Driven by Compression Progress: A Simple Principle Explains Essential Aspects of Subjective Beauty, Novelty, Surprise, Interestingness, Attention, Curiosity, Creativity, Art, Science, Music, Jokes

Jürgen Schmidhuber
TU Munich, Boltzmannstr. 3, 85748 Garching bei München, Germany &
IDSIA, Galleria 2, 6928 Manno (Lugano), Switzerland
juergen@idsia.ch - http://www.idsia.ch/∼juergen

Abstract

I argue that data becomes temporarily interesting by itself to some self-improving, but computationally limited, subjective observer once he learns to predict or compress the data in a better way, thus making it subjectively simpler and more *beautiful.* Curiosity is the desire to create or discover more non-random, non-arbitrary, regular data that is novel and *surprising* not in the traditional sense of Boltzmann and Shannon but in the sense that it allows for compression progress because its regularity was not yet known. This drive maximizes *interestingness*, the first derivative of subjective beauty or compressibility, that is, the steepness of the learning curve. It motivates exploring infants, pure mathematicians, composers, artists, dancers, comedians, yourself, and (since 1990) artificial systems.

First version of this preprint published 23 Dec 2008; revised 15 April 2009. Short version: [91]. Long version: [90]. We distill some of the essential ideas in earlier work (1990-2008) on this subject: [57, 58, 61, 59, 60, 108, 68, 72, 76] and especially recent papers [81, 87, 88, 89].

Dagstuhl Seminar Proceedings 09291
Computational Creativity : An Interdisciplinary Approach
http://drops.dagstuhl.de/opus/volltexte/2009/2197

Contents

1 Store & Compress & Reward Compression Progress

If the history of the entire universe were computable [123, 124], and there is no evidence against this possibility [84], then its simplest explanation would be the shortest program that computes it [65, 70]. Unfortunately there is no general way of finding the shortest program computing any given data [34, 106, 107, 37]. Therefore physicists have traditionally proceeded incrementally, analyzing just a small aspect of the world at any given time, trying to find simple laws that allow for describing their limited observations better than the best previously known law, essentially trying to find a program that compresses the observed data better than the best previously known program. For example, Newton's law of gravity can be formulated as a short piece of code which allows for substantially compressing many observation sequences involving falling apples and other objects. Although its predictive power is limited—for example, it does not explain quantum fluctuations of apple atoms—it still allows for greatly reducing the number of bits required to encode the data stream, by assigning short codes to events that are predictable with high probability [28] under the assumption that the law holds. Einstein's general relativity theory yields additional compression progress as it compactly explains many previously unexplained deviations from Newton's predictions.

Most physicists believe there is still room for further advances. Physicists, however, are not the only ones with a desire to improve the subjective compressibility of their observations. Since short and simple explanations of the past usually reflect some repetitive regularity that helps to predict the future as well, *every* intelligent system interested in achieving future goals should be motivated to compress the history of raw sensory inputs in response to its actions, simply to improve its ability to plan ahead.

A long time ago, Piaget [49] already explained the explorative learning behavior of children through his concepts of assimilation (new inputs are embedded in old schemas—this may be viewed as a type of compression) and accommodation (adapting an old schema to a new input—this may be viewed as a type of compression improvement), but his informal ideas did not provide enough formal details to permit computer implementations of his concepts. How to model a compression progress drive in artificial systems? Consider an active agent interacting with an initially unknown world. We may use our general Reinforcement Learning (RL) framework of artificial curiosity

(1990-2008) [57, 58, 61, 59, 60, 108, 68, 72, 76, 81, 88, 87, 89] to make the agent discover data that allows for additional compression progress and improved predictability. The framework directs the agent towards a better understanding the world through active exploration, even when external reward is rare or absent, through *intrinsic reward* or *curiosity reward* for actions leading to discoveries of previously unknown regularities in the action-dependent incoming data stream.

1.1 Outline

Section 1.2 will informally describe our algorithmic framework based on: (1) a continually improving predictor or compressor of the continually growing data history, (2) a computable measure of the compressor's progress (to calculate intrinsic rewards), (3) a reward optimizer or reinforcement learner translating rewards into action sequences expected to maximize future reward. The formal details are left to the Appendix, which will elaborate on the underlying theoretical concepts and describe discrete time implementations. Section 1.3 will discuss the relation to external reward (external in the sense of: originating outside of the brain which is controlling the actions of its "external" body). Section 2 will informally show that many essential ingredients of intelligence and cognition can be viewed as natural consequences of our framework, for example, detection of novelty & surprise & interestingness, unsupervised shifts of attention, subjective perception of beauty, curiosity, creativity, art, science, music, and jokes. In particular, we reject the traditional Boltzmann / Shannon notion of surprise, and demonstrate that both science and art can be regarded as by-products of the desire to create / discover more data that is compressible in hitherto unknown ways. Section 3 will give an overview of previous concrete implementations of approximations of our framework. Section 4 will apply the theory to images tailored to human observers, illustrating the rewarding learning process leading from less to more subjective compressibility. Section 5 will outline how to improve our previous implementations, and how to further test predictions of our theory in psychology and neuroscience.

1.2 Algorithmic Framework

The basic ideas are embodied by the following set of simple algorithmic principles distilling some of the essential ideas in previous publications on this topic [57, 58, 61, 59, 60, 108, 68, 72, 76, 81, 88, 87, 89]. As mentioned above, formal details are left to the Appendix. As discussed in Section 2, the principles at least qualitatively explain many aspects of intelligent agents such as humans. This encourages us to implement and evaluate them in cognitive robots and other artificial systems.

1. **Store everything.** During interaction with the world, store the entire raw history of actions and sensory observations including reward signals—the data is *holy* as it is the only basis of all that can be known about the world. To see that full data storage is not unrealistic: A human lifetime rarely lasts much longer than 3×10^9 seconds. The human brain has roughly 10^{10} neurons, each with 10^4 synapses on average. Assuming that only half of the brain's capacity is used for storing raw data, and that each synapse can store at most 6 bits, there is still enough capacity

to encode the lifelong sensory input stream with a rate of roughly 10^5 bits/s, comparable to the demands of a movie with reasonable resolution. The storage capacity of affordable technical systems will soon exceed this value. If you can store the data, do not throw it away!

2. **Improve subjective compressibility.** In principle, any regularity in the data history can be used to compress it. The compressed version of the data can be viewed as its simplifying explanation. Thus, to better explain the world, spend some of the computation time on an adaptive compression algorithm trying to partially compress the data. For example, an adaptive neural network [8] may be able to learn to predict or postdict some of the historic data from other historic data, thus incrementally reducing the number of bits required to encode the whole. See Appendix A.3 and A.5.

3. **Let intrinsic curiosity reward reflect compression progress.** The agent should monitor the improvements of the adaptive data compressor: whenever it learns to reduce the number of bits required to encode the historic data, generate an intrinsic reward signal or curiosity reward signal in proportion to the learning progress or compression progress, that is, the number of saved bits. See Appendix A.5 and A.6.

4. **Maximize intrinsic curiosity reward** [57, 58, 61, 59, 60, 108, 68, 72, 76, 81, 88, 87]. Let the action selector or controller use a general Reinforcement Learning (RL) algorithm (which should be able to observe the current state of the adaptive compressor) to maximize expected reward, including intrinsic curiosity reward. To optimize the latter, a good RL algorithm will select actions that focus the agent's attention and learning capabilities on those aspects of the world that allow for finding or creating new, previously unknown but learnable regularities. In other words, it will try to maximize the steepness of the compressor's learning curve. This type of *active unsupervised learning* can help to figure out how the world works. See Appendix A.7, A.8, A.9, A.10.

The framework above essentially specifies the objectives of a curious or creative system, not the way of achieving the objectives through the choice of a particular adaptive compressor or predictor and a particular RL algorithm. Some of the possible choices leading to special instances of the framework (including previous concrete implementations) will be discussed later.

1.3 Relation to External Reward

Of course, the real goal of many cognitive systems is not just to satisfy their curiosity, but to solve externally given problems. Any formalizable problem can be phrased as an RL problem for an agent living in a possibly unknown environment, trying to maximize the future external reward expected until the end of its possibly finite lifetime. The new millennium brought a few extremely general, even universal RL algorithms (universal problem solvers or universal artificial intelligences—see Appendix A.8, A.9) that are optimal in various theoretical but not necessarily practical senses, e. g., [29, 79, 82,

83, 86, 85, 92]. To the extent that learning progress / compression progress / curiosity as above are helpful, these universal methods will automatically discover and exploit such concepts. Then why bother at all writing down an explicit framework for active curiosity-based experimentation?

One answer is that the present universal approaches sweep under the carpet certain problem-independent constant slowdowns, by burying them in the asymptotic notation of theoretical computer science. They leave open an essential remaining question: If the agent can execute only a fixed number of computational instructions per unit time interval (say, 10 trillion elementary operations per second), what is the best way of using them to get as close as possible to the recent theoretical limits of universal AIs, especially when external rewards are very rare, as is the case in many realistic environments? The premise of this paper is that the curiosity drive is such a general and generally useful concept for limited-resource RL in rare-reward environments that it should be prewired, as opposed to be learnt from scratch, to save on (constant but possibly still huge) computation time. An inherent assumption of this approach is that in realistic worlds a better explanation of the past can only help to better predict the future, and to accelerate the search for solutions to externally given tasks, ignoring the possibility that curiosity may actually be harmful and "kill the cat."

2 Consequences of the Compression Progress Drive

Let us discuss how many essential ingredients of intelligence and cognition can be viewed as natural by-products of the principles above.

2.1 Compact Internal Representations or Symbols as By-Products of Efficient History Compression

To compress the history of observations so far, the compressor (say, a predictive neural network) will automatically create internal representations or *symbols* (for example, patterns across certain neural feature detectors) for things that frequently repeat themselves. Even when there is limited predictability, efficient compression can still be achieved by assigning short codes to events that are predictable with high probability [28, 95]. For example, the sun goes up every day. Hence it is efficient to create internal symbols such as *daylight* to describe this repetitive aspect of the data history by a short reusable piece of internal code, instead of storing just the raw data. In fact, predictive neural networks are often observed to create such internal (and hierarchical) codes as a by-product of minimizing their prediction error on the training data.

2.2 Consciousness as a Particular By-Product of Compression

There is one thing that is involved in all actions and sensory inputs of the agent, namely, the agent itself. To efficiently encode the entire data history, it will profit from creating some sort of internal *symbol* or code (e. g., a neural activity pattern) representing the agent itself. Whenever this representation is actively used, say, by activating the

corresponding neurons through new incoming sensory inputs or otherwise, the agent could be called *self-aware* or *conscious.*

This straight-forward explanation apparently does not abandon any essential aspects of our intuitive concept of consciousness, yet seems substantially simpler than other recent views [1, 2, 105, 101, 25, 12]. In the rest of this paper we will not have to attach any particular mystic value to the notion of consciousness—in our view, it is just a natural by-product of the agent's ongoing process of problem solving and world modeling through data compression, and will not play a prominent role in the remainder of this paper.

2.3 The Lazy Brain's Subjective, Time-Dependent Sense of Beauty

Let $O(t)$ denote the state of some subjective observer O at time t. According to our *lazy brain theory* [67, 66, 69, 81, 87, 88], we may identify the subjective beauty $B(D, O(t))$ of a new observation D (but not its interestingness - see Section 2.4) as being proportional to the number of bits required to encode D, given the observer's limited previous knowledge embodied by the current state of its adaptive compressor. For example, to efficiently encode previously viewed human faces, a compressor such as a neural network may find it useful to generate the internal representation of a prototype face. To encode a new face, it must only encode the deviations from the prototype [67]. Thus a new face that does not deviate much from the prototype [17, 48] will be subjectively more beautiful than others. Similarly for faces that exhibit geometric regularities such as symmetries or simple proportions [69, 88]—in principle, the compressor may exploit any regularity for reducing the number of bits required to store the data.

Generally speaking, among several sub-patterns classified as *comparable* by a given observer, the subjectively most beautiful is the one with the simplest (shortest) description, given the observer's current particular method for encoding and memorizing it [67, 69]. For example, mathematicians find beauty in a simple proof with a short description in the formal language they are using. Others like geometrically simple, aesthetically pleasing, low-complexity drawings of various objects [67, 69].

This immediately explains why many human observers prefer faces similar to their own. What they see every day in the mirror will influence their subjective prototype face, for simple reasons of coding efficiency.

2.4 Subjective Interestingness as First Derivative of Subjective Beauty: The Steepness of the Learning Curve

What's beautiful is not necessarily interesting. A beautiful thing is interesting only as long as it is new, that is, as long as the algorithmic regularity that makes it simple has not yet been fully assimilated by the adaptive observer who is still learning to compress the data better. It makes sense to define the time-dependent subjective *Interestingness* $I(D, O(t))$ of data D relative to observer O at time t by

$$I(D, O(t)) \sim \frac{\partial B(D, O(t))}{\partial t}, \tag{1}$$

the *first derivative* of subjective beauty: as the learning agent improves its compression algorithm, formerly apparently random data parts become subjectively more regular and beautiful, requiring fewer and fewer bits for their encoding. As long as this process is not over the data remains interesting and rewarding. The Appendix and Section 3 on previous implementations will describe details of discrete time versions of this concept. See also [59, 60, 108, 68, 72, 76, 81, 88, 87].

2.5 Pristine Beauty & Interestingness vs External Rewards

Note that our above concepts of beauty and interestingness are limited and *pristine* in the sense that they are *not a priori* related to pleasure derived from external rewards (compare Section 1.3). For example, some might claim that a hot bath on a cold day triggers "beautiful" feelings due to rewards for achieving prewired target values of external temperature sensors (external in the sense of: outside the brain which is controlling the actions of its external body). Or a song may be called "beautiful" for emotional (e.g., [13]) reasons by some who associate it with memories of external pleasure through their first kiss. Obviously this is not what we have in mind here—we are focusing solely on rewards of the intrinsic type based on learning progress.

2.6 True Novelty & Surprise vs Traditional Information Theory

Consider two extreme examples of uninteresting, unsurprising, boring data: A vision-based agent that always stays in the dark will experience an extremely compressible, soon totally predictable history of unchanging visual inputs. In front of a screen full of white noise conveying a lot of information and "novelty" and "surprise" in the traditional sense of Boltzmann and Shannon [102], however, it will experience highly unpredictable and fundamentally incompressible data. In both cases the data is boring [72, 88] as it does not allow for further compression progress. Therefore we reject the traditional notion of surprise. Neither the arbitrary nor the fully predictable is *truly* novel or surprising—only data with still *unknown* algorithmic regularities are [57, 58, 61, 59, 60, 108, 68, 72, 76, 81, 88, 87, 89]!

2.7 Attention / Curiosity / Active Experimentation

In absence of external reward, or when there is no known way to further increase the expected external reward, our controller essentially tries to maximize *true novelty* or *interestingness*, the *first derivative* of subjective beauty or compressibility, the steepness of the learning curve. It will do its best to select action sequences expected to create observations yielding maximal expected future compression *progress*, given the limitations of both the compressor and the compressor improvement algorithm. It will learn to focus its attention [96, 116] and its actively chosen experiments on things that are currently still incompressible but are expected to become compressible / predictable through additional learning. It will get bored by things that already are subjectively compressible. It will also get bored by things that are currently incompressible but will apparently remain so, given the experience so far, or where the costs

of making them compressible exceed those of making other things compressible, etc. [57, 58, 61, 59, 60, 108, 68, 72, 76, 81, 88, 87, 89].

2.8 Discoveries

An unusually large compression breakthrough deserves the name *discovery*. For example, as mentioned in the introduction, the simple law of gravity can be described by a very short piece of code, yet it allows for greatly compressing all previous observations of falling apples and other objects.

2.9 Beyond Standard Unsupervised Learning

Traditional unsupervised learning is about finding regularities, by clustering the data, or encoding it through a factorial code [4, 64] with statistically independent components, or predicting parts of it from other parts. All of this may be viewed as special cases of data compression. For example, where there are clusters, a data point can be efficiently encoded by its cluster center plus relatively few bits for the deviation from the center. Where there is data redundancy, a non-redundant factorial code [64] will be more compact than the raw data. Where there is predictability, compression can be achieved by assigning short codes to those parts of the observations that are predictable from previous observations with high probability [28, 95]. Generally speaking we may say that a major goal of traditional unsupervised learning is to improve the compression of the observed data, by discovering a program that computes and thus explains the history (and hopefully does so quickly) but is clearly shorter than the shortest previously known program of this kind.

Traditional unsupervised learning is not enough though—it just analyzes and encodes the data but does not choose it. We have to extend it along the dimension of active action selection, since our unsupervised learner must also choose the actions that influence the observed data, just like a scientist chooses his experiments, a baby its toys, an artist his colors, a dancer his moves, or any attentive system [96] its next sensory input. That's precisely what is achieved by our RL-based framework for curiosity and creativity.

2.10 Art & Music as By-Products of the Compression Progress Drive

Works of art and music may have important purposes beyond their social aspects [3] despite of those who classify art as superfluous [50]. Good observer-dependent art deepens the observer's insights about this world or possible worlds, unveiling previously unknown regularities in compressible data, connecting previously disconnected patterns in an initially surprising way that makes the combination of these patterns subjectively more compressible (art as an eye-opener), and eventually becomes known and less interesting. I postulate that the active creation and attentive perception of all kinds of artwork are just by-products of our principle of interestingness and curiosity yielding reward for compressor improvements.

Let us elaborate on this idea in more detail, following the discussion in [81, 88]. Artificial or human observers must perceive art sequentially, and typically also actively, e.g., through a sequence of attention-shifting eye saccades or camera movements scanning a sculpture, or internal shifts of attention that filter and emphasize sounds made by a pianist, while surpressing background noise. Undoubtedly many derive pleasure and rewards from perceiving works of art, such as certain paintings, or songs. But different subjective observers with different sensory apparati and compressor improvement algorithms will prefer different input sequences. Hence any objective theory of what is good art must take the subjective observer as a parameter, to answer questions such as: Which sequences of actions and resulting shifts of attention should he execute to maximize his pleasure? According to our principle he should select one that maximizes the quickly learnable compressibility that is new, relative to his current knowledge and his (usually limited) way of incorporating / learning / compressing new data.

2.11 Music

For example, which song should some human observer select next? Not the one he just heard ten times in a row. It became too predictable in the process. But also not the new weird one with the completely unfamiliar rhythm and tonality. It seems too irregular and contain too much arbitrariness and subjective noise. He should try a song that is unfamiliar enough to contain somewhat unexpected harmonies or melodies or beats etc., but familiar enough to allow for quickly recognizing the presence of a new learnable regularity or compressibility in the sound stream. Sure, this song will get boring over time, but not yet.

The observer dependence is illustrated by the fact that Schönberg's twelve tone music is less popular than certain pop music tunes, presumably because its algorithmic structure is less obvious to many human observers as it is based on more complicated harmonies. For example, frequency ratios of successive notes in twelve tone music often cannot be expressed as fractions of very small integers. Those with a prior education about the basic concepts and objectives and constraints of twelve tone music, however, tend to appreciate Schönberg more than those without such an education.

All of this perfectly fits our principle: The learning algorithm of the compressor of a given subjective observer tries to better compress his history of acoustic and other inputs where possible. The action selector tries to find history-influencing actions that help to improve the compressor's performance on the history so far. The interesting musical and other subsequences are those with previously unknown yet learnable types of regularities, because they lead to compressor improvements. The boring patterns are those that seem arbitrary or random, or whose structure seems too hard to understand.

2.12 Paintings, Sculpture, Dance, Film etc.

Similar statements not only hold for other dynamic art including film and dance (taking into account the compressibility of controller actions), but also for painting and sculpture, which cause dynamic pattern sequences due to attention-shifting actions [96, 116] of the observer.

2.13 No Objective "Ideal Ratio" Between Expected and Unexpected

Some of the previous attempts at explaining aesthetic experiences in the context of information theory [7, 41, 6, 44] emphasized the idea of an *"ideal"* ratio between expected and unexpected information conveyed by some aesthetic object (its *"order"* vs its *"complexity"*). Note that our alternative approach does not have to postulate an objective ideal ratio of this kind. Instead our dynamic measure of interestingness reflects the *change* in the number of bits required to encode an object, and explicitly takes into account the subjective observer's prior knowledge as well as the limitations of its compression improvement algorithm.

2.14 Blurred Boundary Between *Active* Creative Artists and *Passive* Perceivers of Art

Just as observers get intrinsic rewards for sequentially focusing attention on artwork that exhibits new, previously unknown regularities, the *creative* artists get reward for making it. For example, I found it extremely rewarding to discover (after hundreds of frustrating failed attempts) the simple geometric regularities that permitted the construction of the drawings in Figures 1 and 2. The distinction between artists and observers is blurred though. Both execute action sequences to exhibit new types of compressibility. The intrinsic motivations of both are fully compatible with our simple principle.

Some artists, of course, crave *external* reward from other observers, in form of praise, money, or both, in addition to the *intrinsic* compression improvement-based reward that comes from creating a truly novel work of art. Our principle, however, conceptually separates these two reward types.

2.15 How Artists and Scientists are Alike

From our perspective, scientists are very much like artists. They actively select experiments in search for simple but new laws compressing the resulting observation history. In particular, the *creativity* of painters, dancers, musicians, pure mathematicians, physicists, can be viewed as a mere by-product of our curiosity framework based on the compression progress drive. All of them try to create new but non-random, non-arbitrary data with surprising, previously unknown regularities. For example, many physicists invent experiments to create data governed by previously unknown laws allowing to further compress the data. On the other hand, many artists combine well-known objects in a subjectively novel way such that the observer's subjective description of the result is shorter than the sum of the lengths of the descriptions of the parts, due to some previously unnoticed regularity shared by the parts.

What is the main difference between science and art? The essence of science is to *formally nail down* the nature of compression progress achieved through the discovery of a new regularity. For example, the law of gravity can be described by just a few symbols. In the fine arts, however, compression progress achieved by observing an artwork combining previously disconnected things in a new way (art as an eye-opener) may be *sub*conscious and not at all formally describable by the observer, who may *feel*

the progress in terms of intrinsic reward without being able to say exactly which of his memories became more subjectively compressible in the process.

The framework in the appendix is sufficiently formal to allow for implementation of our principle on computers. The resulting artificial observers will vary in terms of the computational power of their history compressors and learning algorithms. This will influence what is good art / science to them, and what they find interesting.

2.16 Jokes and Other Sources of Fun

Just like other entertainers and artists, comedians also tend to combine well-known concepts in a novel way such that the observer's subjective description of the result is shorter than the sum of the lengths of the descriptions of the parts, due to some previously unnoticed regularity shared by the parts.

In many ways the laughs provoked by witty jokes are similar to those provoked by the acquisition of new skills through both babies and adults. Past the age of 25 I learnt to juggle three balls. It was not a sudden process but an incremental and rewarding one: in the beginning I managed to juggle them for maybe one second before they fell down, then two seconds, four seconds, etc., until I was able to do it right. Watching myself in the mirror (as recommended by juggling teachers) I noticed an idiotic grin across my face whenever I made progress. Later my little daughter grinned just like that when she was able to stand on her own feet for the first time. All of this makes perfect sense within our algorithmic framework: such grins presumably are triggered by intrinsic reward for generating a data stream with previously unknown regularities, such as the sensory input sequence corresponding to observing oneself juggling, which may be quite different from the more familiar experience of observing somebody else juggling, and therefore truly novel and intrinsically rewarding, until the adaptive predictor / compressor gets used to it.

3 Previous Concrete Implementations of Systems Driven by (Approximations of) Compression Progress

As mentioned earlier, predictors and compressors are closely related. Any type of partial predictability of the incoming sensory data stream can be exploited to improve the compressibility of the whole. Therefore the systems described in the first publications on artificial curiosity [57, 58, 61] already can be viewed as examples of implementations of a compression progress drive.

3.1 Reward for Prediction Error (1990)

Early work [57, 58, 61] described a predictor based on a recurrent neural network [115, 120, 55, 62, 47, 78] (in principle a rather powerful computational device, even by today's machine learning standards), predicting sensory inputs including reward signals from the entire history of previous inputs and actions. The curiosity rewards were proportional to the predictor errors, that is, it was implicitly and optimistically assumed that the predictor will indeed improve whenever its error is high.

3.2 Reward for Compression Progress Through Predictor Improvements (1991)

Follow-up work [59, 60] pointed out that this approach may be inappropriate, especially in probabilistic environments: one should not focus on the errors of the predictor, but on its improvements. Otherwise the system will concentrate its search on those parts of the environment where it can always get high prediction errors due to noise or randomness, or due to computational limitations of the predictor, which will prevent improvements of the subjective compressibility of the data. While the neural predictor of the implementation described in the follow-up work was indeed computationally less powerful than the previous one [61], there was a novelty, namely, an explicit (neural) adaptive model of the predictor's improvements. This model essentially learned to predict the predictor's changes. For example, although noise was unpredictable and led to wildly varying target signals for the predictor, in the long run these signals did not change the adaptive predictor parameters much, and the predictor of predictor changes was able to learn this. A standard RL algorithm [114, 33, 109] was fed with curiosity reward signals proportional to the expected long-term predictor changes, and thus tried to maximize information gain [16, 31, 38, 51, 14] within the given limitations. In fact, we may say that the system tried to maximize an approximation of the (discounted) sum of the expected first derivatives of the data's subjective predictability, thus also maximizing an approximation of the (discounted) sum of the expected changes of the data's subjective compressibility.

3.3 Reward for Relative Entropy between Agent's Prior and Posterior (1995)

Additional follow-up work yielded an information theory-oriented variant of the approach in non-deterministic worlds [108] (1995). The curiosity reward was again proportional to the predictor's surprise / information gain, this time measured as the Kullback-Leibler distance [35] between the learning predictor's subjective probability distributions before and after new observations - the relative entropy between its prior and posterior.

In 2005 Baldi and Itti called this approach "Bayesian surprise" and demonstrated experimentally that it explains certain patterns of human visual attention better than certain previous approaches [32].

Note that the concepts of Huffman coding [28] and relative entropy between prior and posterior immediately translate into a measure of learning progress reflecting the number of saved bits—a measure of improved data compression.

Note also, however, that the naive probabilistic approach to data compression is unable to discover more general types of *algorithmic* compressibility [106, 34, 37, 73]. For example, the decimal expansion of π looks random and incompressible but isn't: there is a very short algorithm computing all of π, yet any finite sequence of digits will occur in π's expansion as frequently as expected if π were truly random, that is, no simple statistical learner will outperform random guessing at predicting the next digit from a limited time window of previous digits. More general *program* search

techniques (e.g., [36, 75, 15, 46]) are necessary to extract the underlying algorithmic regularity.

3.4 Zero Sum Reward Games for Compression Progress Revealed by Algorithmic Experiments (1997)

More recent work [68, 72] (1997) greatly increased the computational power of controller and predictor by implementing them as co-evolving, symmetric, opposing modules consisting of self-modifying probabilistic programs [97, 98] written in a universal programming language [18, 111] allowing for loops, recursion, and hierarchical structures. The internal storage for temporary computational results of the programs was viewed as part of the changing environment. Each module could suggest experiments in the form of probabilistic algorithms to be executed, and make confident predictions about their effects by betting on their outcomes, where the *'betting money'* essentially played the role of the intrinsic reward. The opposing module could reject or accept the bet in a zero-sum game by making a contrary prediction. In case of acceptance, the winner was determined by executing the algorithmic experiment and checking its outcome; the money was eventually transferred from the surprised loser to the confirmed winner. Both modules tried to maximize their money using a rather general RL algorithm designed for complex stochastic policies [97, 98] (alternative RL algorithms could be plugged in as well). Thus both modules were motivated to discover *truly novel* algorithmic regularity / compressibility, where the subjective baseline for novelty was given by what the opponent already knew about the world's repetitive regularities.

The method can be viewed as system identification through co-evolution of computable models and tests. In 2005 a similar co-evolutionary approach based on less general models and tests was implemented by Bongard and Lipson [11].

3.5 Improving Real Reward Intake

Our references above demonstrated experimentally that the presence of intrinsic reward or curiosity reward actually can speed up the collection of *external* reward.

3.6 Other Implementations

Recently several researchers also implemented variants or approximations of the curiosity framework. Singh and Barto and coworkers focused on implementations within the option framework of RL [5, 104], directly using prediction errors as curiosity rewards as in Section 3.1 [57, 58, 61] —they actually were the ones who coined the expressions *intrinsic reward* and *intrinsically motivated* RL. Additional implementations were presented at the 2005 AAAI Spring Symposium on Developmental Robotics [9]; compare the Connection Science Special Issue [10].

4 Visual Illustrations of Subjective Beauty and its *First Derivative* Interestingness

As mentioned above (Section 3.3), the probabilistic variant of our theory [108] (1995) was able to explain certain shifts of human visual attention [32] (2005). But we can also apply our approach to the complementary problem of *constructing* images that contain quickly learnable regularities, arguing again that there is no fundamental difference between the motivation of creative artists and passive observers of visual art (Section 2.14). Both create action sequences yielding interesting inputs, where interestingness is a measure of learning progress, for example, based on the relative entropy between prior and posterior (Section 3.3), or the saved number of bits needed to encode the data (Section 1), or something similar (Section 3).

Here we provide examples of subjective beauty tailored to human observers, and illustrate the learning process leading from less to more subjective beauty. Due to the nature of the present written medium, we have to use visual examples instead of acoustic or tactile ones. Our examples are intended to support the hypothesis that unsupervised *attention* and the *creativity* of artists, dancers, musicians, pure mathematicians are just by-products of their compression progress drives.

4.1 A Pretty Simple Face with a Short Algorithmic Description

Figure 1 depicts the construction plan of a female face considered *'beautiful'* by some human observers. It also shows that the essential features of this face follow a very simple geometrical pattern [69] that can be specified by very few bits of information. That is, the data stream generated by observing the image (say, through a sequence of eye saccades) is more compressible than it would be in the absence of such regularities. Although few people are able to immediately see how the drawing was made in absence of its superimposed grid-based explanation, most do notice that the facial features somehow fit together and exhibit some sort of regularity. According to our postulate, the observer's reward is generated by the conscious or subconscious discovery of this compressibility. The face remains interesting until its observation does not reveal any additional previously unknown regularities. Then it becomes boring even in the eyes of those who think it is beautiful—as has been pointed out repeatedly above, beauty and interestingness are two different things.

4.2 Another Drawing That Can Be Encoded By Very Few Bits

Figure 2 provides another example: a butterfly and a vase with a flower. It can be specified by very few bits of information as it can be constructed through a very simple procedure or algorithm based on fractal circle patterns [67]—see Figure 3. People who understand this algorithm tend to appreciate the drawing more than those who do not. They realize how simple it is. This is not an immediate, all-or-nothing, binary process though. Since the typical human visual system has a lot of experience with circles, most people quickly notice that the curves somehow fit together in a regular way. But few are able to immediately state the precise geometric principles underlying the drawing

[81]. This pattern, however, is learnable from Figure 3. The conscious or subconscious discovery process leading from a longer to a shorter description of the data, or from less to more compression, or from less to more subjectively perceived beauty, yields reward depending on the first derivative of subjective beauty, that is, the steepness of the learning curve.

5 Conclusion & Outlook

We pointed out that a surprisingly simple algorithmic principle based on the notions of data compression and data compression *progress* informally explains fundamental aspects of attention, novelty, surprise, interestingness, curiosity, creativity, subjective beauty, jokes, and science & art in general. The crucial ingredients of the corresponding *formal* framework are (1) a continually improving predictor or compressor of the continually growing data history, (2) a computable measure of the compressor's progress (to calculate intrinsic rewards), (3) a reward optimizer or reinforcement learner translating rewards into action sequences expected to maximize future reward. To improve our previous implementations of these ingredients (Section 3), we will (1) study better adaptive compressors, in particular, recent, novel RNNs [94] and other general but practically feasible methods for making predictions [75]; (2) investigate under which conditions learning progress measures can be computed both accurately and efficiently, without frequent expensive compressor performance evaluations on the entire history so far; (3) study the applicability of recent improved RL techniques in the fields of policy gradients [110, 119, 118, 56, 100, 117], artificial evolution [43, 20, 21, 19, 22, 23, 24], and others [71, 75].

Apart from building improved *artificial* curious agents, we can test the predictions of our theory in psychological investigations of *human* behavior, extending previous studies in this vein [32] and going beyond anecdotal evidence mentioned above. It should be easy to devise controlled experiments where test subjects must anticipate initially unknown but causally connected event sequences exhibiting more or less complex, learnable patterns or regularities. The subjects will be asked to quantify their intrinsic rewards in response to their improved predictions. Is the reward indeed strongest when the predictions are improving most rapidly? Does the intrinsic reward indeed vanish as the predictions become perfect or do not improve any more?

Finally, how to test our predictions through studies in neuroscience? Currently we hardly understand the human neural machinery. But it is well-known that certain neurons seem to predict others, and brain scans show how certain brain areas light up in response to reward. Therefore the psychological experiments suggested above should be accompanied by neurophysiological studies to localize the origins of intrinsic rewards, possibly linking them to improvements of neural predictors.

Success in this endeavor would provide additional motivation to implement our principle on robots.

A Appendix

This appendix is based in part on references [81, 88].

The world can be explained to a degree by compressing it. Discoveries correspond to large data compression improvements (found by the given, application-dependent compressor improvement algorithm). How to build an adaptive agent that not only tries to achieve externally given rewards but also to discover, in an unsupervised and experiment-based fashion, explainable and compressible data? (The explanations gained through explorative behavior may eventually help to solve teacher-given tasks.)

Let us formally consider a learning agent whose single life consists of discrete cycles or time steps $t = 1, 2, \ldots, T$. Its complete lifetime T may or may not be known in advance. In what follows, the value of any time-varying variable Q at time t ($1 \leq t \leq T$) will be denoted by $Q(t)$, the ordered sequence of values $Q(1), \ldots, Q(t)$ by $Q(\leq t)$, and the (possibly empty) sequence $Q(1), \ldots, Q(t-1)$ by $Q(< t)$. At any given t the agent receives a real-valued input $x(t)$ from the environment and executes a real-valued action $y(t)$ which may affect future inputs. At times $t < T$ its goal is to maximize future success or *utility*

$$u(t) = E_\mu \left[\sum_{\tau=t+1}^{T} r(\tau) \;\middle|\; h(\leq t) \right], \tag{2}$$

where $r(t)$ is an additional real-valued reward input at time t, $h(t)$ the ordered triple $[x(t), y(t), r(t)]$ (hence $h(\leq t)$ is the known history up to t), and $E_\mu(\cdot \mid \cdot)$ denotes the conditional expectation operator with respect to some possibly unknown distribution μ from a set $\mathcal{M}$ of possible distributions. Here $\mathcal{M}$ reflects whatever is known about the possibly probabilistic reactions of the environment. For example, $\mathcal{M}$ may contain all computable distributions [106, 107, 37, 29]. There is just one life, no need for predefined repeatable trials, no restriction to Markovian interfaces between sensors and environment, and the utility function implicitly takes into account the expected remaining lifespan $E_\mu(T \mid h(\leq t))$ and thus the possibility to extend it through appropriate actions [79, 82, 80, 92].

Recent work has led to the first learning machines that are universal and optimal in various very general senses [29, 79, 82]. As mentioned in the introduction, such machines can in principle find out by themselves whether curiosity and world model construction are useful or useless in a given environment, and learn to behave accordingly. The present appendix, however, will assume *a priori* that compression / explanation of the history is good and should be done; here we shall not worry about the possibility that curiosity can be harmful and "kill the cat." Towards this end, in the spirit of our previous work since 1990 [57, 58, 61, 59, 60, 108, 68, 72, 76, 81, 88, 87, 89] we split the reward signal $r(t)$ into two scalar real-valued components: $r(t) = g(r_{ext}(t), r_{int}(t))$, where g maps pairs of real values to real values, e.g., $g(a, b) = a + b$. Here $r_{ext}(t)$ denotes traditional *external* reward provided by the environment, such as negative reward in response to bumping against a wall, or positive reward in response to reaching some teacher-given goal state. But for the purposes of this paper we are especially interested in $r_{int}(t)$, the internal or intrinsic or *curiosity* reward, which is provided whenever the data compressor / internal world model of the agent improves in some

measurable sense. Our initial focus will be on the case $r_{ext}(t) = 0$ for all valid t. The basic principle is essentially the one we published before in various variants [57, 58, 61, 59, 60, 108, 68, 72, 76, 81, 88, 87]:

Principle 1 *Generate curiosity reward for the controller in response to improvements of the predictor or history compressor.*

So we conceptually separate the goal (explaining / compressing the history) from the means of achieving the goal. Once the goal is formally specified in terms of an algorithm for computing curiosity rewards, let the controller's reinforcement learning (RL) mechanism figure out how to translate such rewards into action sequences that allow the given compressor improvement algorithm to find and exploit previously unknown types of compressibility.

A.1 Predictors vs Compressors

Much of our previous work on artificial curiosity was prediction-oriented, e. g., [57, 58, 61, 59, 60, 108, 68, 72, 76]. Prediction and compression are closely related though. A predictor that correctly predicts many $x(\tau)$, given history $h(< \tau)$, for $1 \leq \tau \leq t$, can be used to encode $h(\leq t)$ compactly. Given the predictor, only the wrongly predicted $x(\tau)$ plus information about the corresponding time steps τ are necessary to reconstruct history $h(\leq t)$, e.g., [63]. Similarly, a predictor that learns a probability distribution of the possible next events, given previous events, can be used to efficiently encode observations with high (respectively low) predicted probability by few (respectively many) bits [28, 95], thus achieving a compressed history representation. Generally speaking, we may view the predictor as the essential part of a program p that re-computes $h(\leq t)$. If this program is short in comparison to the raw data $h(\leq t)$, then $h(\leq t)$ is regular or non-random [106, 34, 37, 73], presumably reflecting essential environmental laws. Then p may also be highly useful for predicting future, yet unseen $x(\tau)$ for $\tau > t$.

It should be mentioned, however, that the compressor-oriented approach to prediction based on the principle of Minimum Description Length (MDL) [34, 112, 113, 54, 37] does not necessarily converge to the correct predictions as quickly as Solomonoff's universal inductive inference [106, 107, 37], although both approaches converge in the limit under general conditions [52].

A.2 Which Predictor or History Compressor?

The complexity of evaluating some compressor p on history $h(\leq t)$ depends on both p and its performance measure C. Let us first focus on the former. Given t, one of the simplest p will just use a linear mapping to predict $x(t+1)$ from $x(t)$ and $y(t+1)$. More complex p such as adaptive recurrent neural networks (RNN) [115, 120, 55, 62, 47, 26, 93, 77, 78] will use a nonlinear mapping and possibly the entire history $h(\leq t)$ as a basis for the predictions. In fact, the first work on artificial curiosity [61] focused on online learning RNN of this type. A theoretically optimal predictor would be Solomonoff's above-mentioned universal induction scheme [106, 107, 37].

A.3 Compressor Performance Measures

At any time t ($1 \leq t < T$), given some compressor program p able to compress history $h(\leq t)$, let $C(p, h(\leq t))$ denote p's compression performance on $h(\leq t)$. An appropriate performance measure would be

$$C_l(p, h(\leq t)) = l(p), \tag{3}$$

where $l(p)$ denotes the length of p, measured in number of bits: the shorter p, the more algorithmic regularity and compressibility and predictability and lawfulness in the observations so far. The ultimate limit for $C_l(p, h(\leq t))$ would be $K^*(h(\leq t))$, a variant of the Kolmogorov complexity of $h(\leq t)$, namely, the length of the shortest program (for the given hardware) that computes an output starting with $h(\leq t)$ [106, 34, 37, 73].

A.4 Compressor Performance Measures Taking Time Into Account

$C_l(p, h(\leq t))$ does not take into account the time $\tau(p, h(\leq t))$ spent by p on computing $h(\leq t)$. An alternative performance measure inspired by concepts of optimal universal search [36, 75] is

$$C_{l\tau}(p, h(\leq t)) = l(p) + \log\ \tau(p, h(\leq t)). \tag{4}$$

Here compression by one bit is worth as much as runtime reduction by a factor of $\frac{1}{2}$. From an asymptotic optimality-oriented point of view this is one of the best ways of trading off storage and computation time [36, 75].

A.5 Measures of Compressor Progress / Learning Progress

The previous sections only discussed measures of compressor performance, but not of performance *improvement*, which is the essential issue in our curiosity-oriented context. To repeat the point made above: *The important thing are the improvements of the compressor, not its compression performance per se.* Our curiosity reward in response to the compressor's progress (due to some application-dependent compressor improvement algorithm) between times t and $t+1$ should be

$$r_{int}(t+1) = f[C(p(t), h(\leq t+1)), C(p(t+1), h(\leq t+1))], \tag{5}$$

where f maps pairs of real values to real values. Various alternative progress measures are possible; most obvious is $f(a, b) = a - b$. This corresponds to a discrete time version of maximizing the first derivative of subjective data compressibility.

Note that both the old and the new compressor have to be tested on the same data, namely, the history so far.

A.6 Asynchronous Framework for Creating Curiosity Reward

Let $p(t)$ denote the agent's current compressor program at time t, $s(t)$ its current controller, and do:

Controller: At any time t ($1 \leq t < T$) do:

1. Let $s(t)$ use (parts of) history $h(\leq t)$ to select and execute $y(t+1)$.
2. Observe $x(t+1)$.
3. Check if there is non-zero curiosity reward $r_{int}(t+1)$ provided by the separate, asynchronously running compressor improvement algorithm (see below). If not, set $r_{int}(t+1) = 0$.
4. Let the controller's reinforcement learning (RL) algorithm use $h(\leq t+1)$ including $r_{int}(t+1)$ (and possibly also the latest available compressed version of the observed data—see below) to obtain a new controller $s(t+1)$, in line with objective (2).

Compressor: Set p_{new} equal to the initial data compressor. Starting at time 1, repeat forever until interrupted by death at time T:

1. Set $p_{old} = p_{new}$; get current time step t and set $h_{old} = h(\leq t)$.
2. Evaluate p_{old} on h_{old}, to obtain $C(p_{old}, h_{old})$ (Section A.3). This may take many time steps.
3. Let some (application-dependent) compressor improvement algorithm (such as a learning algorithm for an adaptive neural network predictor) use h_{old} to obtain a hopefully better compressor p_{new} (such as a neural net with the same size but improved prediction capability and therefore improved compression performance [95]). Although this may take many time steps (and could be partially performed during "sleep"), p_{new} may not be optimal, due to limitations of the learning algorithm, e.g., local maxima.
4. Evaluate p_{new} on h_{old}, to obtain $C(p_{new}, h_{old})$. This may take many time steps.
5. Get current time step τ and generate curiosity reward

$$r_{int}(\tau) = f[C(p_{old}, h_{old}), C(p_{new}, h_{old})], \tag{6}$$

   e.g., $f(a, b) = a - b$; see Section A.5.

Obviously this asynchronuous scheme may cause long temporal delays between controller actions and corresponding curiosity rewards. This may impose a heavy burden on the controller's RL algorithm whose task is to assign credit to past actions (to inform the controller about beginnings of compressor evaluation processes etc., we may augment its input by unique representations of such events). Nevertheless, there are RL algorithms for this purpose which are theoretically optimal in various senses, to be discussed next.

A.7 Optimal Curiosity & Creativity & Focus of Attention

Our chosen compressor class typically will have certain computational limitations. In the absence of any external rewards, we may define *optimal pure curiosity behavior* relative to these limitations: At time t this behavior would select the action that maximizes

$$u(t) = E_\mu \left[\sum_{\tau=t+1}^{T} r_{int}(\tau) \;\middle|\; h(\leq t) \right]. \tag{7}$$

Since the true, world-governing probability distribution μ is unknown, the resulting task of the controller's RL algorithm may be a formidable one. As the system is revisiting previously incompressible parts of the environment, some of those will tend to become more subjectively compressible, and the corresponding curiosity rewards will decrease over time. A good RL algorithm must somehow detect and then *predict* this decrease, and act accordingly. Traditional RL algorithms [33], however, do not provide any theoretical guarantee of optimality for such situations. (This is not to say though that sub-optimal RL methods may not lead to success in certain applications; experimental studies might lead to interesting insights.)

Let us first make the natural assumption that the compressor is not super-complex such as Kolmogorov's, that is, its output and $r_{int}(t)$ are computable for all t. Is there a best possible RL algorithm that comes as close as any other to maximizing objective (7)? Indeed, there is. Its drawback, however, is that it is not computable in finite time. Nevertheless, it serves as a reference point for defining what is achievable at best.

A.8 Optimal But Incomputable Action Selector

There is an optimal way of selecting actions which makes use of Solomonoff's theoretically optimal universal predictors and their Bayesian learning algorithms [106, 107, 37, 29, 30]. The latter only assume that the reactions of the environment are sampled from an unknown probability distribution μ contained in a set $\mathcal{M}$ of all enumerable distributions—compare text after equation (2). More precisely, given an observation sequence $q(\leq t)$ we want to use the Bayes formula to predict the probability of the next possible $q(t+1)$. Our only assumption is that there exists a computer program that can take any $q(\leq t)$ as an input and compute its *a priori* probability according to the μ prior. In general we do not know this program, hence we predict using a mixture prior instead:

$$\xi(q(\leq t)) = \sum_i w_i \mu_i(q(\leq t)), \tag{8}$$

a weighted sum of *all* distributions $\mu_i \in \mathcal{M}$, $i = 1, 2, \ldots$, where the sum of the constant positive weights satisfies $\sum_i w_i \leq 1$. This is indeed the best one can possibly do, in a very general sense [107, 29]. The drawback of the scheme is its incomputability, since $\mathcal{M}$ contains infinitely many distributions. We may increase the theoretical power of the scheme by augmenting $\mathcal{M}$ by certain non-enumerable but limit-computable distributions [73], or restrict it such that it becomes computable, e.g., by assuming the world is computed by some unknown but deterministic computer program sampled

from the Speed Prior [74] which assigns low probability to environments that are hard to compute by any method.

Once we have such an optimal predictor, we can extend it by formally including the effects of executed actions to define an optimal action selector maximizing future expected reward. At any time t, Hutter's theoretically optimal (yet uncomputable) RL algorithm AIXI [29] uses an extended version of Solomonoff's prediction scheme to select those action sequences that promise maximal future reward up to some horizon T, given the current data $h(\leq t)$. That is, in cycle $t+1$, AIXI selects as its next action the first action of an action sequence maximizing ξ-predicted reward up to the given horizon, appropriately generalizing eq. (8). AIXI uses observations optimally [29]: the Bayes-optimal policy p^ξ based on the mixture ξ is self-optimizing in the sense that its average utility value converges asymptotically for all $\mu \in \mathcal{M}$ to the optimal value achieved by the Bayes-optimal policy p^μ which knows μ in advance. The necessary and sufficient condition is that $\mathcal{M}$ admits self-optimizing policies. The policy p^ξ is also Pareto-optimal in the sense that there is no other policy yielding higher or equal value in *all* environments $\nu \in \mathcal{M}$ and a strictly higher value in at least one [29].

A.9 A Computable Selector of Provably Optimal Actions

AIXI above needs unlimited computation time. Its computable variant AIXI*(t,l)* [29] has asymptotically optimal runtime but may suffer from a huge constant slowdown. To take the consumed computation time into account in a general, optimal way, we may use the recent Gödel machines [79, 82, 80, 92] instead. They represent the first class of mathematically rigorous, fully self-referential, self-improving, general, optimally efficient problem solvers. They are also applicable to the problem embodied by objective (7).

The initial software $\mathcal{S}$ of such a Gödel machine contains an initial problem solver, e.g., some typically sub-optimal method [33]. It also contains an asymptotically optimal initial proof searcher based on an online variant of Levin's *Universal Search* [36], which is used to run and test *proof techniques*. Proof techniques are programs written in a universal language implemented on the Gödel machine within $\mathcal{S}$. They are in principle able to compute proofs concerning the system's own future performance, based on an axiomatic system $\mathcal{A}$ encoded in $\mathcal{S}$. $\mathcal{A}$ describes the formal *utility* function, in our case eq. (7), the hardware properties, axioms of arithmetic and probability theory and data manipulation etc, and $\mathcal{S}$ itself, which is possible without introducing circularity [92].

Inspired by Kurt Gödel's celebrated self-referential formulas (1931), the Gödel machine rewrites any part of its own code (including the proof searcher) through a self-generated executable program as soon as its *Universal Search* variant has found a proof that the rewrite is *useful* according to objective (7). According to the Global Optimality Theorem [79, 82, 80, 92], such a self-rewrite is globally optimal—no local maxima possible!—since the self-referential code first had to prove that it is not useful to continue the search for alternative self-rewrites.

If there is no provably useful optimal way of rewriting $\mathcal{S}$ at all, then humans will not find one either. But if there is one, then $\mathcal{S}$ itself can find and exploit it. Unlike the previous *non*-self-referential methods based on hardwired proof searchers [29], Gödel ma-

chines not only boast an optimal *order* of complexity but can optimally reduce (through self-changes) any slowdowns hidden by the $O()$-notation, provided the utility of such speed-ups is provable. Compare [83, 86, 85].

A.10 Non-Universal But Still General and Practical RL Algorithms

Recently there has been substantial progress in RL algorithms that are not quite as universal as those above, but nevertheless capable of learning very general, program-like behavior. In particular, evolutionary methods [53, 99, 27] can be used for training Recurrent Neural Networks (RNN), which are general computers. Many approaches to evolving RNN have been proposed [40, 122, 121, 45, 39, 103, 42]. One particularly effective family of methods uses cooperative coevolution to search the space of network components (*neurons* or individual *synapses*) instead of complete networks. The components are *coevolved* by combining them into networks, and selecting those for reproduction that participated in the best performing networks [43, 20, 21, 19, 22, 24]. Other recent RL techniques for RNN are based on the concept of policy gradients [110, 119, 118, 56, 100, 117]. It will be of interest to evaluate variants of such control learning algorithms within the curiosity reward framework.

A.11 Acknowledgments

Thanks to Marcus Hutter, Andy Barto, Jonathan Lansey, Julian Togelius, Faustino J. Gomez, Giovanni Pezzulo, Gianluca Baldassarre, Martin Butz, for useful comments that helped to improve the first version of this paper.

References

[1] I. Aleksander. *The World in My Mind, My Mind In The World: Key Mechanisms of Consciousness in Humans, Animals and Machines.* Imprint Academic, 2005.

[2] B. Baars and N. M. Gage. *Cognition, Brain and Consciousness: An Introduction to Cognitive Neuroscience*. Elsevier / Academic Press, 2007.

[3] M. Balter. Seeking the key to music. *Science*, 306:1120–1122, 2004.

[4] H. B. Barlow, T. P. Kaushal, and G. J. Mitchison. Finding minimum entropy codes. *Neural Computation*, 1(3):412–423, 1989.

[5] A. G. Barto, S. Singh, and N. Chentanez. Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of International Conference on Developmental Learning (ICDL)*. MIT Press, Cambridge, MA, 2004.

[6] M. Bense. *Einführung in die informationstheoretische Ästhetik. Grundlegung und Anwendung in der Texttheorie (Introduction to information-theoretical aesthetics. Foundation and application to text theory).* Rowohlt Taschenbuch Verlag, 1969.

[7] G. D. Birkhoff. *Aesthetic Measure*. Harvard University Press, 1933.

[8] C. M. Bishop. *Neural networks for pattern recognition.* Oxford University Press, 1995.

[9] D. Blank and L. Meeden. Developmental Robotics AAAI Spring Symposium, Stanford, CA, 2005. http://cs.brynmawr.edu/DevRob05/schedule/.

[10] D. Blank and L. Meeden. Introduction to the special issue on developmental robotics. *Connection Science*, 18(2), 2006.

[11] J. C. Bongard and H. Lipson. Nonlinear system identification using coevolution of models and tests. *IEEE Transactions on Evolutionary Computation*, 9(4), 2005.

[12] M. V. Butz. How and why the brain lays the foundations for a conscious self. *Constructivist Foundations*, 4(1):1–14, 2008.

[13] L. D. Cañamero. Designing emotions for activity selection in autonomous agents. In R. Trappl, P. Petta, and S. Payr, editors, *Emotions in Humans and Artifacts*, pages 115–148. The MIT Press, Cambridge, MA, 2003.

[14] D. A. Cohn. Neural network exploration using optimal experiment design. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 679–686. Morgan Kaufmann, 1994.

[15] N. L. Cramer. A representation for the adaptive generation of simple sequential programs. In J.J. Grefenstette, editor, *Proceedings of an International Conference on Genetic Algorithms and Their Applications, Carnegie-Mellon University, July 24-26, 1985*, Hillsdale NJ, 1985. Lawrence Erlbaum Associates.

[16] V. V. Fedorov. *Theory of optimal experiments.* Academic Press, 1972.

[17] F. Galton. Composite portraits made by combining those of many different persons into a single figure. *Nature*, 18(9):97–100, 1878.

[18] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931.

[19] F. J. Gomez. *Robust Nonlinear Control through Neuroevolution.* PhD thesis, Department of Computer Sciences, University of Texas at Austin, 2003.

[20] F. J. Gomez and R. Miikkulainen. Incremental evolution of complex general behavior. *Adaptive Behavior*, 5:317–342, 1997.

[21] F. J. Gomez and R. Miikkulainen. Solving non-Markovian control tasks with neuroevolution. In *Proc. IJCAI 99*, Denver, CO, 1999. Morgan Kaufman.

[22] F. J. Gomez and R. Miikkulainen. Active guidance for a finless rocket using neuroevolution. In *Proc. GECCO 2003, Chicago*, 2003. *Winner of Best Paper Award in Real World Applications. Gomez is working at IDSIA on a CSEM grant to J. Schmidhuber.*

[23] F. J. Gomez and J. Schmidhuber. Co-evolving recurrent neurons learn deep memory POMDPs. In *Proc. of the 2005 conference on genetic and evolutionary computation (GECCO), Washington, D. C.* ACM Press, New York, NY, USA, 2005. *Nominated for a best paper award.*

[24] F. J. Gomez, J. Schmidhuber, and R. Miikkulainen. Efficient non-linear control through neuroevolution. *Journal of Machine Learning Research JMLR*, 9:937–965, 2008.

[25] P. Haikonen. *The Cognitive Approach to Conscious Machines.* Imprint Academic, 2003.

[26] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[27] J. H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, 1975.

[28] D. A. Huffman. A method for construction of minimum-redundancy codes. *Proceedings IRE*, 40:1098–1101, 1952.

[29] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, Berlin, 2004. (On J. Schmidhuber's SNF grant 20-61847).

[30] M. Hutter. On universal prediction and Bayesian confirmation. *Theoretical Computer Science*, 2007.

[31] J. Hwang, J. Choi, S. Oh, and R. J. Marks II. Query-based learning applied to partially trained multilayer perceptrons. *IEEE Transactions on Neural Networks*, 2(1):131–136, 1991.

[32] L. Itti and P. F. Baldi. Bayesian surprise attracts human attention. In *Advances in Neural Information Processing Systems 19*, pages 547–554. MIT Press, Cambridge, MA, 2005.

[33] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: a survey. *Journal of AI research*, 4:237–285, 1996.

[34] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1:1–11, 1965.

[35] S. Kullback. *Statistics and Information Theory*. J. Wiley and Sons, New York, 1959.

[36] L. A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9(3):265–266, 1973.

[37] M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications (2nd edition)*. Springer, 1997.

[38] D. J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(2):550–604, 1992.

[39] O. Miglino, H. Lund, and S. Nolfi. Evolving mobile robots in simulated and real environments. *Artificial Life*, 2(4):417–434, 1995.

[40] G. Miller, P. Todd, and S. Hedge. Designing neural networks using genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 379–384. Morgan Kauffman, 1989.

[41] A. Moles. *Information Theory and Esthetic Perception.* Univ. of Illinois Press, 1968.

[42] D. E. Moriarty and P. Langley. Learning cooperative lane selection strategies for highways. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 684–691, Madison, WI, 1998.

[43] D. E. Moriarty and R. Miikkulainen. Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, 22:11–32, 1996.

[44] F. Nake. *Ästhetik als Informationsverarbeitung*. Springer, 1974.

[45] S. Nolfi, D. Floreano, O. Miglino, and F. Mondada. How to evolve autonomous robots: Different approaches in evolutionary robotics. In R. A. Brooks and P. Maes, editors, *Fourth International Workshop on the Synthesis and Simulation of Living Systems (Artificial Life IV)*, pages 190–197. MIT, 1994.

[46] J. R. Olsson. Inductive functional programming using incremental program transformation. *Artificial Intelligence*, 74(1):55–83, 1995.

[47] B. A. Pearlmutter. Gradient calculations for dynamic recurrent neural networks: A survey. *IEEE Transactions on Neural Networks*, 6(5):1212–1228, 1995.

[48] D. I. Perrett, K . A. May, and S. Yoshikawa. Facial shape and judgements of female attractiveness. *Nature*, 368:239–242, 1994.

[49] J. Piaget. *The Child's Construction of Reality*. London: Routledge and Kegan Paul, 1955.

[50] S. Pinker. *How the mind works*. Norton, W. W. & Company, Inc., 1997.

[51] M. Plutowski, G. Cottrell, and H. White. Learning Mackey-Glass from 25 examples, plus or minus 2. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 1135–1142. Morgan Kaufmann, 1994.

[52] J. Poland and M. Hutter. Strong asymptotic assertions for discrete MDL in regression and classification. In *Annual Machine Learning Conference of Belgium and the Netherlands (Benelearn-2005)*, Enschede, 2005.

[53] I. Rechenberg. Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Dissertation, 1971. Published 1973 by Fromman-Holzboog.

[54] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[55] A. J. Robinson and F. Fallside. The utility driven dynamic error propagation network. Technical Report CUED/F-INFENG/TR.1, Cambridge University Engineering Department, 1987.

[56] T. Rückstieß, M. Felder, and J. Schmidhuber. State-Dependent Exploration for policy gradient methods. In W. Daelemans et al., editor, *European Conference on Machine Learning (ECML) and Principles and Practice of Knowledge Discovery in Databases 2008, Part II, LNAI 5212*, pages 234–249, 2008.

[57] J. Schmidhuber. Dynamische neuronale Netze und das fundamentale raumzeitliche Lernproblem. Dissertation, Institut für Informatik, Technische Universität München, 1990.

[58] J. Schmidhuber. Making the world differentiable: On using fully recurrent self-supervised neural networks for dynamic reinforcement learning and planning in non-stationary environments. Technical Report FKI-126-90, Institut für Informatik, Technische Universität München, 1990.

[59] J. Schmidhuber. Adaptive curiosity and adaptive confidence. Technical Report FKI-149-91, Institut für Informatik, Technische Universität München, April 1991. See also [60].

[60] J. Schmidhuber. Curious model-building control systems. In *Proceedings of the International Joint Conference on Neural Networks, Singapore*, volume 2, pages 1458–1463. IEEE press, 1991.

[61] J. Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In J. A. Meyer and S. W. Wilson, editors, *Proc. of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 222–227. MIT Press/Bradford Books, 1991.

[62] J. Schmidhuber. A fixed size storage $O(n^3)$ time complexity learning algorithm for fully recurrent continually running networks. *Neural Computation*, 4(2):243–248, 1992.

[63] J. Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242, 1992.

[64] J. Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992.

[65] J. Schmidhuber. A computer scientist's view of life, the universe, and everything. In C. Freksa, M. Jantzen, and R. Valk, editors, *Foundations of Computer Science: Potential - Theory - Cognition*, volume 1337, pages 201–208. Lecture Notes in Computer Science, Springer, Berlin, 1997.

[66] J. Schmidhuber. Femmes fractales, 1997.

[67] J. Schmidhuber. Low-complexity art. *Leonardo, Journal of the International Society for the Arts, Sciences, and Technology*, 30(2):97–103, 1997.

[68] J. Schmidhuber. What's interesting? Technical Report IDSIA-35-97, IDSIA, 1997. ftp://ftp.idsia.ch/pub/juergen/interest.ps.gz; extended abstract in Proc. Snowbird'98, Utah, 1998; see also [72].

[69] J. Schmidhuber. Facial beauty and fractal geometry. Technical Report TR IDSIA-28-98, IDSIA, 1998. Published in the Cogprint Archive: http://cogprints.soton.ac.uk.

[70] J. Schmidhuber. Algorithmic theories of everything. Technical Report IDSIA-20-00, quant-ph/0011122, IDSIA, Manno (Lugano), Switzerland, 2000. Sections 1-5: see [73]; Section 6: see [74].

[71] J. Schmidhuber. Sequential decision making based on direct search. In R. Sun and C. L. Giles, editors, *Sequence Learning: Paradigms, Algorithms, and Applications*. Springer, 2001. Lecture Notes on AI 1828.

[72] J. Schmidhuber. Exploring the predictable. In A. Ghosh and S. Tsuitsui, editors, *Advances in Evolutionary Computing*, pages 579–612. Springer, 2002.

[73] J. Schmidhuber. Hierarchies of generalized Kolmogorov complexities and nonenumerable universal measures computable in the limit. *International Journal of Foundations of Computer Science*, 13(4):587–612, 2002.

[74] J. Schmidhuber. The Speed Prior: a new simplicity measure yielding near-optimal computable predictions. In J. Kivinen and R. H. Sloan, editors, *Proceedings of the 15th Annual Conference on Computational Learning Theory (COLT 2002)*, Lecture Notes in Artificial Intelligence, pages 216–228. Springer, Sydney, Australia, 2002.

[75] J. Schmidhuber. Optimal ordered problem solver. *Machine Learning*, 54:211–254, 2004.

[76] J. Schmidhuber. Overview of artificial curiosity and active exploration, with links to publications since 1990, 2004. http://www.idsia.ch/˜juergen/interest.html.

[77] J. Schmidhuber. Overview of work on robot learning, with publications, 2004. http://www.idsia.ch/˜juergen/learningrobots.html.

[78] J. Schmidhuber. RNN overview, with links to a dozen journal publications, 2004. http://www.idsia.ch/˜juergen/rnn.html.

[79] J. Schmidhuber. Completely self-referential optimal reinforcement learners. In W. Duch, J. Kacprzyk, E. Oja, and S. Zadrozny, editors, *Artificial Neural Networks: Biological Inspirations - ICANN 2005, LNCS 3697*, pages 223–233. Springer-Verlag Berlin Heidelberg, 2005. Plenary talk.

[80] J. Schmidhuber. Gödel machines: Towards a technical justification of consciousness. In D. Kudenko, D. Kazakov, and E. Alonso, editors, *Adaptive Agents and Multi-Agent Systems III (LNCS 3394)*, pages 1–23. Springer Verlag, 2005.

[81] J. Schmidhuber. Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science*, 18(2):173–187, 2006.

[82] J. Schmidhuber. Gödel machines: Fully self-referential optimal universal self-improvers. In B. Goertzel and C. Pennachin, editors, *Artificial General Intelligence*, pages 199–226. Springer Verlag, 2006. Variant available as arXiv:cs.LO/0309048.

[83] J. Schmidhuber. The new AI: General & sound & relevant for physics. In B. Goertzel and C. Pennachin, editors, *Artificial General Intelligence*, pages 175–198. Springer, 2006. Also available as TR IDSIA-04-03, arXiv:cs.AI/0302012.

[84] J. Schmidhuber. Randomness in physics. *Nature*, 439(3):392, 2006. Correspondence.

[85] J. Schmidhuber. 2006: Celebrating 75 years of AI - history and outlook: the next 25 years. In M. Lungarella, F. Iida, J. Bongard, and R. Pfeifer, editors, *50 Years of Artificial Intelligence*, volume LNAI 4850, pages 29–41. Springer Berlin / Heidelberg, 2007. Preprint available as arXiv:0708.4311.

[86] J. Schmidhuber. New millennium AI and the convergence of history. In W. Duch and J. Mandziuk, editors, *Challenges to Computational Intelligence*, volume 63, pages 15–36. Studies in Computational Intelligence, Springer, 2007. Also available as arXiv:cs.AI/0606081.

[87] J. Schmidhuber. Simple algorithmic principles of discovery, subjective beauty, selective attention, curiosity & creativity. In *Proc. 18th Intl. Conf. on Algorithmic Learning Theory (ALT 2007), LNAI 4754*, pages 32–33. Springer, 2007. Joint invited lecture for *ALT 2007 and DS 2007*, Sendai, Japan, 2007.

[88] J. Schmidhuber. Simple algorithmic principles of discovery, subjective beauty, selective attention, curiosity & creativity. In *Proc. 10th Intl. Conf. on Discovery Science (DS 2007), LNAI 4755*, pages 26–38. Springer, 2007. Joint invited lecture for *ALT 2007 and DS 2007*, Sendai, Japan, 2007.

[89] J. Schmidhuber. Driven by compression progress. In I. Lovrek, R. J. Howlett, and L. C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems KES-2008*, Lecture Notes in Computer Science LNCS 5177, Part I, page 11. Springer, 2008. Abstract of invited keynote.

[90] J. Schmidhuber. Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. In G. Pezzulo, M. V. Butz, O. Sigaud, and G. Baldassarre, editors, *Anticipatory Behavior in Adaptive Learning Systems, from Sensorimotor to Higher-level Cognitive Capabilities*, LNAI. Springer, 2009. In press.

[91] J. Schmidhuber. Simple algorithmic theory of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. *Journal of SICE*, 48(1):21–32, 2009.

[92] J. Schmidhuber. Ultimate cognition *à la* Gödel. *Cognitive Computation*, 2009, in press.

[93] J. Schmidhuber and B. Bakker. NIPS 2003 RNNaissance workshop on recurrent neural networks, Whistler, CA, 2003. http://www.idsia.ch/˜juergen/rnnaissance.html.

[94] J. Schmidhuber, A. Graves, F. J. Gomez, S. Fernandez, and S. Hochreiter. *How to Learn Programs with Artificial Recurrent Neural Networks.* Invited by Cambridge University Press, 2009. In preparation.

[95] J. Schmidhuber and S. Heil. Sequential neural text compression. *IEEE Transactions on Neural Networks*, 7(1):142–146, 1996.

[96] J. Schmidhuber and R. Huber. Learning to generate artificial fovea trajectories for target detection. *International Journal of Neural Systems*, 2(1 & 2):135–141, 1991.

[97] J. Schmidhuber, J. Zhao, and N. Schraudolph. Reinforcement learning with self-modifying policies. In S. Thrun and L. Pratt, editors, *Learning to learn*, pages 293–309. Kluwer, 1997.

[98] J. Schmidhuber, J. Zhao, and M. Wiering. Shifting inductive bias with success-story algorithm, adaptive Levin search, and incremental self-improvement. *Machine Learning*, 28:105–130, 1997.

[99] H. P. Schwefel. Numerische Optimierung von Computer-Modellen. Dissertation, 1974. Published 1977 by Birkhäuser, Basel.

[100] F. Sehnke, C. Osendorfer, T. Rückstieß, A. Graves, J. Peters, and J. Schmidhuber. Policy gradients with parameter-based exploration for control. In *Proceedings of the International Conference on Artificial Neural Networks ICANN*, 2008.

[101] A. K. Seth, E. Izhikevich, G. N. Reeke, and G. M. Edelman. Theories and measures of consciousness: An extended framework. *Proc. Natl. Acad. Sciences USA*, 103:10799–10804, 2006.

[102] C. E. Shannon. A mathematical theory of communication (parts I and II). *Bell System Technical Journal*, XXVII:379–423, 1948.

[103] K. Sims. Evolving virtual creatures. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 1994)*, Computer Graphics Proceedings, Annual Conference, pages 15–22. ACM SIGGRAPH, ACM Press, jul 1994. ISBN 0-89791-667-0.

[104] S. Singh, A. G. Barto, and N. Chentanez. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems 17 (NIPS)*. MIT Press, Cambridge, MA, 2005.

[105] A. Sloman and R. L. Chrisley. Virtual machines and consciousness. *Journal of Consciousness Studies*, 10(4-5):113–172, 2003.

[106] R. J. Solomonoff. A formal theory of inductive inference. Part I. *Information and Control*, 7:1–22, 1964.

[107] R. J. Solomonoff. Complexity-based induction systems. *IEEE Transactions on Information Theory*, IT-24(5):422–432, 1978.

[108] J. Storck, S. Hochreiter, and J. Schmidhuber. Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the International Conference on Artificial Neural Networks, Paris*, volume 2, pages 159–164. EC2 & Cie, 1995.

[109] R. Sutton and A. Barto. *Reinforcement learning: An introduction*. Cambridge, MA, MIT Press, 1998.

[110] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pages 1057–1063. The MIT Press, 1999.

[111] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, Series 2*, 41:230–267, 1936.

[112] C. S. Wallace and D. M. Boulton. An information theoretic measure for classification. *Computer Journal*, 11(2):185–194, 1968.

[113] C. S. Wallace and P. R. Freeman. Estimation and inference by compact coding. *Journal of the Royal Statistical Society, Series "B"*, 49(3):240–265, 1987.

[114] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Oxford, 1989.

[115] P. J. Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1, 1988.

[116] S.D. Whitehead. *Reinforcement Learning for the adaptive control of perception and action*. PhD thesis, University of Rochester, February 1992.

[117] D. Wierstra, T. Schaul, J. Peters, and J. Schmidhuber. Fitness expectation maximization. In *Proceedings of Parallel Problem Solving from Nature (PPSN 2008)*, 2008.

[118] D. Wierstra, T. Schaul, J. Peters, and J. Schmidhuber. Natural evolution strategies. In *Congress of Evolutionary Computation (CEC 2008)*, 2008.

[119] D. Wierstra and J. Schmidhuber. Policy gradient critics. In *Proceedings of the 18th European Conference on Machine Learning (ECML 2007)*, 2007.

[120] R. J. Williams and D. Zipser. Gradient-based learning algorithms for recurrent networks and their computational complexity. In *Back-propagation: Theory, Architectures and Applications*. Hillsdale, NJ: Erlbaum, 1994.

[121] B. M. Yamauchi and R. D. Beer. Sequential behavior and learning in evolved dynamical neural networks. *Adaptive Behavior*, 2(3):219–246, 1994.

[122] Xin Yao. A review of evolutionary artificial neural networks. *International Journal of Intelligent Systems*, 4:203–222, 1993.

[123] K. Zuse. Rechnender Raum. *Elektronische Datenverarbeitung*, 8:336–344, 1967.

[124] K. Zuse. *Rechnender Raum*. Friedrich Vieweg & Sohn, Braunschweig, 1969. English translation: *Calculating Space*, MIT Technical Translation AZT-70-164-GEMIT, Massachusetts Institute of Technology (Proj. MAC), Cambridge, Mass. 02139, Feb. 1970.

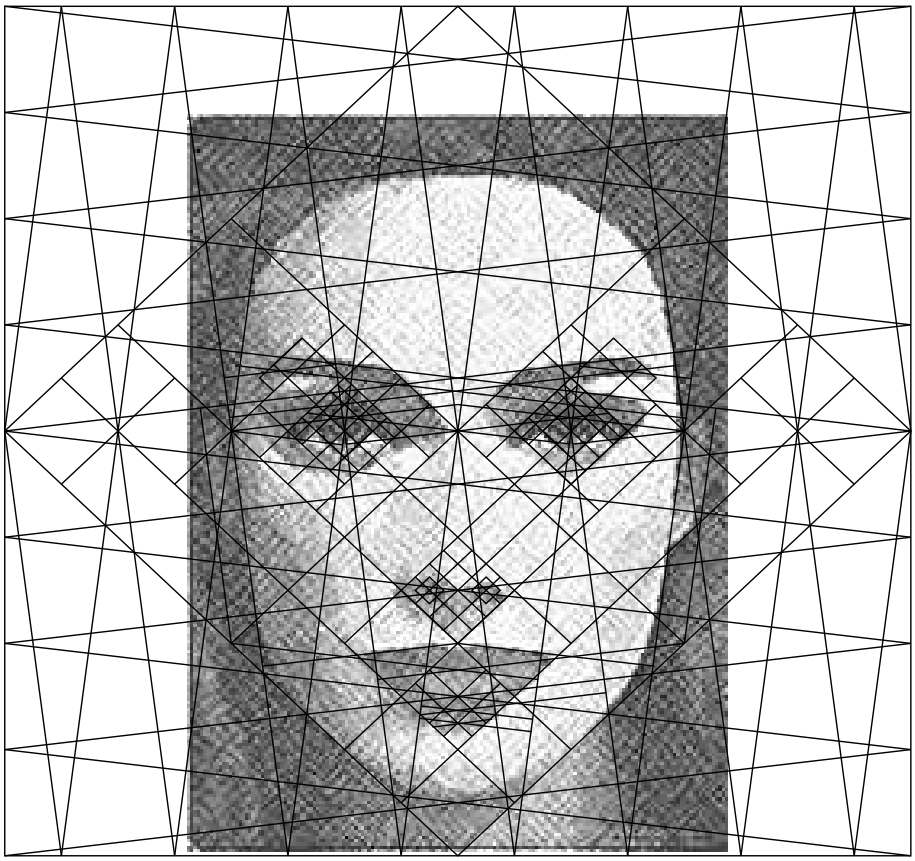

Figure 1: Previously published construction plan [69, 88] of a female face (1998). Some human observers report they feel this face is 'beautiful.' Although the drawing has lots of noisy details (texture etc) without an obvious short description, positions and shapes of the basic facial features are compactly encodable through a very simple geometrical scheme, simpler and much more precise than ancient facial proportion studies by Leonardo da Vinci and Albrecht Dürer. Hence the image contains a highly compressible algorithmic regularity or pattern describable by few bits of information. An observer can perceive it through a sequence of attentive eye movements or saccades, and consciously or subconsciously discover the compressibility of the incoming data stream. How was the picture made? First the sides of a square were partitioned into 2^4 equal intervals. Certain interval boundaries were connected to obtain three rotated, superimposed grids based on lines with slopes ± 1 or $\pm 1/2^3$ or $\pm 2^3/1$. Higher-resolution details of the grids were obtained by iteratively selecting two previously generated, neighboring, parallel lines and inserting a new one equidistant to both. Finally the grids were vertically compressed by a factor of $1 - 2^{-4}$. The resulting lines and their intersections define essential boundaries and shapes of eyebrows, eyes, lid shades, mouth, nose, and facial frame in a simple way that is obvious from the construction plan. Although this plan is simple in hindsight, it was hard to find: hundreds of my previous attempts at discovering such precise matches between simple geometries and pretty faces failed.

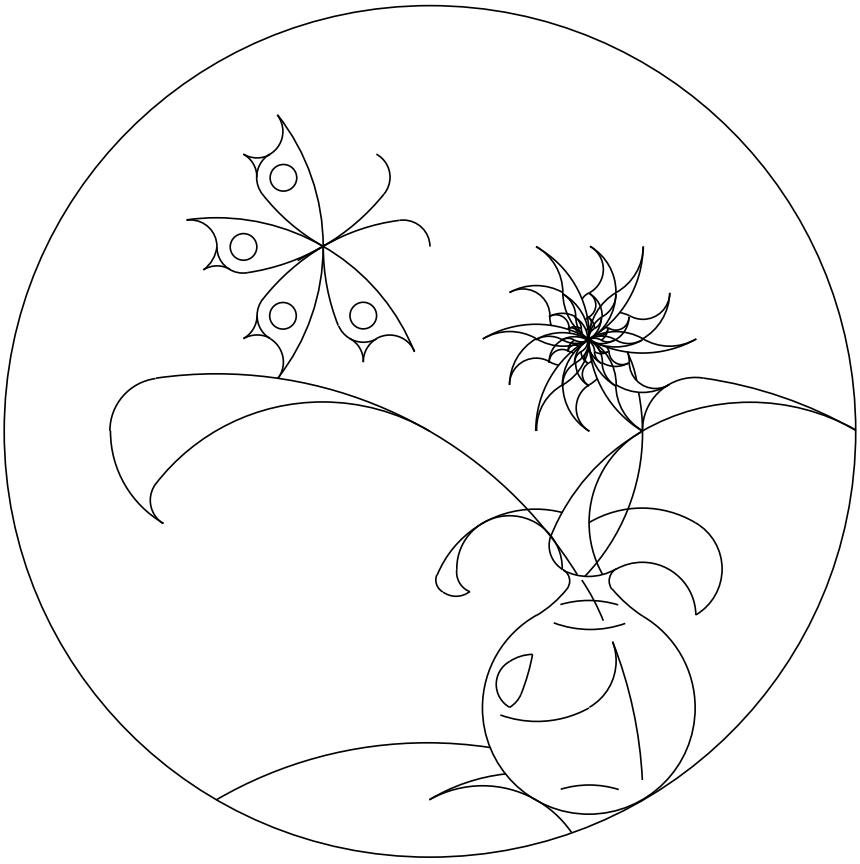

Figure 2: Image of a butterfly and a vase with a flower, reprinted from *Leonardo* [67, 81]. An explanation of how the image was constructed and why it has a very short description is given in Figure 3.

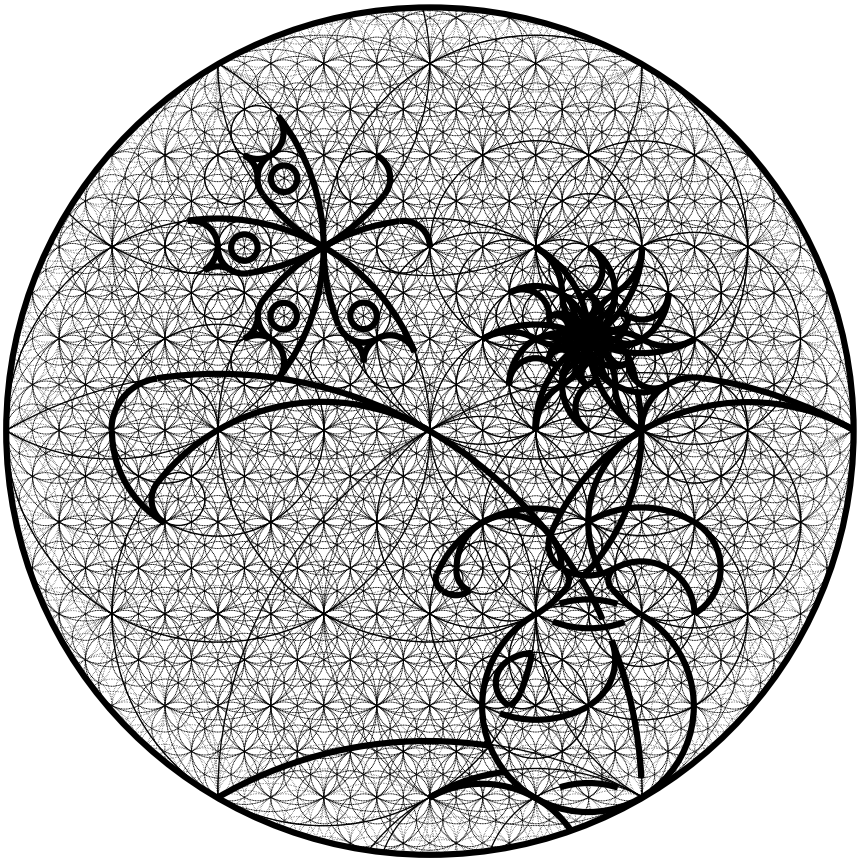

Figure 3: Explanation of how Figure 2 was constructed through a very simple algorithm exploiting fractal circles [67]. The frame is a circle; its leftmost point is the center of another circle of the same size. Wherever two circles of equal size touch or intersect are centers of two more circles with equal and half size, respectively. Each line of the drawing is a segment of some circle, its endpoints are where circles touch or intersect. There are few big circles and many small ones. In general, the smaller a circle, the more bits are needed to specify it. The drawing is simple (compressible) as it is based on few, rather large circles. Many human observers report that they derive a certain amount of pleasure from discovering this simplicity. The observer's learning process causes a reduction of the subjective complexity of the data, yielding a temporarily high derivative of subjective beauty: a temporarily steep learning curve. (Again I needed a long time to discover a satisfactory and rewarding way of using fractal circles to create a reasonable drawing.)