater wollen wir im Folgenden einen neuen, andersartigen fiktionalen Handlungsraum beschreiben: Algorithmen und ihre Laufzeitumgebungen. Als Erbe der Mathematik ist es Algorithmen eigen, Möglichkeitsräume aufzuspannen. Sie repräsentieren extreme Verdichtungen von unzähligen Einzelfällen, welche der Programmierer oder die Programmiererin durch die Programmstruktur kreiert und die sich zur Laufzeit des Algorithmus entfalten und an der Oberfläche der Maschine als Bilder, Töne, Handlungen manifestieren. Algorithmische Handlungsräume bestimmen nicht nur zunehmend unser Bild von der Wirklichkeit, sie erzeugen unsere Wirklichkeit, als Konkretisierungen im selbst erzeugten Möglichkeitsraum. Algorithmen werden also insbesondere als Kulturtechnik verstanden, deren ästhetische Schichten es freizulegen gilt. Am Beispiel von »Membrane«, einer generativen Videoinstallation, beschreiben wir einen künstlerischen Handlungsraum innerhalb Künstlicher Intelligenz. Dabei stellen wir die Frage, ob Algorithmen Neues, Ungedachtes hervorzubringen in der Lage sind, das eine Wertigkeit im Kontext von Kunst zu erreichen vermag.

»Die Philosophie des Als Ob«

Im Jahr 1911 erscheint Hans Vaihingers umstrittenes Hauptwerk »Die Philosophie des Als Ob«. In zahlreichen Beispielen geht Vaihinger der Frage nach, wie man ausgehend von falschen Annahmen dennoch zu Richtigem gelangen könne. Schon in den ersten Sätzen der Vorbemerkungen zur Einleitung des Buches formuliert er seine zentrale Frage: »Wie kommt es, dass wir mit bewusst falschen Vorstellungen doch Richtiges erreichen? Wir operieren mit ›Atomen‹, obgleich wir wissen, dass unser Atombegriff willkürlich und falsch ist, und, was eben das merkwürdige ist, wir operieren glücklich und erfolgreich mit diesem falschen Begriff: wir kämen ohne ihn nicht so gut, ja überhaupt nicht zum Ziele.« Vaihingers Antwort fällt überraschend schlicht aus. Die falschen Annahmen stellen praktisch nützliche Fiktionen dar. Damit kann alles menschliche Wissen und alle Erkenntnis immer nur pragmatisch begründet werden, also durch den Erfolg, der sich in der Anwendung der Fiktionen zeigt. Es geht also nicht darum, sich von diesen Fiktionen zu befreien, sondern darum, ihre Notwendigkeit anzuerkennen, sie als Werkzeuge zu verstehen, die das Denken erst ermöglichen, und sich mit ihnen einzurichten. Bemerkenswert ist, dass Vaihinger seine Philosophie des Als-Ob schon 1877 als Habilitationsschrift mit dem Titel »Logische Untersuchungen. I. Teil: Die Lehre von der wissenschaftlichen Fiktion« ausformuliert hatte. Als Vaihinger habilitiert wurde, existierte der amerikanische Pragmatismus also noch nicht. Fiktionen sind bei Vaihinger Vorstellungen, von denen wir wissen, dass sie nicht haltbar sind, dass wir keine Vertreter in der Realität finden können. Aber auch Semiotik und Sprachphilosophie sagen uns, dass Sprache kein Spiegelbild der Wirklichkeit erzeugt, sondern wir unsere Auffassung von der Realität mittels unserer sprachlichen Unterscheidungen erst konstruieren. Alle sprachlichen Zeichen beziehen sich zunächst nur auf sich selbst und auch wenn ihr Bezug zur Welt gut begründbar ist, bleibt er trotzdem immer gefährdet. Mit jeder Schlussfolgerung, mit jedem Weiterspinnen in der Vorstellung und der Sprache kann der Zusammenhang verloren gehen und muss neu in der Realität geprüft werden. Mit Sprache lässt sich mehr beschreiben, als wir in der Welt wirklich vorfinden. Aus diesen Überschüssen der Vorstellungskraft und der Sprache schöpft jede Form der Fiktion.

Fiktion und Algorithmus

Wenn wir von Algorithmen sprechen, scheint die Ausgangslage eine komplett andere zu sein. Es handelt sich um technische Verfahren mit mathematischen Wurzeln. Algorithmen werden nicht in natürlichen, sondern in formalen Sprachen formuliert, es sind

eindeutige, schrittweise Verfahren zur Lösung präzise formulierter Probleme. Zu jedem Zeitpunkt ist eindeutig definiert, welche Operation als nächstes auszuführen ist. Auf den ersten Blick scheint es hier keinen Raum für Fiktionen zu geben, es geht um Problemlösungen. Es sind genau diese deterministischen Handlungsschemata, die heute in ungekanntem Ausmaß unserer Leben prägen, als Konstruktionselemente unseres Alltags: autonomes Fahren, automatisierte Kreditvergabe, algorithmischer Aktienhandel, predictive policing, Partnersuche: Eine automatisierte Handlungsmacht entsteht, deren gesellschaftliche Auswirkungen heute erst in der Frühphase erkennbar sind. Neben der informatisch-technischen Entwicklung und ihrer spezifischen Probleme werden Algorithmen derzeit vor allem unter den Gesichtspunkten ihrer Wirtschaftlichkeit, Sicherheit, Verantwortung, Gerechtigkeit, Ethik, des Schutzes der Individual- und Menschenrechte oder der Privatsphäre behandelt. Betrachtet man Algorithmen aus dem Blickwinkel der Fiktion im Sinne Vaihingers, zeigt sich ein anderes Bild. Es lassen sich mindestens vier unterschiedliche Ausprägungen fiktionaler Strategien ausmachen, die bei der Implementierung und Ausführung von Algorithmen beteiligt sind. Bei den ersten beiden handelt es sich gewissermaßen um Kunstgriffe unseres Denkens, durch die diese technisch komplexen Systeme überhaupt realisiert und handhabbar werden. Bei der dritten und vierten Form werden die Algorithmen selbst zum Fiktionsgenerator, erst dort zeigen sich das neuartige Potenzial und eine Wirkkraft jenseits der bekannten mathematisch-formalen Fiktionen.

1. Mathematische Fiktionen

Das Entstehen der Computertechnologien lässt sich historisch als Zusammenschluss von Mathematik und Ingenieurwesen verstehen. Mathematische Verfahren werden durch Computer maschinell lösbar. Zu den Grundbegriffen der Mathematik gehören unter anderem der leere Raum, die leere Zeit, Punkte ohne Ausdehnung, Linien ohne Breite, Flächen ohne Tiefe und vieles mehr. Alle diese Fiktionen der Mathematik finden sich folgerichtig auch in unseren Algorithmen wieder. Als Verfahren der analytischen Geometrie bilden sie beispielsweise die Basis der Computergrafik und computergenerierter Bildwelten im Allgemeinen. Was bis heute im Kern der Algorithmen wirkt, die Zahlen, Variablen, Funktionen etc., sind Erfindungen der Mathematik. Die Mathematik nimmt bei dem, was wir heute Digitalisierung nennen, eine Schlüsselposition ein. Sie kann als Mittel der Erkenntnis genutzt werden, um beispielsweise in der Physik den Aufbau der Materie zu beschreiben und zu verstehen. Im Computer wird sie aber immer mehr zum Mittel der Poiesis, also der aktiven Konstruk-

tion, Herstellung und Umgestaltung der Welt, etwa in den Ingenieurwissenschaften, der Architektur, der Robotik, den Biowissenschaften etc. Hier zeigt sich eine grundlegende Verschiebung der Bedeutung der Mathematik durch die Digitaltechnologien. Aus einem Werkzeug der Analyse und Erkenntnis wird verstärkt ein Werkzeug der Synthese. Vorstellungen, die sich in der Mathematik als praktisch erwiesen haben, bleiben nicht mehr reine Fiktionen; wir nutzen sie so, als würden sie existieren. Der kartesische Raum, den Vaihinger noch als Fiktion der Mathematik beschreibt, wird durch den Roboterarm zum Fakt, die ehemals nur gedachten Punkte in Raum werden ansteuerbar, so als würden sie existieren.

2. Softwareebenen

Die Hardware von Computern kann nur sogenannte Maschinensprache verarbeiten. Programmierung auf dieser Ebene wäre nicht nur sehr zeitaufwändig, sondern auch sehr fehleranfällig. Sehr große Anwendungen wären kaum zu realisieren, da auf dieser niedrigsten, hardwarenahen Ebene direkt mit den binären Repräsentationen von Zahlen und Befehlen gearbeitet werden muss. Deshalb werden der Nutzerin und dem Nutzer für die Interaktion und die Programmierung der Rechner verschiedene Software-Ebenen angeboten, die auf dem Maschinencode aufsetzen. Die niedrigste Ebene ist die Assemblersprache, bei der noch keine Abstraktion der Prozessorarchitektur erfolgt. Auf den höheren Ebenen, insbesondere bei den höheren Programmiersprachen wird von der Hardware vollständig abstrahiert. Die Software kann so auf allen Rechnern laufen, die einen Interpreter oder Compiler für die jeweilige Sprache anbieten. Die in der Informatik entwickelten unterschiedlichen Programmierparadigmen sind wieder Fiktionen im Sinne Vaihingers. Zwar muss jede Programmiersprache bestimmte Funktionalitäten anbieten, damit die Problemlösungen effizient beschrieben werden können. Doch die unterschiedlichen Paradigmen sollen vor allem das Denken des Programmierers leiten und ihm dabei helfen, seine komplexe Aufgabe zu strukturieren und Denkfehler bei der Programmierung, die zur Laufzeit zum Versagen des Programms führen würden, zu vermeiden. So behauptet etwa die Objektorientierte Programmierung die Architektur der Software am jeweiligen Wirklichkeitsbereich der Anwendung auszurichten. Doch sind dies vor allem Modellierungen, die viel mehr eine neue Wirklichkeit erzeugen als eine gegebene abzubilden, so wie der Mülleimer auf der Schreibtischoberfläche des Rechners nur eine sehr eingeschränkte, funktionale Abstraktion realer Mülleimer darstellt. Reale Mülleimer kann ich anzünden, sie umdrehen und mich

draufstellen, um damit das oberste Buch in einem Regal zu erreichen. Das programmiersprachliche Objekt bleibt eine Fiktion, die nur über ihre vorbestimmte Funktionalität zugänglich ist. Eine ähnliche Rolle erfüllen Ontologien in der Informatik, die für einen bestimmten Gegenstandsbereich die relevante Menge von Begrifflichkeiten und den zwischen ihnen bestehenden Beziehungen sprachlich fassen und formal geordnet darstellen. Auf diese Weise werden künstliche Welten beschrieben, abstrahiert und abgegrenzt. Was nicht Teil der Ontologie ist, existiert nicht.

3. Laufzeitumgebungen

Damit Aufgaben und Problembereiche berechenbar werden, müssen sie formalisiert werden. Formalisierung bedeutet im Wesentlichen Abstraktion von der Fülle realer Verhältnisse. Wir reduzieren die komplexe Einheit des Gegebenen auf wenige Merkmale, von denen wir annehmen, dass durch sie die Problemstellung vollständig oder zumindest ausreichend erfasst wird. Anscheinend Unwesentliches wird beseitigt, Wesentliches benannt und ins Zentrum gerückt. Auch in den Sensoren und Interfaces, die von Algorithmen eingelesen und verarbeitet werden, geschieht diese Abstraktion. Die Ganzheit eines äußeren Geschehens wird in den Sensoren auf quantifizierbare Größen reduziert. Mit Abstraktion wird also das bezeichnet, was an objektiv Vorhandenem abgezogen und an Ideen, also nicht Vorhandenem, addiert werden muss, damit Phänomene unserer Lebenswelt zur Form werden können. Doch dadurch ist nur der Weg der Phänomene in die Maschine beschrieben. Mit der Ausführung der Codes werden die Formalismen nun operativ und erzeugen ihrerseits Phänomene. Der Abstraktionsprozess läuft nun quasi rückwärts ab und wird zur Konkretisierung. Während auf der Ebene der Codes streng formale Operationen ausgeführt werden, sind die Bilder, Klänge, Handlungen an der Oberfläche der Maschinen wieder angereichert mit Unvorhersehbarem, mit Nebeneffekten und Unbeabsichtigtem, das für den Betrachter neue Interpretationsspielräume eröffnet. Die Materialität der Maschine und ihrer Umgebung erzeugen Phänomene, die unvorhersehbar sind. Sofern die visuellen, taktilen, akustischen Erlebnisse, die durch das Programm generiert werden, im Vordergrund einer Anwendung stehen, wie dies im Kunstkontext meist der Fall ist, dann bleibt nur, sich durch iteratives Testen und Modifizieren dem zufriedenstellenden Bild, der überzeugenden Bewegung oder dem ergreifenden Klang anzunähern. Die Frage, ob es sich richtig »anfühlt«, kann nicht auf der Ebene der formalen Codes entschieden werden, obwohl die Wirkung von dort ausgeht.

4. Möglichkeitsräume und Reflexivitäten

Algorithmen sind allgemein, das heißt sie lösen nicht nur eine spezielle Aufgabe, beispielsweise die Frage »Wie viel ist 3 mal 7?«, sondern immer eine Problemklasse, also etwa die Multiplikation. Die Problemklasse kann dabei aus einer sehr großen, im Prinzip unendlichen Menge konkreter Aufgaben bestehen. Ein Algorithmus, der parametergesteuert ein Bild generiert, kann bei Variation seiner Parameter weitere, anders aussehende Bilder erzeugen. Das einzelne Bild ist nur noch ein Stellvertreter aus dem vom Algorithmus aufgespannten Bildraum. Lesen wir den Programmcode als Text, können wir ihn demnach auch als Kompression eines Möglichkeitsraums verstehen. Wenn wir Programmcodes nicht mehr als Problemlösungen begreifen, sondern als kompakte Beschreibungen von Möglichkeitsräumen, ergibt sich ein anderer Blick auf die gesamte Algorithmik. Wir sind nicht mehr an der Korrektheit der Verfahren interessiert, sondern daran, welches fiktive und ästhetische Potenzial sie uns anbieten. Wenn wir spekulativ mit Algorithmen agieren, ohne feste Aufgabe, dann stellen sich auch andere Fragen an die Softwareentwicklung. Die Herausforderung besteht dann darin, Programme zu entwickeln, die interessante Möglichkeitsräume aufspannen, und sich suchend in diesen zu bewegen. Hierfür erweisen sich reflexive Strategien als besonders hilfreich. Mit reflexiv bezeichnen wir Programmstrukturen, bei denen Teile des Ergebnisses benutzt werden, um den Code oder mindestens seine Parameter zu modifizieren. Auf diese Weise erhält die Bewegung durch den jeweiligen Raum von Bildern, Klängen, Bewegungen, Formen etc. eine Richtung. Zu den reflexiven Verfahren, die Prinzipien der Natur nachahmen – Vaihinger würde auch diese Abstraktionen und Übertragungen aus guten Gründen Fiktionen nennen –, gehören unter anderem genetische Algorithmen und neuronale Netze.

Membrane – oder die Veranschaulichung algorithmischer Fiktion im digitalen Bild

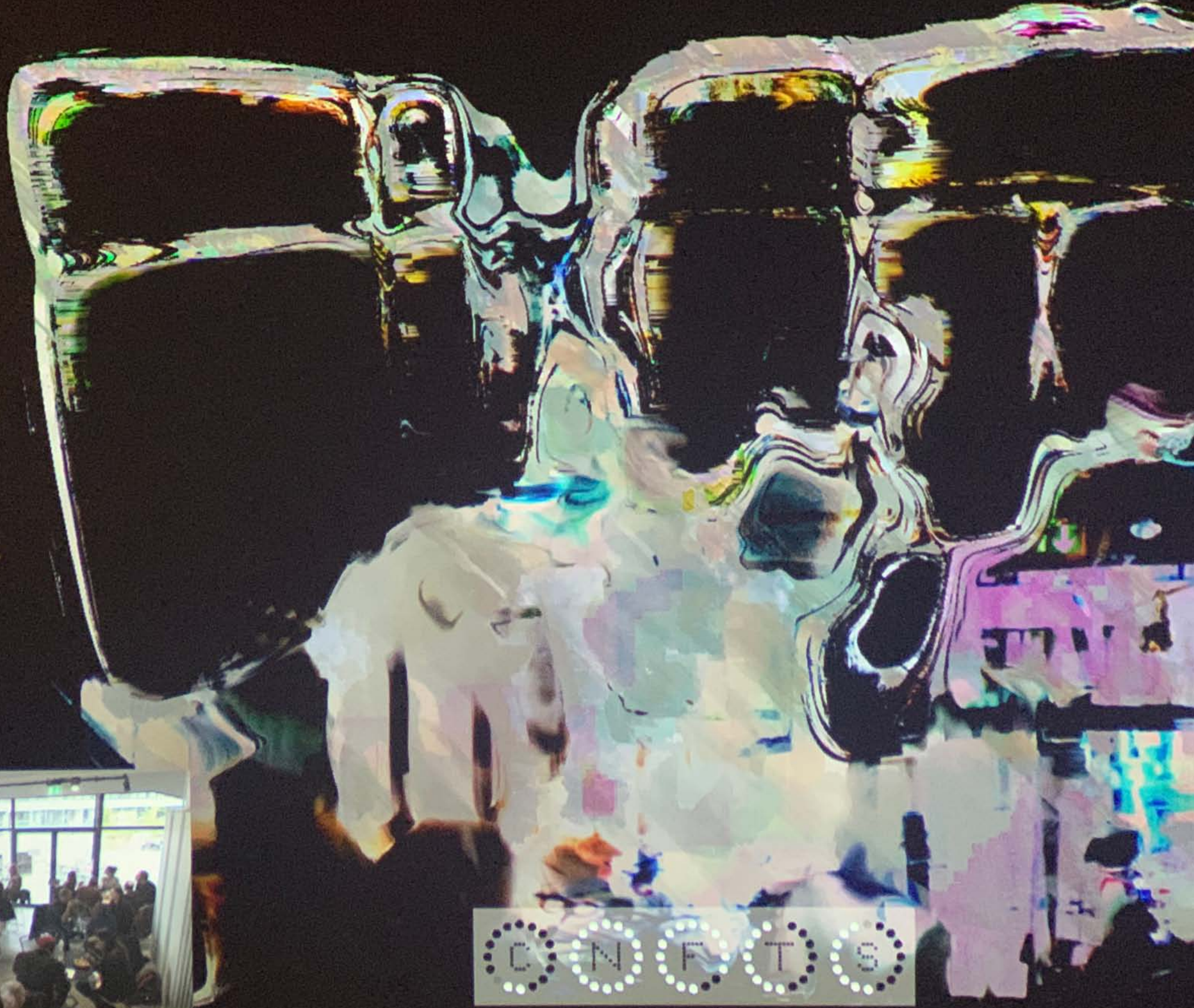
Membrane heißt eine Installation, die für eine Ausstellung in Berlin entwickelt wurde. Sie setzt eine Serie von generativen Videoinstallationen mit Echtzeitvideoinformation der Autorin fort, beispielsweise »598« oder Transits. Membrane soll dem Betrachter ermöglichen, ähnlich wie in *Chromatographic Ballads*, aktiv in die algorithmische Genese des Bildes einzugreifen und das »Imaginieren« des Computers nicht nur zu erleben, sondern auch in Richtung seiner Neugier und Vorliebe zu lenken.

Im Computer werden Bilder als Matrix von rgb-Werten in der x- und y-Achse repräsentiert. Es gehört heute zum Allgemeinwissen, dass das digitale Bild

aus einem Raster von Bildpunkten besteht, die Werte aus einem Spektrum von Farbnuancen annehmen, üblicherweise 256 Werte pro Kanal. Im Falle der Installation Membrane werden die verarbeiteten Bilder von einer statischen Kamera erzeugt, die eine reale Straßenszene in Berlin beobachtet. Insofern können wir das Bild als eine Momentaufnahme des Ortes innerhalb eines zeitlichen Verlaufs betrachten. Die temporären Veränderungen der beobachteten Alltagsituation sind hier eine weitere Dimension, die algorithmischen Experimenten zugänglich gemacht werden kann. Im Allgemeinen leisten die verwendeten Algorithmen – neuronale Netze – eine Komplexitätsreduktion, das heißt, sie wirken wie Filter auf die eingegebenen Bilddaten. Die prinzipielle Offenheit, die Kontingenz des Bildmaterials wird dabei gezielt verringert, im Extremfall wird der Strom an Bildern auf eine Aussage reduziert. Beispielsweise: Ja, es ist ein Hund zu sehen, oder nein, es ist kein Hund zu sehen. Neuronale Netze, eine zentrale Methode maschinellen Lernens, arbeiten dabei mit vernetzten »Sensoren«, welche zusätzlich zu ihren »Wahrnehmungsfähigkeiten« Werte speichern können. Diese sogenannten Neuronen messen und bewerten ein Datenpaket, das durch sie hindurch geschleust wird. Das folgende Datenpaket wird dann aufgrund des zuvor gelesenen bewertet und weitergeleitet und passt sich gleichzeitig den gelesenen Daten an. Hier liegt ein wechselseitiges Feedback vor, das völlig ohne äußere Kategorien operiert. Ein Neuron ist in seiner Wirkung völlig von der Architektur der Vernetzung zu anderen Neuronen abhängig. Die Gesamtheit der Neuronen erzeugt abstrakte, netzwerkartige Lernumgebungen, welche in der Lage sind, Eigenschaften von Datensets zu reproduzieren und auf Ähnlichkeit oder Unterscheidung zu bewerten. In der Wahrnehmung des Menschen operieren solche Verfahren subsymbolisch und bedürfen nicht der Vorabinterpretation durch Labels, Metaphern oder Symbole.

Frühere Experimente im gleichen Kontext (<http://ursuladamm.de/transits-2012/>) haben jeden Pixel der Bildmatrix eines solchen Videostreams zur operativen Einheit gemacht. Ein Pixel lernt also über die Laufzeit des Programms unterschiedliche Farbwerte und gibt als Endresultat eine Farbe aus, die die Summe von Farben über eine bestimmte Laufzeit der Kamera darstellt. So kreiert diese banale Form der »Erinnerung« von Farbwerten im Bild etwas völlig Neues: eine Aufzeichnung von Bewegungsspuren, welche gleichzeitig Eigenschaften eines Ortes in der Zeit beschreiben. Hier wird offensichtlich, wie die Anordnung von Input, Verarbeitung und Output bereits neue Welten und Zusammenhänge schaffen kann.

Auf technischer Ebene werden bei Membrane nicht nur Pixel oder »harte« Bildausschnitte kontrolliert, sondern Bild-»Features«, die beispielsweise



C N F T S

Abb. 1 Screenprint von Membrane in der Ausstellung »Entangled Realities« im HEK Basel, 2019. Das Bild besteht aus dem Live-Zuspieler aus dem Restaurant des Museums, der durch den Algorithmus veränderten Bildinformation und fünf Reglern, die die jeweilige Einstellung zeigen. Die Regler ändern die Werte einer »Shader [C]rossbar«, der »[N]umber of Shaders«, »[F]eedback«, »[T]ime« und »[S]pace«.



Abb. 2 Installationsaufbau von Membrane in der Ausstellung »Entangled Realities« im HEK Basel, 2019; Bedienpult der Installation, Projektion des Bildes der Aussenkamera, Innenkamera auf Stativ, Projektion des Bildes der Innenkamera.



Abb. 3 Membrane angewandt auf Footage vom Times Square New York, Screenprint 2018

unter Berücksichtigung von Farbe lernen, erinnern und sich neu zusammensetzen lassen: Wir wählen Merkmale, deren Auswahl wir einem Algorithmus überlassen. Das sogenannte TGAN realisiert die Funktion des *Unsupervised Learnings* durch das gegenläufige, rückgekoppelte Wirken zweier Teilnetze: des Bildmodelle produzierenden Generators und eines bewertenden Diskriminators. Dieser Algorithmus ist speziell entwickelt worden, um Repräsentationen von un kategorisierten Videodaten zu erstellen und mithilfe der Repräsentationen neue Bildsequenzen zu erzeugen. (Temporal Generative Adversarial Nets, <https://pfnet-research.github.io/tgan/>, <https://arxiv.org/abs/1611.06624>). Wir erweitern den TGAN-Algorithmus durch die Zwischenschaltung einer Wavelet-Analyse, die es erlaubt, vom Start weg mit Bildfeatures statt mit bloßen Pixeln zu operieren und zu interagieren. Mehr noch als klassisches Machine Learning »erfindet« unser Algorithmus also eigene Bilder. Der Algorithmus spekuliert quasi auf der Basis des Gelernten und entwickelt eine eigenständige Zeitlichkeit. Dies geschieht aber nicht ohne Kontrolle: Die Kontrolle besteht, neben der Wahl von Feature-Klassen, in einem frei zu wählenden Datenset von Videos, welche als »Zielvorgabe« gesehen werden können. In unserem Fall möchten wir Videos aus verschiedenen Städten weltweit zur »Diskriminierung« des erfundenen Videos heranziehen. Es geht uns dabei nicht um eine Anpassung von Berliner Stadtansichten an globale Urbanität, sondern ums Ausloten von Spezifika, um das Erfinden durch Assoziation. Diese Assoziation kann im Algorithmus verortet, variiert und manipuliert werden im sogenannten Referenzdatenset.

Unser erweiterter TGAN-Algorithmus wird darüber hinaus also vielfältige Möglichkeiten generieren, um durch Feedback-Loops, also zusätzliche Rückkopplungen über die einzelnen Layer hinweg, dynamisches »Lernen« sowohl auf kurzen als auch auf längeren Zeitskalen zu bewirken, durch den Live-Videooutput implizit zu visualisieren und letztlich durch die Benutzerin oder den Benutzer steuerbar zu machen. Die Installation vor Ort erlaubt, alle Schritte vom unveränderten Bild bis hin zur völligen Abstraktion auf der Ebene eines Live Video-Inputs zu wählen.

Der Installation liegt die Frage zugrunde, zugunsten welcher Ansichten wir – auch als Besucherin oder Besucher einer Ausstellung – bereit sind, realistische Darstellungen zu verfälschen. Und welcher Aspekte des Gesehenen wir verlustig gehen. Wohin steuern wir mit solchen Verschiebungen der Bildinhalte – was wird das wahrzunehmende Erlebnis im Zentrum des künstlerischen Ausdrucks?

Das fiktionale Potenzial von Machine Learning wurde zuletzt durch Googles Deep-Dream-Algorithmen bekannt. Trainierte Netze synthetisieren Bilder

aus Begrifflichkeiten, sie zitieren quasi Symbole, die sie in einem Bild zu erkennen glauben. Aus ästhetischer Sicht wirken die entstandenen Bilder paranoid: Anstatt eine zusammenhängende Aussage zu vermitteln, verlieren sie sich immer wieder in formale Details und reproduzieren Vergangenes. Aus künstlerischer Sicht stellt sich nun die Frage, wie mit Algorithmen etwas Originelles, Neues geschaffen werden kann. Diese Fragestellung liegt der Softwarekonstruktion von Membrane zugrunde. Anders als Googles *DeepDream*-Algorithmen und -Bilder wollen wir nichts Konkretes in den Straßenszenen erkennen und identifizieren, sondern interessieren uns dafür, wie Menschen die Szenen wahrnehmen. Deshalb lassen wir unsere Maschinen auf kleinere, formale Bildelemente und -merkmale schauen, deren Eigenwertigkeit wir erkennen und verstärken wollen. Wir möchten die Besucherinnen und Besucher bewusst offen gehaltenen Bildartefakten, etwa Kanten, Linien, Farbe, geometrischen Grundformen, Bewegungen, aussetzen, die anstelle der Imitation menschlichen Sehens und Verstehens die maschinellen Seh- und Interpretationsweisen offenlegen. Interessanterweise ähneln diese Bilder Entwicklungen der klassischen Moderne, beispielsweise fortschreitende Abstraktion auf der Ebene formaler Aspekte, und wiederholen Stile wie Impressionismus, Pointillismus, Kubismus, Tachismus auf ihre ganz eigene unintentionale Art. Diese Stile fragmentierten das Wahrgenommene im Zuge der bildlichen Umsetzung in einzelne sensorische Eindrücke. Bildgegenstände sind jetzt Merkmale von vormaligen Gegenständen und verlieren sukzessive völlig den Bezug zur Realität. Gleichzeitig hinterfragen wir, ob diese Fragmentierung der Wahrnehmung nicht willkürlich verläuft und möglicherweise noch andere Konzepte der Abstraktion und Bildlichkeit für uns bedeutsam werden können.

Betrachten wir Membrane etwas gezielter vor dem Hintergrund des weiter oben eingeführten Versuchs einer Typologie fiktionaler Strategien im Zusammenhang mit Algorithmen, dann zeigen sich alle vier Strategien. In der Installation Membrane finden wir die mathematischen Fiktionen genauso wie die Softwarefiktionen und die Fiktionen der Laufzeitumgebungen. Den vierten Bereich, die Möglichkeitsräume und Reflexivitäten, wollen wir abschließend noch etwas genauer ansehen. Wichtig ist festzustellen, dass die in der Informatik entwickelten Algorithmen hier nicht als Analysewerkzeug eingesetzt werden, sondern zur Synthese. Es geht also nicht um das Erkennen eines Bildinhaltes, sondern um das Erstellen, im vorliegenden Fall insbesondere von Bewegungsbildmustern. Die Software soll auf Neigungen der Betrachter in einem an sich offenen Feld von Entwicklungsmöglichkeiten reagieren können. Wie können wir das bewerkstelligen?

In dem 2002 erschienenen Buch »A New Kind of Science« macht Stephen Wolfram am Beispiel einfacher zellulärer Automaten deutlich, wie bereits ganz einfache Regeln und Programme große Komplexität erzeugen können. Anhand eindimensionaler Automaten zeigt er, dass sich qualitativ unterschiedliche Klassen von Verhaltensweisen finden lassen und bereits sehr einfache Regeln die Mächtigkeit universeller Kalküle besitzen. Wolfram untersucht alle 256 möglichen eindimensionalen, binären Automaten. Schon für zweidimensionale Automaten lässt sich diese Analyse nicht mehr umfassend durchführen. Betrachten wir das Einzelbild der vorliegenden Installation beispielsweise als zweidimensionalen Automaten, wobei sich der Zustand jedes Bildpunktes aus jeweils drei 256-wertigen Zahlen zusammensetzt und gleichzeitig die Nachbarschaft der in eine Verarbeitungsregel einbezogenen Punkte offen bleibt, ergibt sich ein analytisch nicht mehr handhabbarer Möglichkeitsraum für die Bildgenerierung. Rückkopplungen fügen eine weitere Komplexitätsstufe hinzu. Die Frage lautet also: Wie gehen wir mit solchen extrem großen Möglichkeitsräumen um? Die Antwort sollte nun nicht mehr überraschen: Wir lassen uns bei der Programmentwicklung von Fiktionen leiten. Wir wissen, dass unser Gehirn anders funktioniert als künstliche neuronale Netze. Schon die Form der Codierung unterscheidet sich grundlegend: Frequenz-Codierung beim menschlichen Gehirn versus Amplituden-Codierung bei künstlichen neuronalen Netzen. Auch wenn wir von Wahrnehmungsfähigkeiten der Netze sprechen, von Lernen, Neuronen, Erinnerung und Ähnlichkeit, dann handelt es sich um kaum mehr als Metaphern. Im Grunde ließen sich die Verfahren auch komplett in der Sprache der Mathematik beschreiben. Denn obwohl Wolframs Zellularautomaten ohne jegliche Reflexivität auf ein Außen entstehen und in unserem Beispiel reflexive Strategien für eine Anbindung des Zustands der Maschine an einen Zustand der Realität sorgen sollen, bleiben auch unsere Programme Rechenoperationen im mathematischen Möglichkeitsraum. Doch das softwareseitige Konstrukt der neuronalen Netze bietet eine Vorstellung an, die uns offensichtlich dabei hilft, den unüberschaubaren Möglichkeitsraum unserer (digitalen) Bilder nicht nur anschaulicher, sondern auch gezielter anzugehen und die Ergebnisse im Hinblick auf das Modell zu interpretieren, obwohl die Realität, von der wir hier sprechen, konstruiert bleibt. Sobald wir nicht das Erkennen und Analysieren, sondern das Synthesepotenzial von Algorithmen in den Mittelpunkt stellen, sind Verfahrensklassen wie neuronale Netze, genetische Algorithmen und viele andere Methoden kaum mehr als hilfreiche Metaphern, die es uns erlauben, mit dem Überschuss an regelbasierten Handlungsmöglichkeiten umzugehen.

Aus kultureller Sicht bleiben zwei große Fragen: Wie sind in diesen künstlichen Handlungsräumen ästhetische Entscheidungen zu treffen? Kann die Blickverschiebung von der Analyse hin zur Synthese und Fiktion dabei helfen, dass wir auch unsere analytischen Verfahren anders bewerten und sie als spezielle, normative Formen unserer gesellschaftlichen Fiktion erkennen, welche vor allem einer Selbstverstärkung vorhandener Strukturen dienen? So kann ein künstlerisch unbefangenes Navigieren im Überschuss regelbasierter Handlungsmöglichkeiten zum Garant für Unbekanntes und Neues werden.

Die Arbeit wurde produziert mit der freundlichen Unterstützung des Kreativfonds der Bauhaus Universität Weimar und des Hauses der elektronischen Künste (HEK) Basel.

Software: Peter Serocka
Sound: Teresa Carasco
Konzept: Ursula Damm

PROF. URSULA DAMM ist Leiterin der Professur Gestaltung Medialer Umgebungen an der Fakultät Kunst und Gestaltung.

GEORG TROGEMANN ist Professor für Experimentelle Informatik an der Kunsthochschule für Medien Köln und forscht zu alternativen Sicht- und Handlungsweisen in der Technik.