

The Next Step

EXPONENTIAL LIFE

Artificial Intelligence and the Arts: Toward Computational Creativity

RAMÓN LÓPEZ DE MÁNTARAS



Opening image:
Martial Raysse
America, America (1964)
Neon, painted metal
2.4 x 1.65 x 0.45 m
Centre Pompidou–Musée national
d'art moderne–Centre de création
industrielle, Paris, France.



Ramón López de Mántaras

Artificial Intelligence Research Institute (IIIA)
Spanish National Research Council (CSIC), Bellaterra, Spain

Ramón López de Mántaras is a Research Professor of the Spanish National Research Council (CSIC) and Director of the Artificial Intelligence Research Institute (IIIA). He has an MSc in Computer Science from the University of California Berkeley, USA, a PhD in Physics from the University of Toulouse, France, and a PhD in Computer Science from the Technical University of Barcelona, Spain. López de Mántaras is a pioneer of artificial intelligence (AI) in Spain and an invited plenary speaker at numerous international conferences. He is an editorial board member of several international journals, an ECCAI Fellow, and the recipient, among other awards, of the "City of Barcelona" Research Prize, the "2011 American Association of Artificial Intelligence (AAAI) Robert S. Englemore Memorial Award", the 2012 "Spanish National Computer Science Award" from the Spanish Computer Society, and the 2016 "Distinguished Service Award" of the European Association of Artificial Intelligence (EurAI). He serves on a variety of panels and advisory committees for public and private institutions based in the USA and Europe.

MORE ABOUT THE AUTHOR [+]

New technologies, and in particular artificial intelligence, are drastically changing the nature of creative processes. Computers are playing very significant roles in creative activities such as music, architecture, fine arts, and science. Indeed, the computer is already a canvas, a brush, a musical instrument, and so on. However, we believe that we must aim at more ambitious relations between computers and creativity. Rather than just seeing the computer as a tool to help human creators, we could see it as a creative entity in its own right. This view has triggered a new subfield of Artificial Intelligence called Computational Creativity. This article addresses the question of the possibility of achieving computational creativity through some examples of computer programs capable of replicating aspects of creative artistic behavior. We end with some reflections on the recent trend of democratization of creativity by means of assisting and augmenting human creativity.

INTRODUCTION

Computational creativity is the study of building software that exhibits behavior that would be deemed creative in humans. Such creative software can be used for autonomous creative tasks, such as inventing mathematical theories, writing poems, painting pictures, and composing music. However, computational creativity studies also enable us to understand human creativity and to produce programs for creative people to use, where the software acts as a creative collaborator rather than a mere tool. Historically, it has been difficult for society to come to terms with machines that purport to be intelligent and even more difficult to admit that they might be creative. Even within Computer Science, people are still skeptical about the creative potential of software. A typical statement of detractors of computational creativity is that “simulating artistic techniques means also simulating human thinking and reasoning, especially creative thinking. This is impossible to do using algorithms or information processing systems.” We could not disagree more. As is hopefully evident from the examples in this paper, creativity is not some mystical gift that is beyond scientific study but rather something that can be investigated, simulated, and harnessed for



the good of society. And while society might still be catching up, computational creativity as a discipline has come of age. This maturity is evident in the amount of activity related to computational creativity in recent years; in the sophistication of the creative software we are building; in the cultural value of the artifacts being produced by our software; and most importantly, in the consensus we are finding on general issues of computational creativity.

Computational creativity is a very lively subject area, with many issues still open to debate. For instance, many people still turn to the Turing test (Turing, 1950) to approximate the value of the artifacts produced by their software. That is, if a certain number of people cannot determine which artifacts were computer generated and which were human generated, then the software is doing well. Other people believe that the Turing test is inappropriate for creative software. One has to ask the question: “Under full disclosure, would people value the artifacts produced by a computer as highly as they would the human produced ones?” In some domains, the answer could be yes: for instance, a joke is still funny whether or not it is produced by a computer. In other domains, such as the visual arts, however, the answer is very likely to be no. This highlights the fact that the production process, and not just the outcome of it, is taken into account when assessing artworks. Hence, one could argue that such Turing-style tests are essentially setting the computers up for a fall.

Building creative software provides both a technical challenge and a social one. To proceed further, we need to embrace the fact that computers are not human. We should be loud and proud about the artifacts being produced by our software. We should celebrate the sophistication of the artificial intelligence (AI) techniques we have employed to endow the software with creative behavior. And we should help the general public to appreciate the value of these computer creations by describing the methods employed by the software to create them.

Creativity seems mysterious because when we have creative ideas it is very difficult to explain how we got them and we often talk about vague notions like “inspiration” and “intuition” when we try to explain creativity. The fact that we are not conscious of how a creative idea manifests itself does not necessarily imply that a scientific explanation cannot exist. As a matter of fact, we are not aware of how we perform other activities such as language understanding, pattern recognition, and so on, but we have better and better AI techniques able to replicate such activities.

Since nothing can arise from the emptiness, we must understand that every creative work or creative idea is always preceded by a historical-cultural scheme; it is a fruit of the cultural inheritance and the lived experiences. As Margaret Boden states in her book *Artificial Intelligence and Natural Man* (Boden, 1987):

Probably the new thoughts that originate in the mind are not completely new, because have their seeds in representations that already are in the mind. To put it differently, the germ of our culture, all our knowledge and our experience, is behind each creative idea. The greater the knowledge and the experience, the greater the possibility of finding an unthinkable relation that leads to a creative idea. If we understand creativity like the result of establishing new relations between pieces of knowledge that we already have, then the more previous knowledge one has the more capacity to be creative.



With this understanding in mind, an operational, and widely accepted, definition of creativity is: “A creative idea is a novel and valuable combination of known ideas.” In other words, physical laws, theorems, musical pieces can be generated from a finite set of existing elements and, therefore, creativity is an advanced form of problem solving that involves memory, analogy, learning, and reasoning under constraints, among others, and is therefore possible to replicate by means of computers.

This article addresses the question of the possibility of achieving computational creativity through some examples of computer programs capable of replicating some aspects of creative behavior. Due to space limitations we could not include other interesting areas of application such as: storytelling (Gervás, 2009), poetry (Montfort et al., 2014), science (Langley et al., 1987), or even humor (Ritchie, 2009). Therefore, the paper addresses, with different levels of detail, representative results of some achievements in the fields of music and visual arts. The reason for focusing on these artistic fields is that they are by far the ones in which there is more activity and where the results obtained are most impressive. The paper ends with some reflections on the recent trend of democratization of creativity by means of assisting and augmenting human creativity.

For further reading regarding computational creativity in general, I recommend the *AI Magazine special issue on Computational Creativity* (Colton et al., 2009), as well as the books by Boden (1991, 1994, 2009), Dartnall (1994), Partridge & Rowe (1994), Bentley & Corne (2002), and McCormack & d’Inverno (2012).

COMPUTATIONAL CREATIVITY IN MUSIC

Artificial intelligence has played a crucial role in the history of computer music almost since its beginnings in the 1950s. However, until quite recently, most effort had been on compositional and improvisational systems and little effort had been devoted to expressive performance. In this section we review a selection of some significant achievements in AI approaches to music composition, music performance, and improvisation, with an emphasis on the performance of expressive music.

Composing Music

Hiller and Isaacson’s (1958) work, on the ILLIAC computer, is the best-known pioneering work in computer music. Their chief result is the *Illiac Suite*, a string quartet composed following the “generate and test” problem-solving approach. The program generated notes pseudo-randomly by means of Markov chains. The generated notes were next tested by means of heuristic compositional rules of classical harmony and counterpoint. Only the notes satisfying the rules were kept. If none of the generated notes satisfied the rules, a simple backtracking procedure was used to erase the entire composition up to that point, and a new cycle was started again. The goals of Hiller and Isaacson excluded anything related to expressiveness and emotional content. In an interview (Schwanauer and Levitt, 1993, p. 21), Hiller and Isaacson said that, before addressing the expressiveness issue, simpler problems needed to be handled first. We believe that this was a very correct observation in the 1950s. After this seminal work, many other researchers based their computer compositions on Markov probability transitions



US composer
Lejaren Hiller
in the recently
inaugurated
experimental
music studio at
the University of
Illinois School
of Music, 1958.



but also with rather limited success judging from the standpoint of melodic quality.

Indeed, methods relying too heavily on Markovian processes are not informed enough to produce high-quality music consistently.

However, not all the early work on composition relies on probabilistic approaches. A good example is the work of Moorer (1972) on tonal melody generation. Moorer's program generated simple melodies, along with the underlying harmonic progressions, with simple internal repetition patterns of notes. This approach relies on simulating human composition processes using heuristic techniques rather than on Markovian probability chains. Levitt (1993) also avoided the use of probabilities in the composition process. He argues that "randomness tends to obscure rather than reveal the musical constraints needed to represent simple musical structures." His work is based on constraint-based descriptions of musical styles. He developed a description language that allows expressing musically meaningful transformations of inputs, such as chord progressions and melodic lines, through a series of constraint relationships that he calls "style templates." He applied this approach to describe a traditional jazz walking bass player simulation as well as a two-handed ragtime piano simulation.

The early systems by Hiller-Isaacson and Moorer were both based also on heuristic approaches. However, possibly the most genuine example of the early use of AI techniques is the work of Rader (1974). Rader used rule-based AI programming in his musical round (a circle canon such as "Frère Jacques") generator. The generation of the melody and the harmony were based on rules describing how notes or chords may be put together. The most interesting AI component of this system are the applicability rules, determining the applicability of the melody and chords generation rules, and the weighting rules indicating the likelihood of application of an applicable rule by means of a weight. We can already appreciate the use of meta-knowledge in this early work.

AI pioneers such as Herbert Simon or Marvin Minsky also published works relevant to computer music. Simon and Sumner (1968) describe a formal pattern language for music, as well as a pattern induction method, to discover patterns more or less implicit in musical works. One example of pattern that can be discovered is: "The opening section is in C Major, it is followed by a section in dominant and then a return to the original key." Although the program was not completed, it is worth noticing that it was one of the first in dealing with the important issue of music modelling, a subject that has been, and still is, widely studied. For example, the use of models based on generative grammars has been, and continues to be, an important and very useful approach in music modelling (Lerdahl and Jackendoff, 1983).



Marvin Minsky in his well-known paper “Music, Mind, and Meaning” (1981) addresses the important question of “how music impresses our minds.” He applies his concepts of agent and its role in a society of agents as a possible approach to shed light on that question. For example, he hints that one agent might do nothing more than notice that the music has a particular rhythm. Other agents might perceive small musical patterns, such as repetitions of a pitch; differences such as the same sequence of notes played one fifth higher, and so on. His approach also accounts for more complex relations within a musical piece by means of higher order agents capable of recognizing large sections of music. It is important to clarify that in that paper Minsky does not try to convince the reader about the question of the validity of his approach, he just hints at its plausibility.

Among the compositional systems there is a large number dealing with the problem of automatic harmonization using several AI techniques. One of the earliest works is that of Rothgeb (1969). He wrote a SNOBOL program to solve the problem of harmonizing the unfigured bass (given a sequence of bass notes infer the chords and voice leadings that accompany those bass notes) by means of a set of rules such as: “If the bass of a triad descends a semitone, then the next bass note has a sixth.” The main goal of Rothgeb was not the automatic harmonization itself but to test the computational soundness of two bass harmonization theories from the eighteenth century.

One of the most complete works on harmonization is that of Ebcioglu (1993). He developed an expert system, CHORAL, to harmonize chorales in the style of J. S. Bach. CHORAL is given a melody and produces the corresponding harmonization using heuristic rules and constraints. The system was implemented using a logic programming language designed by the author. An important aspect of this work is the use of sets of logical primitives to represent the different viewpoints of the music (chords view, time-slice view, melodic view, etc.). This was done to tackle the problem of representing large amounts of complex musical knowledge.

MUSACT (Bharucha, 1993) uses neural networks to learn a model of musical harmony. It was designed to capture musical intuitions of harmonic qualities. For example, one of the qualities of a dominant chord is to create in the listener the expectancy that the tonic chord is about to be heard. The greater the expectancy, the greater the feeling of consonance of the tonic chord. Composers may choose to satisfy or violate these expectancies to varying degree. MUSACT is capable of learning such qualities and generating graded expectancies in a given harmonic context.

In HARMONET (Feulner, 1993), the harmonization problem is approached using a combination of neural networks and constraint satisfaction techniques. The neural network learns what is known as harmonic functionality of the chords (chords can play the function



of tonic, dominant, subdominant, etc.) and constraints are used to fill the inner voices of the chords. The work on HARMONET was extended in the MELONET system (Hörnelt and Degenhardt, 1997; Hörnelt and Menzel, 1998). MELONET uses a neural network to learn and reproduce a higher level structure in melodic sequences. Given a melody, the system invents a Baroque-style harmonization and variation of any chorale voice. According to the authors, HARMONET and MELONET together form a powerful music-composition system that generates variations whose quality is similar to those of an experienced human organist.

Pachet and Roy (1998) also used constraint satisfaction techniques for harmonization. These techniques exploit the fact that both the melody and the harmonization knowledge impose constraints on the possible chords. Efficiency is, however, a problem with purely constraint satisfaction approaches.

In Sabater et al. (1998), the problem of harmonization is approached using a combination of rules and case-based reasoning. This approach is based on the observation that purely rule-based harmonization usually fails because, in general, “the rules do not make the music, it is the music that makes the rules.” Then, instead of relying only on a set of imperfect rules, why not make use of the source of the rules, that is, the compositions themselves? Case-based reasoning allows the use of examples of already harmonized compositions as cases for new harmonizations. The system harmonizes a given melody by first looking for similar, already harmonized, cases; when this fails, it looks for applicable general rules of harmony. If no rule is applicable, the system fails and backtracks to the previous decision point. The experiments have shown that the combination of rules and cases results in much fewer failures in finding an appropriate harmonization than using either technique alone. Another advantage of the case-based approach is that each newly correctly harmonized piece can be memorized and made available as a new example to harmonize other melodies; that is, a learning-by-experience process takes place. Indeed, the more examples the system has, the less often the system needs to resort to the rules and therefore it fails less. MUSE (Schwanauer, 1993) is also a learning system that extends an initially small set of voice leading constraints by learning a set of rules of voice doubling and voice leading. It learns by reordering the rules agenda and by chunking the rules that satisfy the set of voice leading constraints. MUSE successfully learned some of the standard rules of voice leading included in traditional books of tonal music.

Morales-Manzanares et al. (2001) developed a system called SICIB capable of composing music using body movements. This system uses data from sensors attached to the dancer and applies inference rules to couple the gestures with the music in real time.

Certainly the best-known work on computer composition using AI is David Cope’s EMI project (Cope, 1987, 1990). This work focuses on the emulation of styles of various composers. It has successfully composed music in the styles of Cope, Mozart, Palestrina, Albinoni, Brahms, Debussy, Bach, Rachmaninoff, Chopin, Stravinsky, and Bartok. It works by searching for recurrent patterns in several (at least two) works of a given composer. The discovered patterns are called signatures. Since signatures are location dependent, EMI uses one of the composer’s works as a guide to fix them to their appropriate locations when composing a new piece. To compose the musical motives between signatures, EMI uses a compositional rule analyzer to discover the constraints used by the composer in his works. This analyzer counts musical events such as voice leading directions, use of repeated notes, and so on, and represents them as a statistical model of the analyzed works. The program follows this model to compose the motives to be inserted in the



empty spaces between signatures. To properly insert them, EMI has to deal with problems such as: linking initial and concluding parts of the signatures to the surrounding motives avoiding stylistic anomalies; maintaining voice motions; maintaining notes within a range, and so on. Proper insertion is achieved by means of an **Augmented Transition Network** (Woods, 1970). The results, although not perfect, are quite consistent with the style of the composer.

Synthesizing Expressive Music

One of the main limitations of computer-generated music has been its **lack of expressiveness**, that is, lack of **gesture**. Gesture is what musicians call the nuances of performance that are uniquely and subtly interpretive or, in other words, creative.

One of the first attempts to address expressiveness in music is that of Johnson (1992). She developed an expert system to determine the tempo and the articulation to be applied when playing Bach's fugues from "The Well-Tempered Clavier." The rules were obtained from two expert human performers. The output gives the base tempo value and a list of performance instructions on notes duration and articulation that should be followed by a human player. The results very much coincide with the instructions given in well-known commented editions of "The Well-Tempered Clavier." The main limitation of this system is its lack of generality because it only works well for fugues written on a 4/4 meter. For different meters, the rules should be different. Another obvious consequence of this **lack of generality** is that the rules are only applicable to Bach fugues.

The work of the **KTH** group from Stockholm (Friberg, 1995; Friberg et al., 1998, 2000; Bresin, 2001) is one of the best-known long-term efforts on performance systems. Their current **Director Musices** system incorporates rules for tempo, dynamic, and articulation transformations constrained to MIDI. These rules are inferred both from theoretical musical knowledge and experimentally by training, specially using the so-called analysis-by-synthesis approach. The rules are divided in three main classes: differentiation rules, which enhance the differences between scale tones; grouping rules, which show what tones belong together; and ensemble rules, which synchronize the various voices in an ensemble.

Canazza et al. (1997) developed a system to analyze how the musician's expressive intentions are reflected in the performance. The analysis reveals two different expressive dimensions: one related to the energy (dynamics) and the other related to the kinetics (rubato) of the piece. The authors also developed a program for generating expressive performances according to these two dimensions.

The work of **Dannenberg** and Derenyi (1998) is also a good example of articulation transformations using manually constructed rules. They developed a trumpet synthesizer that combines a physical model with a performance model. The goal of the performance model is to generate control information for the physical model by means of a collection of rules manually extracted from the analysis of a collection of controlled recordings of human performance.

Another approach taken for performing tempo and dynamics transformation is the use of neural network techniques. In Bresin (1998), a **system that combines symbolic decision rules with neural networks is implemented for simulating the style of real piano performers**. The outputs of the neural networks express time and loudness deviations. These neural networks extend the standard feed-forward network trained with the back propagation algorithm with feedback connections from the output neurons to the input neurons.





We can see that, except for the work of the KTH group that considers three expressive resources, the other systems are limited to two resources, such as rubato and dynamics, or rubato and articulation. This limitation has to do with the use of rules. Indeed, the main problem with the rule-based approaches is that it is very difficult to find rules general enough to capture the variety present in different performances of the same piece by the same musician and even the variety within a single performance (Kendall and Carterette, 1990). Furthermore, the different expressive resources interact with each other. That is, the rules for dynamics alone change when rubato is also taken into account. Obviously, due to this interdependency, the more expressive resources one tries to model, the more difficult it is to find the appropriate rules.

We developed a case-based reasoning system called **SaxEx** (Arcos et al., 1998), a computer program capable of synthesizing high-quality expressive tenor sax solo performances of jazz ballads, based on cases representing human solo performances. As mentioned above, previous rule-based approaches to that problem could not deal with more than two expressive parameters (such as dynamics and rubato) because it is too difficult to find rules general enough to capture the variety present in expressive performances. Besides, the different expressive parameters interact with each other making it even more difficult to find appropriate rules taking into account these interactions.

With case-based reasoning, we have shown that it is possible to deal with the five most important expressive parameters: dynamics, rubato, vibrato, articulation, and attack of the notes. To do so, SaxEx uses a case memory containing examples of human performances, analyzed by means of spectral modelling techniques and background musical knowledge. The score of the piece to be performed is also provided to the system. The core of the method is to analyze each input note determining (by means of the background musical knowledge) its role in the musical phrase it belongs to, identify and retrieve (from the case base of human performances) notes with similar roles, and, finally, transform the input note so that its expressive properties (dynamics, rubato, vibrato, articulation, and attack) match those of the most similar retrieved note. Each note in the case base is annotated with its role in the musical phrase it belongs to, as well as with its expressive values. Furthermore, cases do not contain just information on each single note but they include contextual knowledge at the phrase level. Therefore, cases in this system have a complex object-centered representation.



Although limited to monophonic performances, the results are very convincing and demonstrate that case-based reasoning is a very powerful methodology to directly use the knowledge of a human performer that is implicit in her playing examples rather than trying to make this knowledge explicit by means of rules. Some audio results can be listened to at: <http://www.iiia.csic.es/%7Earcos/noos/Demos/Example.html>. More recent papers (Arcos and López de Mántaras, 2001; López de Mántaras and Arcos, 2002, López de Mántaras and Arcos, 2012), describe this system in great detail.

Based on the work on SaxEx, we developed TempoExpress (Grachten et al., 2004), a case-based reasoning system for applying musically acceptable tempo transformations to monophonic audio recordings of musical performances. TempoExpress has a rich description of the musical expressivity of the performances, which includes not only timing deviations of performed score notes, but also represents more rigorous kinds of expressivity, such as note ornamentation, consolidation, and fragmentation. Within the tempo transformation process, the expressivity of the performance is adjusted in such a way that the result sounds natural for the new tempo. A case base of previously performed melodies is used to infer the appropriate expressivity. The problem of changing the tempo of a musical performance is not as trivial as it may seem because it involves a lot of musical knowledge and creative thinking. Indeed, when a musician performs a musical piece at different tempos the performances are not just time-scaled versions of each other (as if the same performance were played back at different speeds). Together with the changes of tempo, variations in musical expression are made (Desain and Honing, 1993). Such variations do not only affect the timing of the notes, but can also involve, for example, the addition or deletion of ornamentations, or the consolidation/fragmentation of notes. Apart from the tempo, other domain-specific factors seem to play an important role in the way a melody is performed, such as meter, and phrase structure. Tempo transformation is one of the audio post-processing tasks manually done in audio labs. Automatizing this process may, therefore, be of industrial interest.

Other applications of case-based reasoning to expressive music are those of Suzuki et al. (1999), and those of Tobudic and Widmer (2003, 2004). Suzuki et al. (1999) use examples cases of expressive performances to generate multiple performances of a given piece with varying musical expression; however, they deal only with two expressive parameters. Tobudic and Widmer (2003) apply Instance-Based Learning also to the problem of generating expressive performances. Their approach is used to complement a note-level rule-based model with some predictive capability at the higher level of musical phrasing. More concretely, the Instance-Based Learning component recognizes performance patterns, of

a concert pianist, at the phrase level and learns how to apply them to new pieces by analogy. The approach produced some interesting results but, as the authors recognize, was not very convincing due to the limitation of using an attribute-value representation for the phrases. Such simple representation cannot take into account relevant structural information of the piece, both at the sub-phrase level and at the inter-phrase level. In a subsequent paper, Tobudic and Widmer (2004) succeeded in partly overcoming this limitation by using a relational phrase representation.

Widmer et al. (2009) describe a computer program that learns to expressively perform classical piano music. The approach is data intensive and based on statistical learning. Performing music expressively certainly requires high levels of creativity, but the authors take a very pragmatic view to the question of whether their program can be said to be creative or not and claim that “creativity is in the eye of the beholder.” In fact, the main goal of the authors is to investigate and better understand music performance as a creative human behavior by means of AI methods.

The possibility for a computer to play expressively is a fundamental component of the so-called *hyper-instruments*. These are instruments designed to augment an instrument sound with such idiosyncratic nuances as to give it human expressiveness and a rich, live sound. To make a hyper-instrument, take a traditional instrument, like for example a cello, and connect it to a computer through electronic sensors in the neck and in the bow; also equip the hand that holds the bow with sensors; and program the computer with a system similar to SaxEx that allows analysis of the way the human interprets the piece, based on the score, on musical knowledge, and on the readings of the sensors. The results of such analysis allows the hyper-instrument to play an active role, altering aspects such as timbre, tone, rhythm, and phrasing, as well as generating an accompanying voice. In other words, you have got an instrument that can be its own intelligent accompanist. Tod Machover, from MIT's Media Lab, developed such a hyper-cello and the great cello player Yo-Yo Ma premiered, playing the hyper-cello, a piece, composed by Tod Machover, called “Begin Again Again...” at the Tanglewood Festival several years ago.

Improvising Music

Music improvisation is a very complex creative process that has also been computationally modelled. It is often referred to as “composition on the fly” and, therefore, it is, creatively speaking, more complex than composition and is, probably, the most complex of the three music activities surveyed here. An early work on computer improvisation is the *Flavors Band* system of Fry (1984). Flavors Band is a procedural language, embedded in LISP, for specifying





“Man projects the autonomy of his awareness, his power to control, his own individuality and the idea of his persona onto automated objects.”

JEAN BAUDRILLARD (1929–2007)
French philosopher and sociologist. *The System of Objects* (1969).

Ryoji Ikeda
Sound installation (2009)
5 Meyer SB-1s, computer.
12 × 6 × 24 m
Tokyo Museum of Contemporary Art.



jazz and popular music styles. Its **procedural representation** allows the generation of scores in a pre-specified style by making changes to a score specification given as input. It allows combining random functions and musical constraints (chords, modes, etc.) to generate improvisational variations. The most remarkable result of Flavors Band was an interesting arrangement of the bass line, and an improvised solo, of John Coltrane's composition "Giant Steps."

GenJam (Biles, 1994) builds a model of a jazz musician learning to improvise by means of a **genetic algorithm**. A human listener plays the role of fitness function by rating the offspring improvisations. Papadopoulos and Wiggins (1998) also used a **genetic algorithm to improvise jazz melodies on a given chord progression**. Contrarily to GenJam, the program includes a fitness function that automatically evaluates the quality of the offspring improvisations rating eight different aspects of the improvised melody such as the melodic contour, notes duration, intervallic distances between notes, and so on.

Franklin (2001) uses **recurrent neural networks to learn how to improvise jazz solos from transcriptions of solo improvisations by jazz saxophonist Sonny Rollins**. A reinforcement learning algorithm is used to refine the behavior of the neural network. The reward function rates the system solos in terms of jazz harmony criteria and according to Rollins's style.

The lack of interactivity, with a human improviser, of the above approaches has been criticized (Thom, 2001) on the grounds that they remove the musician from the physical and spontaneous creation of a melody. Although it is true that the most fundamental characteristic of improvisation is the spontaneous, real-time creation of a melody, it is also true that interactivity was not intended in these approaches and nevertheless they could generate very interesting improvisations. Thom (2001) with her **Band-out-of-a-Box (BoB) system addresses the problem of real-time interactive improvisation between BoB and a human player. In other words, BoB is a "music companion" for real-time improvisation. Thom's approach follows Johnson-Laird's (1991) psychological theory of jazz improvisation. This theory opposes the view that improvising consists of rearranging and transforming pre-memorized "licks" under the constraints of a harmony. Instead Johnson-Laird proposes a stochastic model based on a greedy search over a constrained space of possible notes to play at a given point in time. The very important contribution of Thom is that her system learns these constraints, and therefore**

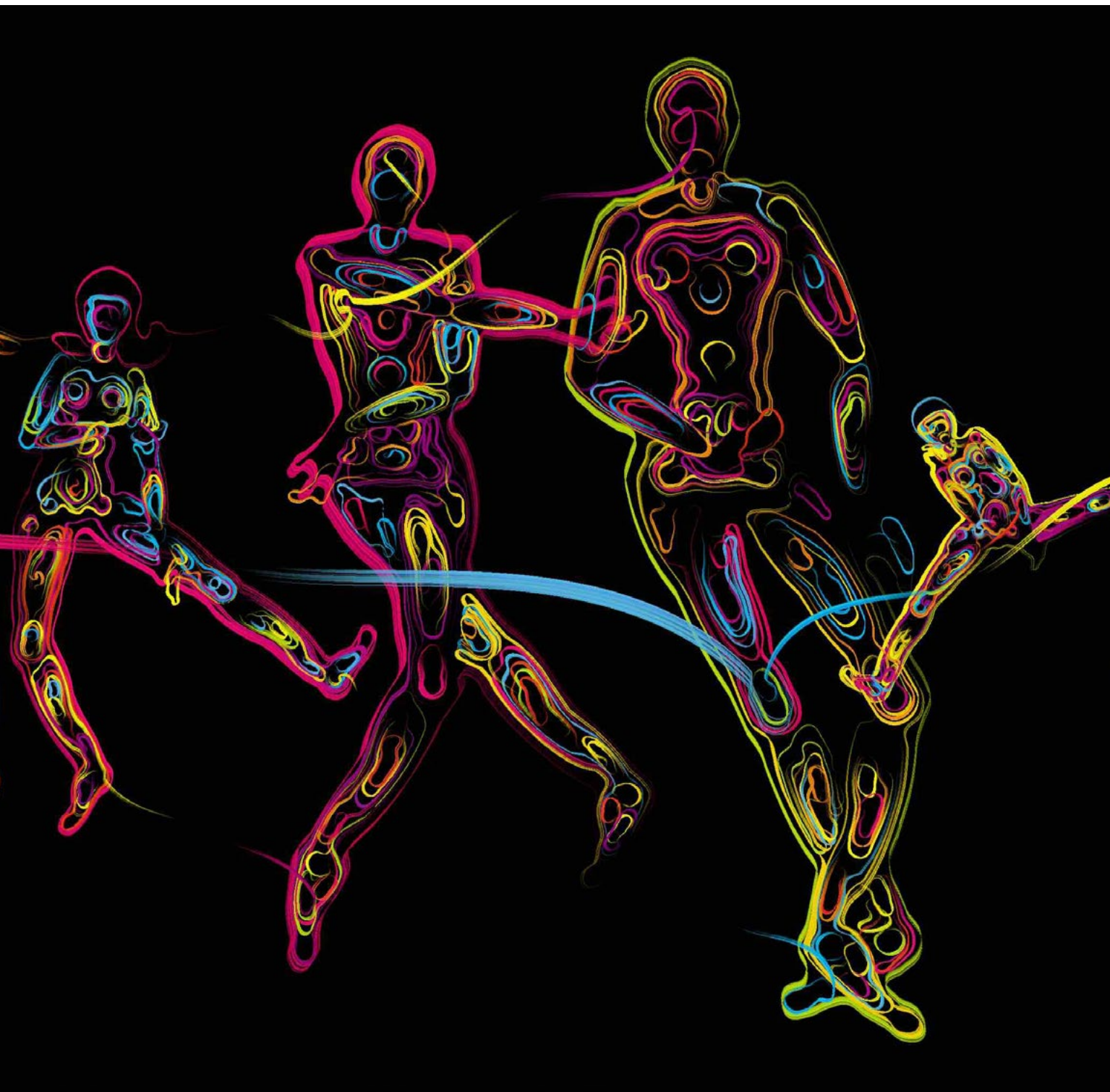
the stochastic model, from the human player by means of an unsupervised probabilistic clustering algorithm. The learned model is used to abstract solos into user-specific playing modes. The parameters of that learned model are then incorporated into a stochastic process that generates the solos in response to four bar solos of the human improviser. BoB has been very successfully evaluated by testing its real-time solo tradings in two different styles, that of saxophonist Charlie Parker, and that of violinist Stephane Grapelli.

Another remarkable interactive improvisation system was developed by Dannenberg (1993). The difference with Thom's approach is that in Dannenberg's system, music generation is mainly driven by the composer's goals rather than the performer's goals. Wessel's (1998) interactive improvisation system is closer to Thom's in that it also emphasizes the accompaniment and enhancement of live improvisations.

COMPUTATIONAL CREATIVITY IN VISUAL ARTS

AARON is a robotic system, developed over many years by the artist and programmer Harold Cohen (Cohen, 1995), that can pick up a paintbrush with its robotic arm and paint on canvas on its own. It draws people in a botanical garden not just making a copy of an existing drawing but generating as many unique drawings on this theme as may be required of it. AARON has never seen a person or walked through a botanical garden but has been given knowledge about body postures and plants by means of rules. AARON's knowledge and the way AARON uses its knowledge are not like the knowledge that we, humans, have and use because human knowledge is based on experiencing the world, and people experience the world with their bodies, their brains, and their reproductive systems, which computers do not have. However, just like humans, AARON'S knowledge has been acquired cumulatively. Once it understands the concept of a leaf cluster, for example, it can make use of that knowledge whenever it needs it. Plants exist for AARON in terms of their size, the thickness of limbs with respect to height, the rate at which limbs get thinner with respect to spreading, the degree of branching, the angular spread where branching occurs, and so on. Similar principles hold for the formation of leaves and leaf clusters. By manipulating these factors, AARON is able to generate a wide range of plant types and will never draw quite the same plant twice, even when it draws a number of plants recognizably of the same type. Besides, AARON must know what the human body consists of, what the different parts are, and how big they are in relation to each other. Then it has to know how the parts of the body are articulated and what are the types and ranges of movements at each joint. Finally,





because a coherently moving body is not merely a collection of independently moving parts, AARON has to know something about how body movements are coordinated: what the body has to do to keep its balance, for example. Conceptually, this is not as difficult as it may seem, at least for standing positions with one or both feet on the ground. It is just a matter of keeping the center of gravity over the base and, where necessary, using the arms for achieving balanced positions. It also has knowledge about occlusions so that a partially occluded human body might have, for example, just one arm and/or one leg visible but AARON knows that normal people have two arms and two legs and therefore when not occluded it will always draw two limbs. This means that AARON cannot “break” rules and will never “imagine” the possibility of drawing humans with one leg, for example, or other forms of abstraction. In that sense, AARON’s creativity is limited and very far from a human one. Nevertheless AARON’s paintings have been exhibited in London’s Tate Modern and the San Francisco Museum of Modern Art. In some respects, then, AARON passes some kind of creative Turing test for its works are good enough to be exhibited alongside some of the best human artists.

Simon Colton’s Painting Fool (Colton et al., 2015) is much more autonomous than AARON. Although the software does not physically apply paint to canvas, it simulates many styles digitally, from collage to paint strokes. In Colton’s words:

The Painting Fool only needs minimal direction and can come up with its own concepts by going online for source material. The software runs its own web searches and crawls through social media websites. The idea is that this approach will let it produce art that is meaningful to the audience, because it is essentially drawing on the human experience as we act, feel and argue on the web.

For instance, in 2009, the Painting Fool produced its own interpretation of the war in Afghanistan, based on a news story. The result is a juxtaposition of Afghan citizens, explosions, and war graves.

Other examples of computational creativity applied to painting and other visual arts are the works of Karl Sims and of Jon McCormack. Karl Sims’s *Reaction-Diffusion Media Wall* (Sims, 2016) is based on the interactive simulation of chemicals that react and diffuse to create emergent dynamic patterns according to the reaction-diffusion equations governing biological morphogenesis. This work is exhibited at the Museum of Science in Boston. Previous works of Karl Sims include the application of evolutionary computations techniques to interactively evolved images in his Genetic Images system (Sims, 1994).

Jon McCormack also looks at how biological processes could be successfully applied to creative systems in his “Design After Nature Project” (McCormack, 2014). In another project, “Creative Ecosystems,” he looks at concepts and metaphors from biological ecosystems (McCormack and d’Inverno, 2012) as a means to enhance human creativity in the digital arts.

There are numerous other examples related to the visual arts. The reported ones are not just a representative set but, in my opinion, also the most important contributions to this field.





SUPPORTING AND AUGMENTING HUMAN CREATIVITY: THE DEMOCRATIZATION OF CREATIVITY

Can we use artificial intelligence to support human creativity and discovery? A new trend known as Assisted Creation has important implications for creativity: on the one hand, assistive creation systems are making a wide range of creative skills more accessible. On the other hand, collaborative platforms, such as the one developed within the European project PRAISE for learning music (Yee-King and d’Inverno, 2014), are making it easier to learn new creative skills. PRAISE is a social network-based learning platform that includes humans and intelligent software agents that give feedback to a music student regarding music composition, arrangement, and performance. Students upload their solutions to a given lesson plan provided by a tutor (compositions, arrangements, or performances). Then the intelligent agents, as well as other fellow students and tutors, analyze these solutions and provide feedback. For instance, in the case of a composition the agent might say: “Your modulation sounds pretty good but you could try modulating everything up a major third for bars 5 to 8.”

In the case of performances, other intelligent software agents compare those of the students with one previously recorded by the tutor when she uploaded the lesson plan. A camera captures the gesture of the student and the software agents also provide feedback about possibly incorrect postures. Tools like this one that accelerate the skill acquisition time lead to a phenomenon known as “the democratization of creativity.”

As early as 1962, Douglas Engelbart (Engelbart, 1962) wrote about a “writing machine that would permit you to use a new process of composing text [...] You can integrate your new ideas more easily, and thus harness your creativity more continuously.” Engelbart’s vision was not only about augmenting individual creativity. He also wanted to augment the collective intelligence and creativity of groups by improving collaboration and group problem-solving ability. A basic idea is that creativity is a social process that can be augmented through technology. By projecting these ideas into the future, we could imagine a world where creativity is highly accessible and (almost) anyone can write at the level of the best writers, paint like the great masters, compose high-quality music, and even discover new forms of creative expression. For a person who does not have a particular creative skill, gaining a new capability through assisted creation systems is highly empowering.

Although the above futuristic scenario is currently pure fiction, there already exist several examples of assisted creation. One of the most interesting is the assisted drumming system developed at Georgia Institute of Technology (Bretan and Weinberg, 2016). It consists of a wearable robotic limb that allows drummers to play with three arms. The 61-centimeter-long (two-foot) “smart arm” can be attached to a musician’s shoulder. It responds to human gestures and the music it hears. When the drummer plays the high hat cymbal, for example, the robotic arm maneuvers to play the ride cymbal. When the drummer switches to the snare, the mechanical arm shifts to the tom.

Another very interesting result in assisted creativity is the music style and harmony transfer, genre to genre, developed at the SONY Computer Science Lab in Paris (Martin et al., 2015; Papadopoulos et al., 2016) that assists composers in harmonizing a music piece in a genre according to the style of another completely different genre. For instance harmonizing a jazz standard in the style of Mozart.

CONCLUDING REMARKS: APPARENTLY OR REALLY CREATIVE?

Margaret Boden pointed out that even if an artificially intelligent computer would be as creative as Bach or Einstein, for many it would be just apparently creative but not really creative. I fully agree with her for two main reasons, which are: the lack of intentionality and our reluctance to give a place in our society to artificially intelligent agents. The lack of intentionality is a direct consequence of Searle’s Chinese Room argument (Searle, 1980), which states that computer programs can only perform syntactic manipulation of symbols but are unable to give them any semantics. It is generally admitted that intentionality can be explained in terms of causal relations. However, it is also true that existing computer programs lack too many relevant causal connections to exhibit intentionality but perhaps future, possibly anthropomorphic, “embodied” artificial intelligences, that is, agents equipped not only with sophisticated software but also with different types of advanced sensors allowing them to interact with the environment, may have enough causal connections to give meaning to symbols and have intentionality.

Regarding social rejection, the reasons why we are so reluctant to accept that nonbiological agents can be creative (even biological ones, as was the case with “Nonja,” a twenty-year-old painter from Vienna, whose abstract paintings had been exhibited and appreciated in art galleries but as soon as it was known that she was an orangutan from the Vienna Zoo her





work was much less appreciated!) is that they do not have a natural place in our society of human beings and a decision to accept them would have important social implications. It is, therefore, much simpler to say that they appear to be intelligent, creative, and so on, instead of saying that they are. In a word, it is a moral not a scientific issue. A third reason for denying creativity to computer programs is that they are not conscious of their accomplishments. It is true that machines do not have consciousness, and possibly will never have conscious thinking; however, the lack of consciousness is not a fundamental reason to deny the potential for creativity or even the potential for intelligence. After all, computers would not be the first example of unconscious creators; evolution is the first example, as Stephen Jay Gould (1996) brilliantly points out: “If creation demands a visionary creator, then how does blind evolution manage to build such splendid new things as ourselves?”

ACKNOWLEDGMENTS

This research has been partially supported by the 2014-SGR-118 grant from the Generalitat de Catalunya.



SELECT BIBLIOGRAPHY

- Arcos, J. L., and López de Mántaras, R. 2001. "An interactive case-based reasoning approach for generating expressive music." *Applied Intelligence* 14(1): 115–129.
- Arcos, J. L., López de Mántaras, R., and Serra, X. 1998. "Saxex: A case-based reasoning system for generating expressive musical performances." *Journal of New Music Research* 27(3): 194–210.
- Bentley, P. J., and Corne, D. W. (eds.). 2001. *Creative Evolutionary Systems*. Burlington, MA: Morgan Kaufmann.
- Bharucha, J. 1993. "MUSACT: A connectionist model of musical harmony." In *Machine Models of Music*, S. M. Schwanauer and D. A. Levitt (eds.). Cambridge, MA: The MIT Press, 497–509.
- Biles, J. A. 1994. "GenJam: A genetic algorithm for generating jazz solos." In *Proceedings of the 1994 International Computer Music Conference*. San Francisco: International Computer Music Association.
- Boden, M. 1987. *Artificial Intelligence and Natural Man*. New York: Basic Books.
- Boden, M. 1991. *The Creative Mind: Myths and Mechanisms*. New York: Basic Books.
- Boden, M. (ed.) 1994. *Dimensions of Creativity*. Cambridge, MA: The MIT Press.
- Boden, M. 2009. "Computers models of creativity." *AI Magazine* 30(3): 23–34.
- Bresin, R. 1998. "Artificial neural networks based models for automatic performance of musical scores." *Journal of New Music Research* 27(3): 239–270.
- Bresin, R. 2001. "Articulation rules for automatic music performance." In *Proceedings of the 2001 International Computer Music Conference*. San Francisco: International Computer Music Association.
- Bretan, M., and Weinberg, G. 2016. "A survey of robotic musicianship." *Commun. ACM* 59(5): 100–109.
- Canazza, S., De Poli, G., Roda, A., and Vidolin, A. 1997. "Analysis and synthesis of expressive intention in a clarinet performance." In *Proceedings of the 1997 International Computer Music Conference*. San Francisco: International Computer Music Association, 113–120.
- Cohen, H. 1995. "The further exploits of Aaron, painter." *Stanford Humanities Review* 4(2): 141–158.
- Colton, S., López de Mántaras, R., and Stock, O. 2009. "Computational creativity: Coming of age." Special issue of *AI Magazine* 30(3): 11–14.
- Colton, S., Halskov, J., Ventura, D., Gouldstone, I., Cook, M., and Pérez-Ferrer, B. 2015. "The Painting Fool sees! New projects with the automated painter." *International Conference on Computational Creativity 2015*: 189–196.
- Cope, D. 1987. "Experiments in music intelligence." In *Proceedings of the 1987 International Computer Music Conference*. San Francisco: International Computer Music Association.
- Cope, D. 1990. "Pattern matching as an engine for the computer simulation of musical style." In *Proceedings of the 1990 International Computer Music Conference*. San Francisco: International Computer Music Association.
- Dannenberg, R. B. 1993. "Software design for interactive multimedia performance." *Interface* 22(3) 213–218.
- Dannenberg, R. B., and Derenyi, I. 1998. "Combining instrument and performance models for high quality music synthesis." *Journal of New Music Research* 27(3) 211–238.
- Dartnall, T. (ed.). 1994. *Artificial Intelligence and Creativity*. Dordrecht: Kluwer Academic Publishers.
- Desain, P., and Honing, H. 1993. "Tempo curves considered harmful." In *Time in Contemporary Musical Thought*, J. D. Kramer (ed.). *Contemporary Music Review* 7(2).
- Ebcioglu, K. 1993. "An expert system for harmonizing four-part chorales." In *Machine Models of Music*, S. M. Schwanauer and D. A. Levitt (eds.). Cambridge, MA: The MIT Press, 385–401.
- Engelbart, D. C. 1962. *Augmenting Human Intellect: A Conceptual Framework*. Menlo Park, CA: Stanford Research Institute, 5.
- Feulner, J. 1993. "Neural networks that learn and reproduce various styles of harmonization." In *Proceedings of the 1993 International Computer Music Conference*. San Francisco: International Computer Music Association.
- Franklin, J. A. 2001. "Multi-phase learning for jazz improvisation and interaction." In *Proceedings of the Eighth Biennial Symposium on Art and Technology*. New London, CT: Center for Arts and Technology, Connecticut College.
- Friberg, A. 1995. "A quantitative rule system for musical performance." (Unpublished PhD dissertation). KTH, Stockholm.
- Friberg, A., Bresin, R., Fryden, L., and Sunberg, J. 1998. "Musical punctuation on the microlevel: automatic identification and performance of small melodic units." *Journal of New Music Research* 27(3): 271–292.
- Friberg, A., Sunberg, J., and Fryden, L. 2000. "Music From motion: sound level envelopes of tones expressing human locomotion." *Journal of New Music Research* 29(3): 199–210.
- Fry, C. 1984. "Flavors Band: a language for specifying musical style." Reprinted in *Machine Models of Music*, S. M. Schwanauer and D. A. Levitt (eds.). Cambridge, MA: The MIT Press, 1993, 427–451.
- Gervás, P. 2009. "Computational approaches to storytelling and creativity." *AI Magazine* 30(3): 49–62.
- Grachten, M., Arcos, J. L., and López de Mántaras, R. 2004. "TempoExpress, a CBR approach to musical tempo transformations." In *Proceedings of the 7th European Conference on Case-Based Reasoning*, P. Funk and P. A. Gonzalez Calero (eds.). *Lecture Notes in Artificial Intelligence*, vol. 3155. Heidelberg: Springer, 601–615.
- Gould, S. J. 1996. "Creating the creators." *Discover Magazine* October: 42–54.



- Hiller, L., and Isaacson, L. 1958. "Musical composition with a high-speed digital computer." Reprinted in *Machine Models of Music*, S. M. Schwanauer and D. A. Levitt (eds.). Cambridge, MA: The MIT Press, 1993, 9–21.
- Hörnel, D., and Degenhardt, P. 1997. "A neural organist improvising Baroque-style melodic variations." In *Proceedings of the 1997 International Computer Music Conference*. San Francisco: International Computer Music Association, 430–433.
- Hörnel, D., and Menzel, W. 1998. "Learning musical structure and style with neural networks." *Journal of New Music Research* 22(4): 44–62.
- Johnson, M. L. 1992. "An expert system for the articulation of Bach fugue melodies." In *Readings in Computer Generated Music*, D. L. Baggi (ed.). Los Alamitos, CA: IEEE Press, 41–51.
- Johnson-Laird, P. N. 1991. "Jazz improvisation: A theory at the computational level." In *Representing Musical Structure*, P. Howell, R. West, and I. Cross (eds.). London: Academic Press.
- Kendall, R. A., and Carterette, E. C. 1990. "The communication of musical expression." *Music Perception* 8(2): 129.
- Langley, P., Simon, H. A., Bradshaw, G. L., and Zytkow, J. M. 1987. *Scientific Discovery, Computational Explorations of the Creative Mind*. Cambridge, MA: The MIT Press.
- Lerdahl, F., and Jackendoff, R. 1983. "An overview of hierarchical structure in music." *Music Perception* 1: 229–252.
- Levitt, D. A. 1993. "A representation for musical dialects." In *Machine Models of Music*, S. M. Schwanauer and D. A. Levitt (eds.). Cambridge, MA: The MIT Press, 455–469.
- López de Mántaras, R., and Arcos, J. L. 2002. "AI and music: From composition to expressive performance." *AI Magazine* 23(3): 43–57.
- López de Mántaras, and R., Arcos J. L. 2012. "Playing with cases: Rendering expressive music with case-based reasoning." *AI Magazine* 33(4): 22–31.
- Martín, D., Frantz, B., and Pachet, F. 2015. "Improving music composition through peer feedback: Experiment and preliminary results." In *Music Learning with Massive Open Online Courses (MOOCs)*, Luc Steels (ed.). Amsterdam: IOS Press, 195–204.
- McCormack, J. 2014. "Balancing act: variation and utility in evolutionary art." In *Evolutionary and Biologically Inspired Music, Sound, Art and Design. Lecture Notes in Computer Science*, Vol. 8601. Heidelberg: Springer, 26–37.
- McCormack, J., and d'Inverno, M. 2012. *Computers and Creativity*. Heidelberg: Springer.
- Minsky, M. 1981. "Music, mind, and meaning." Reprinted in *Machine Models of Music*, S. M. Schwanauer and D. A. Levitt (eds.). Cambridge, MA: The MIT Press, 1993, 327–354.
- Nick Montfort, Patsy Baudoin, John Bell, Ian Bogost, Jeremy Douglass, Mark C. Marino, Michael Mateas, Casey Reas, Mark Sample, and Noah Vawter. 2014. 10PRINT CHR\$(205.5+RND(1)); GOTO 10. Cambridge, MA: The MIT Press.
- Moorer, J. A. 1972. "Music and computer composition." Reprinted in *Machine Models of Music*, S. M. Schwanauer and D. A. Levitt (eds.). Cambridge, MA: The MIT Press, 1993, 167–186.
- Morales-Manzanares, R., Morales, E. F., Dannenberg, R., and Berger, J. 2001. "SICIB: An interactive music composition system using body movements." *Computer Music Journal* 25(2): 25–36.
- Pachet, F., and Roy, P. 1998. "Formulating constraint satisfaction problems on part-whole relations: The case of automatic harmonization." In *ECAI'98 Workshop on Constraint Techniques for Artistic Applications*. Brighton, UK.
- Papadopoulos, G., and Wiggins, G. 1998. "A genetic algorithm for the generation of jazz melodies." In *Proceedings of the SteP'98 Conference*. Finland.
- Papadopoulos, A., Roy, P., and Pachet, F. 2016. "Assisted lead sheet composition using FlowComposer." *Proceedings of the 22nd International Conference on Principles and Practice of Constraint Programming—CP*. Toulouse, France.
- Partridge, D., and Rowe, J. 1994. *Computers and Creativity*. Bristol: Intellect Books.
- Rader, G. M. 1974. "A method for composing simple traditional music by computer." Reprinted in *Machine Models of Music*, S. M. Schwanauer and D. A. Levitt (eds.). Cambridge, MA: The MIT Press, 1993, 243–260.
- Ritchie, G. D., and Hanna, F. K. 1984. "AM: A case study in AI methodology." *Artificial Intelligence* 23: 249–268.
- Ritchie, G. D. 2009. "Can computers create humour." *AI Magazine* 30(3): 71–81.
- Rothgeb, J. 1969. "Simulating musical skills by digital computer." Reprinted in *Machine Models of Music*, S. M. Schwanauer and D. A. Levitt (eds.). Cambridge, MA: The MIT Press, 1993, 157–164.
- Sabater, J., Arcos, J. L., and López de Mántaras, R. 1998. "Using rules to support case-based reasoning for harmonizing melodies." In *AAAI Spring Symposium on Multimodal Reasoning*. Menlo Park, CA: American Association for Artificial Intelligence, 147–151.
- Schwanauer, S. M. 1993. "A learning machine for tonal composition." In *Machine Models of Music*, S. M. Schwanauer and D. A. Levitt (eds.). Cambridge, MA: The MIT Press, 511–532.
- Schwanauer, S. M., and Levitt, D. A. (eds.). 1993. *Machine Models of Music*. Cambridge, MA: The MIT Press.
- Searle, J. 1980. "Minds, brains and programs." *Behavioral and Brain Sciences* 3(3): 417–457.
- Simon, H. A., Sumner, R. K. 1968. "Patterns in music." Reprinted in *Machine Models of Music*, S. M. Schwanauer and D. A. Levitt (eds.). Cambridge, MA: The MIT Press, 83–110.
- Sims, K. 1994. "Evolving virtual creatures. Computer graphics." In *SIGGRAPH 94 21st International ACM Conference on Computer Graphics and Interactive Techniques*. New York: ACM, 15–22.
- Sims, K. 2016. "Reaction-diffusion media wall." <http://www.karlsims.com/rd-exhibit.html>.



- Suzuki, T., Tokunaga, T., and Tanaka, H. 1999. "A case-based approach to the generation of musical expression." In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*. Burlington, MA: Morgan Kaufmann, 642–648.
- Thom, B. 2001. "BoB: An improvisational music companion." (Unpublished PhD dissertation). School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.
- Tobudic, A., and Widmer, G. 2003. "Playing Mozart phrase by phrase." In *Proceedings of the 5th International Conference on Case-Based Reasoning*, K. D. Ashley and D. G. Bridge (eds.). *Lecture Notes in Artificial Intelligence*, Vol. 3155. Heidelberg: Springer, 552–566.
- Tobudic, A., and Widmer, G. 2004. "Case-based relational learning of expressive phrasing in classical music." In *Proceedings of the 7th European Conference on Case-Based Reasoning*, P. Funk and P. A. Gonzalez Calero (eds.). *Lecture Notes in Artificial Intelligence*, Vol. 3155. Heidelberg: Springer, 419–433.
- Turing, A. M. 1950. "Computing machinery and intelligence." *Mind* LIX(236): 433–460.
- Wessel, D., Wright, M., and Kahn, S. A. 1998. "Preparation for improvised performance in collaboration with a Khyal singer." In *Proceedings of the 1998 International Computer Music Conference*. San Francisco: International Computer Music Association.
- Widmer, G., Flossmann, S., and Grachten, M. 2009. "YQX plays Chopin." *AI Magazine* 30(3): 35–48.
- Woods, W. 1970. "Transition network grammars for natural language analysis." *Communications of the ACM* 13(10): 591–606.
- Yee-King, M., and d'Inverno, M. 2014. "Pedagogical agents for social music learning in crowd-based socio-cognitive systems." In *Proceedings of the First International Workshop on the Multiagent Foundations of Social Computing, AAMAS-2014*. Paris, France.



OPENMIND CHANNEL



READ THE FULL BOOK

The Next Step: exponential life

[+]

RELATED ARTICLES

Frontiers and Knowledge in Music?

A Few Notes

[+]

Future Opera for Robots and People Too

[+]

**The Critical Role of Artists
in Advancing Augmented Reality**

[+]

OTHER BOOKS

