Amy K. Hoover, Paul A. Szerlip, and Kenneth O. Stanley
Department of Electrical Engineering and Computer Science (CS Division)
University of Central Florida
4328 Scorpius Street
Building 116-Room 346 Orlando
Florida 32816-2362, USA
ahoover@eecs.ucf.edu,
paul.szerlip@gmail.com,
kstanley@eecs.ucf.edu

Functional Scaffolding for Composing Additional Musical Voices

Abstract: Many tools for computer-assisted composition contain built-in music-theoretical assumptions that may constrain the output to particular styles. In contrast, this article presents a new musical representation that contains almost no built-in knowledge, but that allows even musically untrained users to generate polyphonic textures that are derived from the user's own initial compositions. This representation, called functional scaffolding for musical composition (FSMC), exploits a simple yet powerful property of multipart compositions: The pattern of notes and rhythms in different instrumental parts of the same song are functionally related. That is, in principle, one part can be expressed as a function of another. Music in FSMC is represented accordingly as a functional relationship between an existing human composition, or scaffold, and a generated set of one or more additional musical voices. A human user without any musical expertise can then explore how the generated voice (or voices) should relate to the scaffold through an interactive evolutionary process akin to animal breeding. By inheriting from the intrinsic style and texture of the piece provided by the user, this approach can generate additional voices for potentially any style of music without the need for extensive musical expertise.

Because musical structure is well-established and understood by music theoreticians, researchers often impart this expertise to the compositional tools that they create. However, such formalized rules or carefully extracted statistical relationships inevitably yield a musical space that constrains results to particular styles and genres, thus limiting the users' ability to explore outside their bounds (Cope 1987; McCormack 1996; Todd and Werner 1999; Conklin 2003; Pachet 2003, Chuan 2009).

In contrast, this article introduces a new representation of musical relationships for computer-assisted composition that includes almost no built-in knowledge of musical structure (apart from the key and the smallest unit of rhythm) yet still helps users compose starting from some preexisting music, called a *scaffold*, that contains one or more simultaneous musical voices. Called *functional scaffolding for musical composition* (FSMC), this approach exploits two rarely explored mathematical properties of music: (1) music can be represented as a function of time (Putnam 1994) and (2) multiple simultaneous voices in a coherent piece are functionally related to each other. Interestingly, these properties alone are sufficient to be harnessed to create additional musical voices. In particular, the existing parts from the scaffold are functionally transformed into one or more additional voices through a neural-network-like representation called a *compositional pattern-producing network* (CPPN; Stanley 2007), which takes the scaffold as input and produces a corresponding generated voice as output. The key insight that makes this approach interesting is that simply creating a functional relationship between one sequence of notes and another, with no other musical principles, is enough to create the effect that one sequence is a plausible simultaneous countermelody, harmony, or accompaniment of the other.

To implement this idea in practice, a program called MaestroGenesis (available online at maestrogenesis.org) was introduced, with the intent of allowing users to explore the space of computer-generated musical voices created by such CPPN-based transformations. It helps users navigate the space of possible transformations (i.e., additional voices) by presenting the user with a set of candidate

Computer Music Journal, 38:4, pp. 80–99, Winter 2014
doi:10.1162/COMJ_a_00269

generated voices and allowing the user to choose the best. A new set of candidate generated voices then inherit some of the appealing traits of those chosen from the previous generation. This process, called *interactive evolutionary computation* (IEC, cf. Dawkins 1986; Sims 1991; Takagi 2001), can be repeated many times to evolve towards a desired "feel." The underlying evolutionary algorithm that enables this process is called *neuroevolution of augmenting topologies* (NEAT), which will be discussed in greater detail subsequently.

Because FSMC emphasizes the importance of functional relationships between parts of a song, the hope is that MaestroGenesis can create additional high-quality, computer-generated voices through functional transformations. This article is the first to present such a comprehensive overview of FSMC and MaestroGenesis by distilling and expanding the results from a series of prior conference papers (Hoover, Szerlip, and Stanley 2011a,b; 2012) and also by presenting an entirely new study exploring experiences of amateur musicians with MaestroGenesis. FSMC is a significant step toward assisted music generation based on the surprisingly simple hypothesis that functional relationships alone are sufficient to generate plausible musical voices. Although the impact of this technology on the musical creativity of amateurs will emerge from its use over time, the identification of such a simple and generic principle can potentially broaden the application of assistive musical technologies in the future.

The next section provides relevant background that is followed by an introduction to the FSMC approach. Subsequent sections present four experiments and their results, whose implications are then discussed.

Background

Much of the expressive potential of computer-generated music derives from the power of the chosen musical representations. This section discusses prior approaches and representations in computer-generated music and describes a precursor to the FSMC method called NEAT Drummer.

Representations in Computer Music and Human–Computer Collaboration

Many musical representations have been proposed before FSMC, although their focus is not necessarily on representing the functional relationship between parts. For example, Holtzman (1981) describes a musical grammar that generates harp solos based on the physical constraints faced by harpists. Similarly, Cope (1987) derives grammars from the linguistic principles of haiku to generate music in a particular style. Although grammars can produce a plausible rhythmic and melodic structure, deciding which aspects of musical structure should be represented by a grammar is often difficult and ad hoc (Kippen and Bel 1992; Marsden 2000).

An alternative to manually constructing grammars is to discover important musical relationships through statistical analyses of musical corpora that then guide decision making (Ponsford, Wiggins, and Mellish 1999; Gillick, Tang, and Keller 2009). Although this approach represents a significant contribution towards understanding how music can be generated by computers, the challenge is to gather sufficient data to generate plausible music without being too restrictive.

There have been many different approaches to incorporating human input into computer-generated music. The grammar-based program Impro-Visor helps users create monophonic jazz solos by automatically composing any number of measures in the style of famous jazz artists (Keller and Morrison 2007; Keller et al. 2014). Styles are represented as grammars that the user can invoke to complete compositions. Impro-Visor is an innovative tool for teaching jazz styles to experienced musicians, but it focuses on emulating prior musicians rather than exploration of a new sound. Another program, MySong, generates chord-based accompaniments for a vocal piece using hidden Markov models (Simon, Morris, and Basu 2008). However, MySong also requires a significant database of specific examples that must be carefully constructed by the programmers. Some programs address the need for data by offloading the responsibility of rule and database construction to the user (Zicarelli 1987; Chuan 2009).

In contrast to works that depend upon specific rules or trained transition tables, the aim in this article is to exploit very general, high-level principles that can be applied across a broad range of compositions and styles while still taking users' input in the tradition of interactive evolution, described next.

Interactive Evolutionary Computation

A popular approach to facilitating creativity in non-experts in a variety of domains is a process similar to animal breeding called interactive evolutionary computation (IEC, see Dawkins 1986; Sims 1991; Takagi 2001). The idea is that humans, rather than hard-coded rules, can rate candidate generated music in place of an explicit fitness function. Interactive evolutionary computation originated in Richard Dawkins' book *The Blind Watchmaker*, in which he described a simple program called Biomorphs, which is meant to illustrate evolutionary principles. The program displays a set of several pictures (also called Biomorphs) on the screen at one time. The user then selects his or her favorite from among those pictures (called the *population*). From that selection, a new generation of *offspring* is spawned that replaces the original population. Because the offspring are generated through slight mutations of the underlying genes of the selected parents, they tend to resemble their parents, while still suggesting novel traits. In this way, over many generations, the user in effect *breeds* new forms (Dawkins 1986).

Music composition is a popular application of IEC. Users specify the candidate compositions they like best, which are then mutated to create new candidates (Nelson 1993; Biles 1994, 2007; Johanson and Poli 1998; Tokui and Iba 2000; Collins 2002; Hoover and Stanley 2009). Most such systems impose explicit musical rules (often grammatical) conceived by the developer to constrain the search spaces of possible musical voices, thereby narrowing the potential for discovery. Thus the unexploited opportunity at the focus of this article is to borrow from the creative seed already present in the user-created scaffold, to enhance the generated output with very few formalized constraints.

NEAT Drummer and CPPNs

FSMC builds upon the authors' previous work on an IEC-based system called NEAT Drummer, which creates percussion patterns for existing compositions (Hoover, Rosario, and Stanley 2008; Hoover and Stanley 2009). The drum generator transforms an input song into a drum pattern that embellishes the pitch and rhythmic patterns of the original song

This transformation occurs through a special type of function representation called a compositional pattern-producing network (CPPN, cf. Stanley 2007), which is also the representation used in FSMC. The CPPN is a network of interconnected nodes similar to a neural network. Unlike traditional neural networks, however, each node in a CPPN can compute a different type of function. In both NEAT Drummer and MaestroGenesis, hidden-node activation functions include Gaussian, sigmoid, linear, sine, and multiplicative functions. These induce different types of symmetries and patterns with particular regularities. For example, in addition to generating drum patterns in NEAT Drummer, they are the representation behind the images evolved interactively by users in the Picbreeder and Endlessforms online services (Clune and Lipson 2011; Secretan et al. 2011), which yielded spatial patterns with regularities and symmetries. CPPNs are, in effect, generic pattern generators capable of producing patterns in space (such as images), just as they can produce patterns in time (such as music).

CPPNs are typically evolved by an algorithm called neuroevolution of augmenting topologies (NEAT, cf. Stanley and Miikkulainen 2002), which produces a new generation of CPPNs from those selected in the current generation. Although the NEAT method was originally developed to solve difficult control and sequential decision tasks, in both NEAT Drummer and MaestroGenesis it is chosen for its ability to evolve minimal CPPN topologies. Both programs begin evolution with simple random CPPNs, each with one hidden node and a variable number of input and output nodes chosen by the user; the weights and activation functions are assigned through uniform random numbers at the beginning. NEAT then incrementally evolves CPPNs by gradually adding nodes and connections through crossover and mutation, which means the patterns they generate can become more complex. Only those structures that are found to be useful through interactive fitness evaluations survive. By starting with simple networks, NEAT searches through a minimal number of weight dimensions to find the appropriate complexity level for the problem.

NEAT Drummer (Hoover and Stanley 2009), the predecessor to FSMC, demonstrated that the generic pattern-generating capability of CPPNs can indeed be applied to musical patterns by generating percussion accompaniment. In NEAT Drummer, users breed percussion accompaniments by selecting those with the most appealing musical qualities. Although NEAT Drummer showed that functional scaffolding (implemented through CPPNs) can produce credible percussion accompaniment, it left open the question of whether such an approach can produce a complete polyphonic texture, containing parts for multiple instruments, from a monophonic or polyphonic piece provided as input. This is the aim of FSMC.

Approach

Extending the idea of NEAT Drummer to pitch as well as rhythm, instrumental parts in FSMC are generated from existing compositions. These compositions form a scaffold from which generated musical voices are built. In contrast to NEAT Drummer, however, these scaffolds include not only timing information but also pitch information, thereby providing the foundation for melodic and harmonic creation.

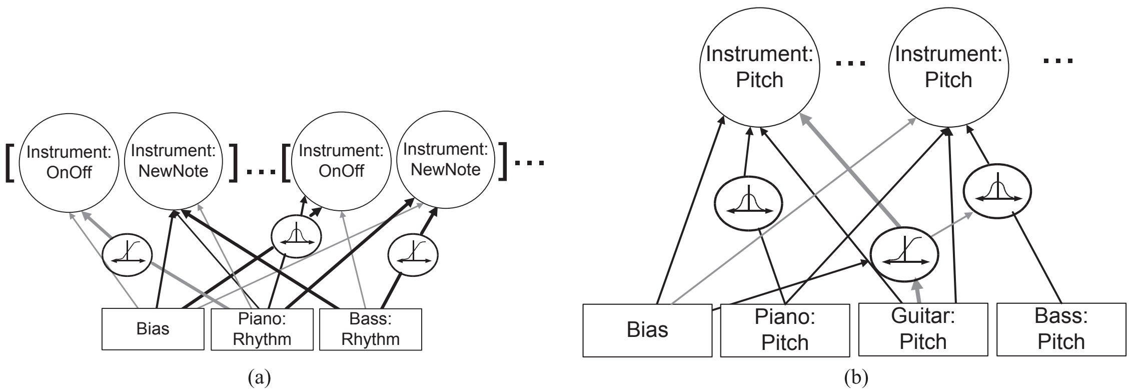
To understand the idea behind FSMC, consider the proposition that if different simultaneous instrumental parts in the same composition were not somehow related to each other, they would probably sound inappropriate together. This relationship can be conceived as a function that describes how one part might be transformed into another. That is, theoretically there exists a function that can transform one sequence of notes and rhythmic information into another. The idea in FSMC is to exploit this fact by literally evolving the function that relates one part to another. That way, instead of searching for a sequence of notes, FSMC can search for a transforming function that bootstraps off the existing parts (i.e., the scaffold) to generate the additional voices. In effect, FSMC is the hidden function that relates different simultaneous parts of a composition to each other.

In particular, this transforming function is encoded in FSMC by CPPNs (Stanley 2007), as detailed herein. It is important to note, however, that any representation of functional relationships could, in principle, serve the role of CPPNs in FSMC. CCPNs are chosen in this implementation for their practical convenience and for their precedent in NEAT Drummer.

Users help to define the search space in FSMC by first selecting the musical starting point, the scaffold. The terms monophonic and polyphonic indicate the number of voices contained in the piece (i.e., a single voice versus multiple voices). Initial scaffolds can be composed in any style and at almost any level of expertise. Advanced users who may only need a single new part for existing compositions can start with a polyphonic composition, whereas single monophonic parts needing multiple layers of additional generated voices can be composed by users within a wide range of musical skill and expertise. The main insight behind the representation in FSMC is that a robust space of generated musical voices can be created with only this initial scaffold. Because of the relationship of different generated voices to the scaffold and, therefore, to each other, the space is easily created and explored.

Recall that a CPPN is in effect a kind of neural network with heterogeneous activation functions at its nodes. The CPPNs depicted in Figure 1 implement the idea of functional scaffolding. Each generated voice is encoded by two CPPNs: one for rhythm and one for pitch. Each CPPN is itself just a formalism for specifying a function that can be artificially evolved. The inputs to the CPPNs are the patterns of notes and durations within the scaffold, and the outputs form the generated voice. In this way, each CPPN is literally a function of the scaffold that thereby transforms the

Figure 1. Compositional pattern-producing networks (CPPNs) compute a function of the input scaffold. The rhythm CPPN (a) and pitch CPPN (b) together form the generated music of FSMC. The inputs to the CPPNs (at bottom) are the scaffold rhythms and pitches for the respective networks, and the outputs (at top) indicate the rhythms and pitches of the generated voices. The internal topologies of these networks, which encode the functions they perform, change through evolution. Although these particular CPPNs depict an evolved arrangement of hidden nodes and activation functions (i.e., Gaussian and sigmoid functions), an unlimited number and arrangement of hidden nodes can occur through evolution. In MaestroGenesis, each hidden node is represented by either a Gaussian, a sigmoid, a linear, a sine, or a multiplicative function. In this example, two generated voices are depicted, but the number of instrument outputs is, in principle, unlimited. The number of input instruments selected by the user can also vary, depending on how many voices from the scaffold the user includes.



scaffold into a functionally related rhythm or pitch pattern.

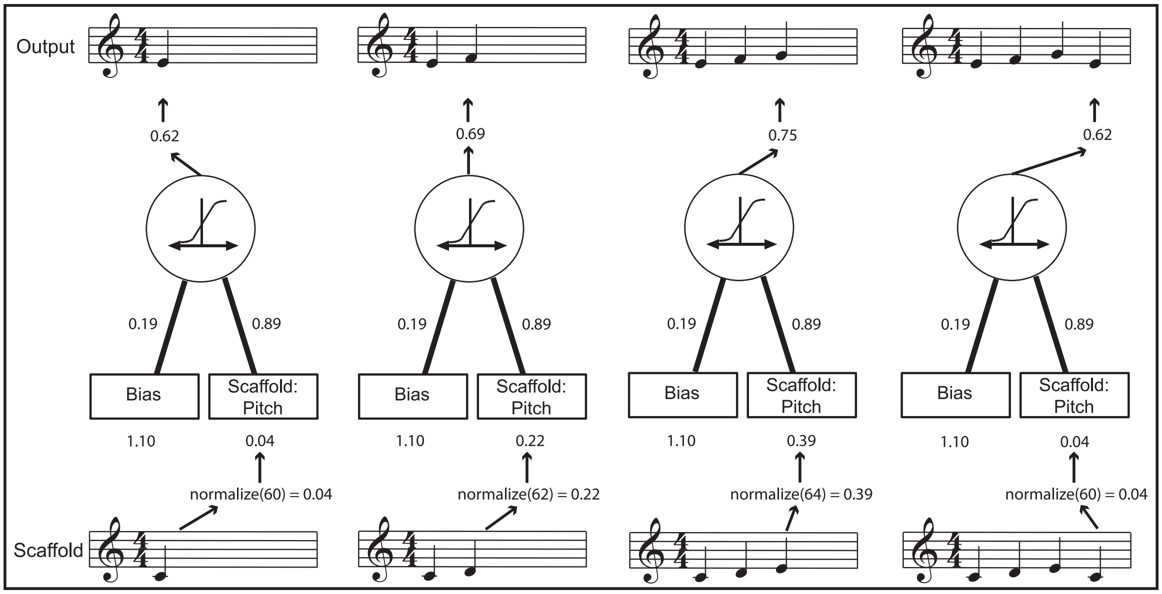
The hidden nodes in the CPPNs depicted in Figure 1 are added by mutations that occasionally occur over the evolutionary process. In effect, they increase the complexity of the transforming function by adding intervening nonlinearities. For example, the Gaussian function introduces symmetry (i.e., such as the same sequence of notes ascending and then descending) and the sigmoid is nonlinear yet asymmetric. By accumulating such transformations within a single CPPN, the relationship between scaffold and generated voice can become more complex.

Each instrumental voice in the output is the result of the two separate functions that independently relate rhythmic and pitch information in the scaffold (i.e., the inputs) to the computer-generated additional voice, as shown in Figure 1. It is important to note that the rhythm and pitch CPPNs are separated intentionally, because combining them into a single CPPN would, in effect, imply that rhythmic positions within a piece are semantically similar to pitches of notes. Such a conflation leads to incoherent patterns, as preliminary experiments with such a setup confirmed. As Figure 1 shows, multiple instruments can be used as input simultaneously, and multiple instruments can be similarly produced as output by the same CPPN. In effect, pitch information from the scaffold is fed into the pitch CPPN at the same time as rhythmic information is fed into the rhythm CPPN. The output of both CPPNs then determines how the generated voice should behave in response. That way, they compute a function of the scaffold, establishing the essential functional relationship.

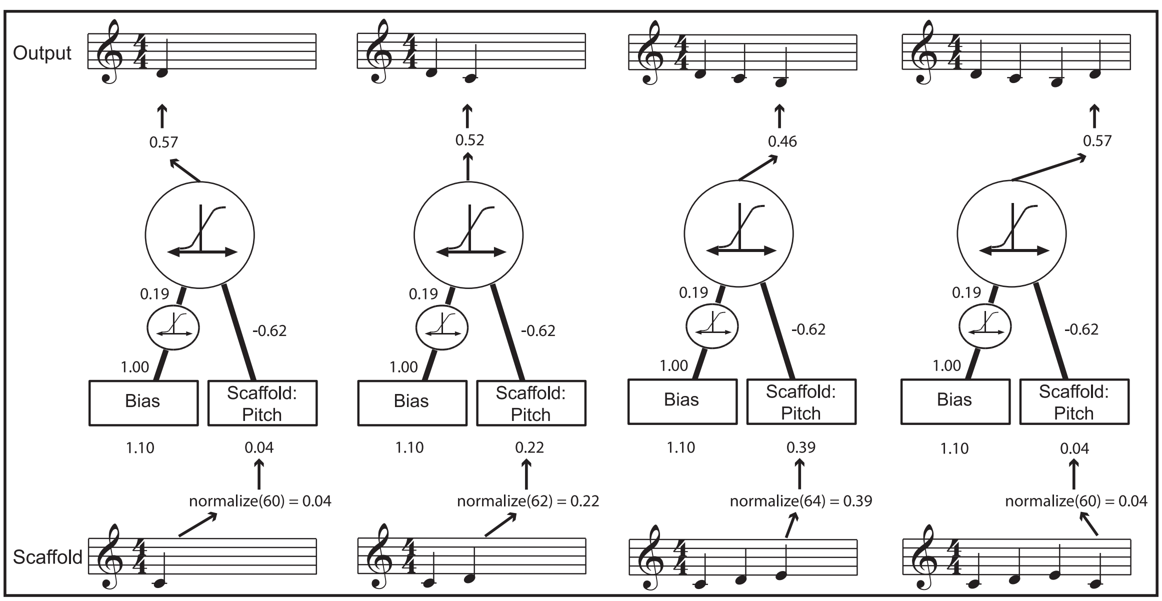
FSMC's musical outputs are divided into a series of discrete time intervals called *ticks*, which are concatenated together to form an entire piece. Each tick typically represents the length of an eighth note, although it could be a shorter or longer unit. Outputs are gathered from both the rhythmic and pitch CPPNs at each tick and are combined to determine the particular note (or rest) at that tick. As shown in Figure 1a, the two outputs of the rhythm network for each generated voice are (1) *OnOff*, which indicates whether a note or rest is played and (in the case of a note) its volume, and (2) *NewNote*, which indicates whether or not to sustain the previous note. The single pitch output for each generated voice in Figure 1b determines the instrument's pitch at the current tick (if a note is played).

To help illustrate intuitively how CPPNs work to encode functional transformations in FSMC, Figure 2 shows how pitch outputs are calculated at each tick and how CPPN mutations can affect the generated output. The sequence in Figure 2 is a

Figure 2. Pitch CPPNs over two generations over time: initial generation pitch CPPN (a) and second generation pitch CPPN, with mutations (b). The pitch CPPNs in the two generations illustrate how scaffolds are transformed to musical outputs. Each of the four identical CPPNs in the first generation and the four identical CPPNs in the second generation represent a calculation made at the four quarter-note-length ticks in this one measure scaffold. The CPPN from the first generation has yet to evolve hidden nodes, whereas the CPPN from the second generation has evolved a hidden node between the bias and output and mutated the existing connection weights.



(a)



(b)

simple example of both how CPPNs calculate their outputs and also how mutations to the CPPN in Figure 2a alter the output it produces for the same scaffold (shown in Figure 2b). This example focuses on the pitch CPPN, but the rhythm CPPN computes its transformations in an analogous manner. Each of the four identical pitch CPPNs in Figure 2a and the four identical CPPNs in Figure 2b represent a calculation made at a particular tick from both a bias (which is just a constant input) and scaffold input. To calculate the output value for the simple CPPN in Figure 2a (which has just one activation function), the bias value of 1.1 and the particular scaffold value (which represents a normalized MIDI pitch) at the given tick are multiplied by their respective connection weights within the CPPN (0.19 and 0.89) and added together to produce a sum called ActivationSum. The value that results from

$$\mathrm{sigmoid}(\mathrm{ActivationSum}) = 1/(1 + e^{-2\times \mathrm{ActivationSum}})$$

is a real number in the range $[0, 1]$ that is then mapped to one of the 15 diatonic notes in the two-octave range in a key that was set by the user. In Figure 2, the first note calculated by the CPPN is E above middle C, because the key is C major and the calculated number 0.62 is a little more than 9/15, or nine diatonic steps above the starting pitch of C below middle C. Whereas the output of MaestroGenesis is constrained to standard diatonic keys, the value of scaffold inputs can be any chromatic note. This example shows that it is in effect the weights of the connections within the CPPN (which act like coefficients) and the particular activation functions within its nodes that determine what the CPPN calculates as output for a particular scaffold input.

Unlike the network in Figure 2a, the CPPN in Figure 2b has evolved the existing connection weights and a new sigmoid hidden function between the bias input and the output. Because there are now two activation functions in the CPPN, two separate activation sums and values must be considered. Starting from the bottom of the CPPN, the activation sum for the hidden node is calculated first and is sent to its sigmoid activation function such that the output (activation level) for the hidden node is

$$\mathrm{sigmoid}(1.1 \times 1.0).$$

For the second activation sum, the previously calculated hidden node value is multiplied by its connection weight of -0.19 and added to the scaffold input value multiplied by its connection weight,

$$\mathrm{normalize}(\mathrm{midi}) \times -0.62.$$

The final output depends on the current tick, but is represented by the function

$$\mathrm{output} = \mathrm{sigmoid}(\mathrm{sigmoid}(1.1 \times 1.0) \times -0.19 + (\mathrm{normalize}(\mathrm{midi}) \times -0.98)).$$

While the generated melody in Figure 2a is transposed a diatonic third from the scaffold pitches, the additional hidden function and corresponding weight mutations in Figure 2b in effect mirror (invert around E) the melody generated in Figure 2a and transpose it down a diatonic second.

In summary, the example in Figure 2 shows how it is possible for a network of weights and activation functions (the CPPN) to compute functional transformations of a sequence of pitches, and how mutations to the CPPN can perturb the nature of such transformations, enabling the discovery of different relationships through an evolutionary process.

Figure 3 illustrates through another example how FSMC interprets the rhythmic (at left) and pitch (at right) information contained in the scaffold. In this example, the scaffold is from the folk song "Scarborough Fair." Each instrument in the scaffold, i.e., oboe, nylon-string guitar, clavinet I, and clavinet II, is an input to both the rhythm and pitch CPPNs, which are evolved (as explained in the next section) to create the computer-generated additional voice for "Scarborough Fair." To produce the outputs, rhythmic and pitch information from the scaffold is sent to the CPPN at each tick. To encode rhythm, when a note strikes or a rest begins, it is represented as a maximum input level that decays linearly over time (i.e., over a number of ticks) until the note ends. The decay does not affect audio or amplitude envelope. Instead, it indicates to the CPPN how many ticks have elapsed since the note was struck. At the same tick, pitch information on the current note is sent into the pitch CPPN as a MIDI pitch value modulus 24. That is, C4 and C5 are differentiated, but C4 is equivalent to C6. The net effect is that the time within each note,

Figure 3. Representing the scaffold. The additional voice for "Scarborough Fair" (top) is generated from the four instruments in the scaffold (bottom). Each of these instruments is used as input to both the rhythmic CPPN (middle left) and the pitch CPPN (middle right). This example is available online at eplex.cs.ucf.edu/fsmc/cmj. (See text for further explanation.)

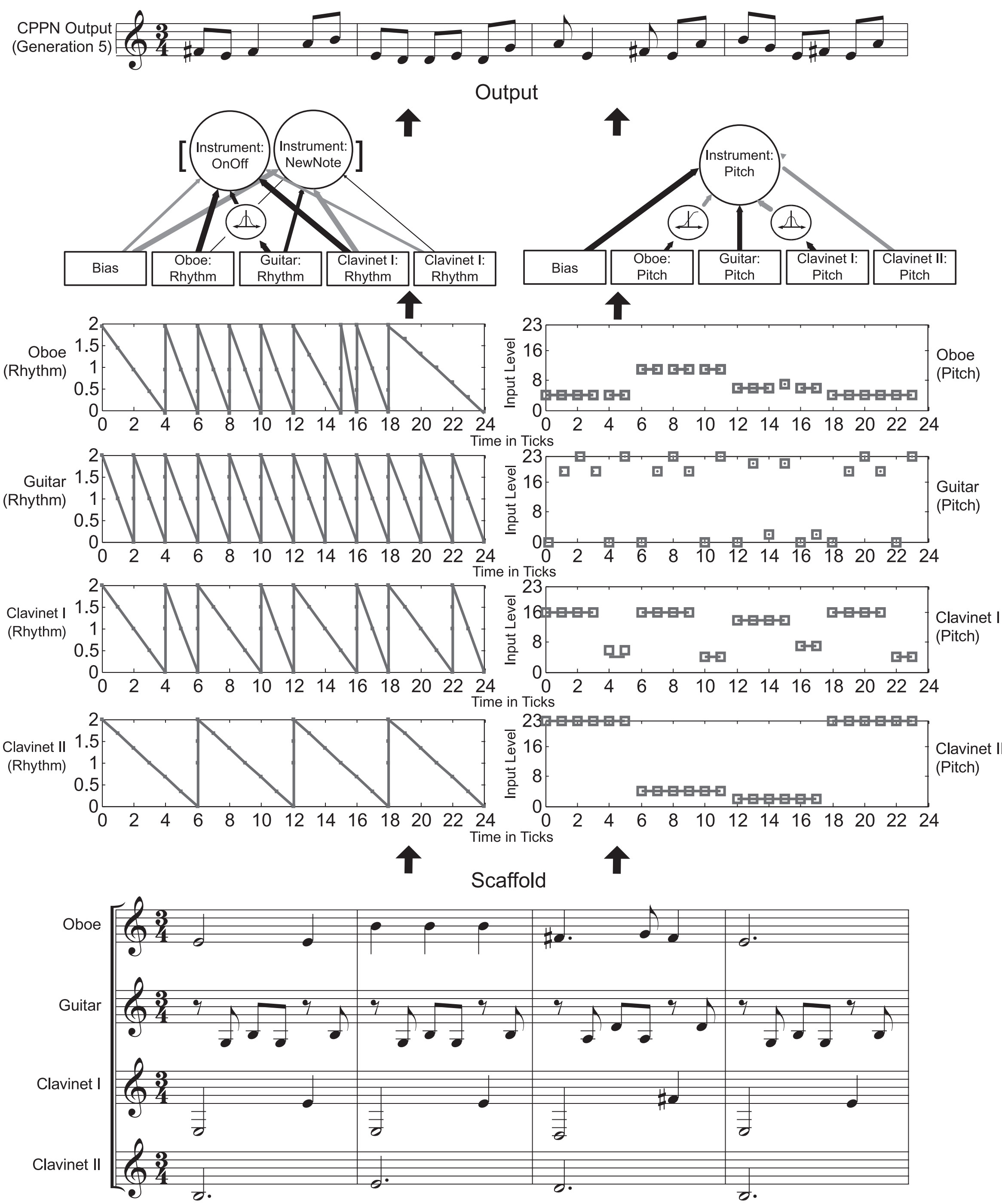
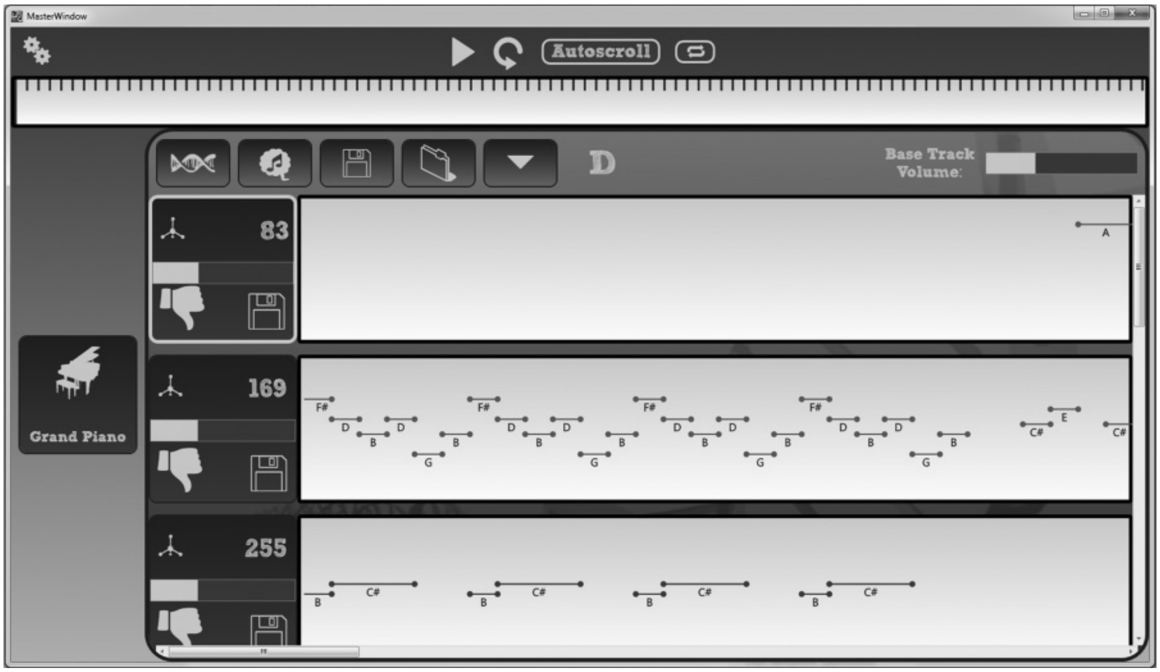


Figure 4. Screenshot of the program MaestroGenesis, which implements FSMC, showing generated voices for a melody entered by the user. The user selects his or her favorites and then requests a new generation of candidates.



or the number of elapsed ticks, and its pitch are known to the rhythm and pitch CPPNs at every tick.

To implement FSMC in practice, the instrument MaestroGenesis program provides the interface to help the user explore possible CPPN-encoded transforms to user-chosen scaffolds (see Figure 4). The sound of instruments in MaestroGenesis can be altered through instrument choice or key. A user can pick any of 128 pitched MIDI instruments and can request any of the major or natural minor keys. Whereas FSMC can potentially generate additional voices in any key, the outputs from MaestroGenesis are restricted to the seven pitch classes of the current key signature. Once the user selects a preexisting piece to provide the scaffold and the output instruments most appropriate for the piece, candidate CPPNs can be generated, thus establishing the musical space of generated voices. The theory behind this approach is that, by exploring the potential relationships between a scaffold and the voices generated from that scaffold (as opposed to exploring direct representations of the voice itself), the user is constrained to a space in which candidate generated voices are almost all likely to be coherent with respect to the scaffold.

Of course, selecting the scaffold itself an important task. It requires selecting an existing composition and then choosing to which instrument tracks the rhythm and pitch networks should "listen." For example, whether or not the rhythm network listens to a fast-changing instrument can affect the perceived complexity of the corresponding generated output. In fact, chosen tracks do not have to be the same for each network (e.g., the rhythm network can have piano and guitar inputs while the pitch network only has a bass guitar input).

Exploration of musical space in FSMC begins with the presentation to the user of the output of ten randomly generated CPPN pairs, each defining the musical relationships between the scaffold and generated output (as shown in Figure 4). These can be played through either MIDI or MP3 formats, the latter resulting from the open-source FluidSynth sound-font simulator, available online at sourceforge.net/apps/trac/fluidsynth. The user-guided process of exploration that combines and mutates these candidates is IEC, explained earlier.

The user explores musical voices in this space by selecting and rating one or more of the computer-generated voices from one generation to parent the individuals of the next. The idea is that the good musical ideas from both the rhythmic and pitch functions are preserved with slight alterations or combined to create a variety of new but related functions, some of which may be more appealing than their parents. The space can also be explored without combination by selecting only a single generated voice. The next generation then contains slight mutations of the original functions.

Although IEC has previously been applied to music generation (Nelson 1993; Biles 1994; Moroni et al. 2000; Bäckman and Dahlstedt 2008), instead of manipulating single notes or features of a composition, FSMC permits the evolution of entire functional relationships, thereby ensuring that the search space only considers generated voices that have some relationship to the scaffold. Because the parts of the scaffold themselves are composed by humans, and therefore sound appealing, generated voices built from any combination of such tracks end up acknowledging and transforming the pitch and rhythmic patterns of the original song.

One application of FSMC is generating single-instrument voices for an existing polyphonic composition. For this purpose, the user selects any number of precomposed tracks and generates a single generated voice for the piece. Because this approach requires an existing composition, it can help composers with writer's block who would only like creative assistance with single voices, or amateurs with little composition experience.

To achieve a polyphonic feel, another application is to evolve multiple generated voices from monophonic melodies, rather than from polyphonic pieces. A natural approach to generating such polyphony is through a layering technique, whereby generated voices from previous generations can be manually entered as input to new CPPNs that then generate more layers of harmony. The result is the ability to spawn an entire multilayered piece from a single monophonic starting melody.

With any of these approaches, or a combination of them, users can further influence the output by holding either the rhythm CPPN or pitch CPPN constant while letting the other CPPN evolve. When two generated voices share the same rhythm network but differ slightly in the pitch network, the two monophonic instruments effectively combine to create the rhythmic and melodic structure of a single polyphonic instrument. Similarly, the pitch networks can be shared while the rhythm networks are evolved separately, creating a different rhythmic and melodic feel. Notice that no musical expertise is needed in any of these scenarios to generate multiple musical voices.

Experiments

Functional scaffolding exploits the insight that music is a function of time and that musical parts are functionally related to one another. The experiments in this section are designed to address the hypothesis that the functional relationship is sufficient to enable users to discover plausible musical voices. Although these experiments focus on the generation of folk music, with slight modification to the MaestroGenesis program, additional voices could be generated in other musical domains, such as experimental styles of contemporary art music.

The first experiment explores the structure of the search space by tracking musical quality over the evolution of a particular generated voice. Independent listeners to the generated works were asked to rate the quality of the pieces at the beginning, middle, and end of evolution.

A separate, but related, issue is the level of quality of generated voices that are completed. For example, is it possible to tell that such pieces are partly composed by a computer? To answer this question, the next experiment tests whether listeners can distinguish between two partially computer-composed pieces and two fully human-composed pieces. It also explores the internal structure of evolved voices.

In the experiments outlined so far, the scaffolds are polyphonic, thereby providing a rich context for generating additional musical voices. The third experiment, therefore, examines whether there is enough information in a single monophonic melody to scaffold an entire multipart piece. If there is, FSMC can potentially enhance the creativity of amateur musicians who may only feel comfortable or capable of composing their own monophonic melodies. The third experiment concludes, accordingly, with a study of user self-assessment to provide a perspective on users' own perceptions of their experience with FSMC.

Audio of the results is available online at eplex.cs.ucf.edu/fsmc/cmj.

System Parameters and Setup

In each experiment, the user chooses the number and type of inputs based on the number of voices contained in the scaffold. The selected voices can be a subset of the scaffold voices or can be the entire piece itself. Each CPPN in the initial population also starts with either zero hidden nodes or one hidden node with equal probability. During reproduction, the probability of crossover is 30 percent. Otherwise, the offspring is created by mutating only a single parent. In that case, each individual connection weight has an 80 percent probability of being mutated by adding uniform random noise in the range $[-2.0, 2.0]$. The activation functions within each node (except output nodes that are required to be sigmoidal) can also mutate with 80 percent probability to sigmoid, Gaussian, linear, sine, or multiplication. It is important to note, however, that the user is free to adjust mutation rates through the MaestroGenesis interface to provide more or less variability, thereby avoiding the potential for too many trivial variations. The NewNote threshold is 0.3, which was found in preliminary tests to ensure a reasonable quality of generated music for many different scaffolds. Furthermore, when the OnOff output in the rhythm network (which also indicates volume) falls below 0.3, no note is played. Population size is ten per generation. The initial random weights in the first generation of CPPNs are chosen from a uniform distribution in the range $[-2.0, 2.0]$. The next generation is created through mutation and recombination solely of the user's choices. In general, all these settings were found effective through preliminary testing, and minor variations of these parameters will likely yield similar results.

Investigating the Evolution of Generated Musical Voices

To study the capabilities of FSMC, it is helpful to first analyze in detail a representative evolutionary progression of generated musical voices. Such an analysis, coupled with a user study of perceived quality over generations, help to illuminate how generated voices are evolved and what interactive evolution contributes to the results.

In this experiment, the focus was on the evolution of the generated musical voice. Therefore the scaffold (i.e., music for which the additional voice will be evolved) was chosen to meet an established level of quality. That way, it was possible to determine whether the generated voice could maintain and complement the original quality in the scaffold. For this purpose, the well-known folk song "Bad Girl's Lament" was chosen, using a MIDI file sequenced by Barry Taylor.

The interactive evolutionary process for the example piece was guided by the authors. They applied no knowledge of music theory (such as avoidance of non-chord tones) beyond simply choosing which candidates sounded best. The process proceeded as follows: A set of ten random CPPNs, corresponding to an initial population of FSMC-generated voices, was first created by MaestroGenesis. Among these, those that sounded best were selected by the user. From the selected candidates a new generation of CPPNs was created that are offspring (i.e., mutations and crossovers) of the original generation. This process of listening to candidates, selecting the best, and creating new generations was repeated until a satisfactory generated voice appeared. Although user input was an important aspect of this process, no session lasted more than twelve generations (i.e., no more than twelve preference decisions were ever made), highlighting the overriding importance of the FSMC relationship to constraining generated musical voices to a reasonable set of candidates. Interestingly, in contrast to data-intensive approaches, the only human

Figure 5. Evolutionary musical sequence for part of "Bad Girl's Lament." Three measures of the evolved steel guitar voice from the first, sixth, and twelfth generations of this excerpt are shown at the top, followed by the pitch and rhythm inputs to the CPPN from the scaffold. In this experiment, the tick length (i.e., the smallest rhythmic unit) is a sixteenth note.

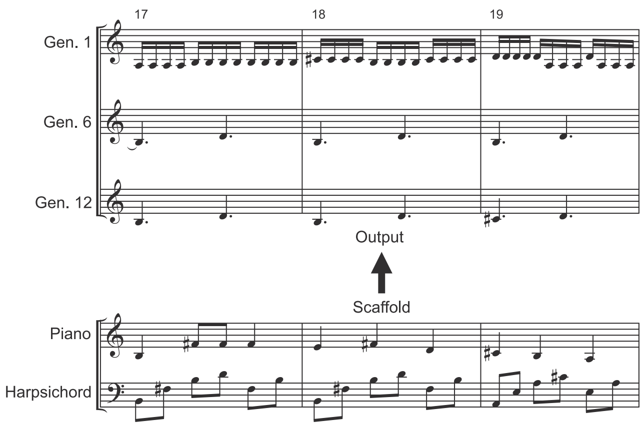


Table 1. Similarity Comparison Results

Voices	*Dissimilarity by Pitch Class*	*Dissimilarity by Contour*
Generation 12 versus Piano	5.8916	3.3074
Generation 12 versus Harpsichord	4.9616	3.9240
Piano versus Harpsichord	5.08401	4.1073

Comparisons illustrate that differences between the human-composed voices are similar to those between the generated voice and human-composed voices. Pitch Class looks at differences between notes between two voices, and Contour looks at differences between note transitions between two voices. For both metrics, the square root of the mean squared error is calculated for pitch values at sixteenth-note intervals. In this piece, the sixteenth note is the smallest rhythmic unit.

knowledge needed to generate musical voices through this approach (aside from the key) was imparted in 10 to 15 clicks of IEC.

To explore the space created by FSMC, the evolutionary progression of an instrumental voice for "Bad Girl's Lament" between the first and twelfth generations was studied by highlighting important milestones at the first, sixth, and twelfth generations. Each sequence represents the one parent chosen out of ten possible candidates. This twelve-generation progression took about 30 minutes, in total, for the user to complete. Most of the time was spent listening to candidate generated voices. The piano and harpsichord channels from the scaffold were inputs to the rhythm CPPN, and only the harpsichord was used as input to the pitch CPPN. For both networks, the smallest rhythmic unit was the sixteenth note.

Figure 5 shows measures 17–19 of the generated voices for "Bad Girl's Lament" in the first, sixth, and twelfth generations. The pitches in measures 17 and 18 of the first generation differ from those created for sixth and twelfth generations. Pitches in the first generation ascend across notes A and B in measure 17, followed by C♯ and B in measure 18. In the sixth and twelfth generations, however, the pattern more closely follows the harpsichord input from the scaffold, with the pitches B and D occurring at the first and fourth eighth note in measures 17 and 18, demonstrating the influence of the functional relationship to the harpsichord on the evolved progressions. In measure 19, the twelfth generation descends to a C♯, thereby echoing the same note in the piano input, even though the CPPN is only aware of pitch changes in the harpsichord. This variation adds a chord tone missing in measure 19 during the first and sixth generations.

Overall, although the three depicted generations in "Bad Girl's Lament" exhibit some similar characteristics, they progressively change over evolutionary time. For example, the sixth and twelfth generations are rhythmically similar, but the first generation uses significantly shorter notes. The pitch evolution progresses similarly to rhythm. Many pitches change from the first to the sixth generation, but the sixth and twelfth generations differ in pitch by only a few choice notes.

Because the different voices within a folk piece are conventionally related, it is reasonable to expect that appealing computer-generated voices in the same context would exhibit a degree of similarity roughly equivalent to the similarity exhibited between the preexisting human-composed voices. To investigate the relationship of the generated voice to the scaffold, Table 1 shows actual similarity comparisons within "Bad Girl's Lament." To obtain these measurements, voices are broken into pitch components (measured in semitone increments) that are calculated at sixteenth-note intervals. In the first column of Table 1, voices are compared by pitch class or note name, and the second column shows differences in melodic contour (i.e., changes from one sixteenth note to the next). For both metrics, the dissimilarity in Table 1 is obtained by first taking the mean squared difference at each sixteenth-note interval. The reported number is then the square root of this number, which thereby reveals on average how many semitones separate the simultaneously sounding pitch classes in two voices, or (in the case of contour) how many semitones of difference there are between the melodic interval in one voice and the simultaneous melodic interval in the other voice.

Given that FSMC generated the pitches for the additional voice from the harpsichord alone, it is not surprising that pitch classes in the generated voice would differ more from the piano than the harpsichord (as shown in the first column of Table 1). With the pitch class metric, however, the difference between the generated voice and the harpsichord and between the generated voice and the piano is similar to the difference between the piano and the harpsichord, which were both part of the original human-composed piece. Thus, the new voice varies in pitch as much relative to the pre-existing voices as these human-composed voices do to each other. Furthermore, the values of these differences, at about 5.3 semitones on average for pitch class (averaged over both scaffold voices), represents a nontrivial gap, thereby suggesting that the evolved transformation is itself nontrivial.

The nature of this relationship, however, cannot be fully elucidated solely from pitch class. For example, the generated voice could potentially be a simple transposition of the human-composed scaffold. Therefore, the dissimilarity in melodic contour between the voices is also calculated (second column in Table 1). In this case, at each sixteenth note, instead of comparing the instantaneous pitch class differences, the difference between the previous and the current sixteenth note's pitch class is calculated. These differences are then compared between two voices. Interestingly, this contour-based metric shows that the generated voice moves more like the piano melody than the arpeggiated chords in the harpsichord from which pitches were generated. However, the slightly higher similarity to the contour of the piano is mainly due to the fact that the piano sustains its notes for longer than the harpsichord, as does the generated voice. More importantly, as with pitch class, the contour difference between both the generated voice and the harpsichord and the generated voice and the piano is similar to the contour difference between the piano and the harpsichord, which again are both part of the original human-composed piece. Thus the new voice varies in melodic contour compared with the pre-existing voices similarly to how the human-composed voices compared to each other. The absolute average dissimilarity of 3.6 (averaged over both scaffold voices) also demonstrates that the generated voice is following a substantially different contour than either of the other scaffold voices, in effect different in its movement by almost four semitones on average for every single sixteenth-note increment in the entire composition. This result, along with the pitch class difference, again reinforces the nontriviality of the evolved transformation.

To understand the effect of evolution on subjective appreciation, a total of 60 listeners, all of whom were students in a diversity of majors at the University of Central Florida, participated in a survey after listening to the evolved variants of "Bad Girl's Lament." In particular, without knowing which was which, they listened to (1) an intentionally poor-quality control with an inappropriate additional generated voice (which helped to establish that participants indeed generally agree on something subjective), (2) the original "Bad Girl's Lament" without additional voicing, (3) the song with an FSMC-generated additional voice selected from the first generation of IEC, (4) the song with the generated voice selected from the sixth generation of IEC, and (5) the final selected song with the additional voice from the twelfth generation. It is important to note that the control, which was from a randomly generated CPPN, benefits from the same key and rhythmic constraints as the other results, ruling out that these alone account for the music's plausibility. For each of the five variants, the listener was asked to rate its quality scale of one to ten, with one as "the worst" and ten as "the best."

By determining the perceived quality of an established composition, it becomes possible to estimate how well evolution can maintain a similar

Table 2. Perceived Quality by Survey Participants

MIDI Name	*Mean*	*Std. Dev.*
Poor Control	4.35	1.93
BGL without Additional Voice	7.30	1.85
BGL, Generation 1	5.15	2.20
BGL, Generation 6	6.07	1.96
BGL, Generation 12	6.83	1.98

This table shows the average ratings and the mean and standard deviation for the control and four "Bad Girl's Lament" (BGL) MIDI sequences.

standard. Table 2 summarizes the results from the listener study, which focused on the same IEC-evolved voices for "Bad Girl's Lament" discussed in the previous section. As expected, the control is rated significantly worse than every other example in the survey (at least $p < 0.05$ for all pair-wise comparisons, using Student's t test). This result established that listeners likely understood the questions in the survey.

Importantly, the sixth generation was judged to be of significantly higher quality than the first generation ($p < 0.05$), and the twelfth generation was judged to be significantly better than the sixth ($p < 0.05$). Furthermore, although the original MIDI sequence, without any additional generated voices, was judged to be significantly better than the sixth generation ($p < 0.001$), it was not judged significantly better than the twelfth generation. Thus evolution, guided by the user, eventually achieved a level of quality, in a short number of generations, that the participants could not distinguish from that of the original, suggesting that FSMC-generated parts can meet an acceptable level of quality.

Comparing FSMC with Fully Human Compositions

The aim of this experiment was to explore whether additional voices generated by FSMC can sound human. To explore this question, an additional voice was generated for the folk song "Nancy Whiskey." Then, the generated voicing for "Nancy Whiskey" and the final generation of "Bad Girl's Lament" from the previous section were included in a "musical Turing test" to determine whether they are distinguishable from completely human-composed pieces.

It is important to note that these pieces were chosen for this experiment because they exemplify entirely human compositions that meet a minimum standard of recognizable quality. That way, it is possible to discern whether the generated additional voices reduce the human plausibility of the work, or whether they complement it successfully.

Evolved Voice for "Nancy Whiskey"

Like the previous experiment with "Bad Girl's Lament," the interactive evolutionary process for "Nancy Whiskey" was guided by the authors using the same experimental settings as previously described. For this song, the main result required only two generations to evolve. The low number of generations necessary to obtain this result is a result of the strong bias provided by FSMC towards generating additional voices related to the scaffold. In sum, the generated voice incorporates pitch and rhythmic elements from all three scaffold instruments, while also varying and combining them in new ways, yielding an original pattern that complements the whole.

The internal structure of the CPPNs that generate the additional voices from "Bad Girl's Lament," in the previous section, and "Nancy Whiskey" are surprisingly simple, each with no more than one hidden node. It is important to understand that the simplicity of these relationships resulted from a process of human selection through IEC that ended when the user was satisfied, which means it reflects his or her implicit preferences.

These results show that simple relationships in the CPPN can yield appealing and convincing musical relationships.

Musical Turing Test

In this second listener study, participants were asked to anonymously rate examples with and without FSMC-generated voices. The key focus in the study is on whether the fact that a computer is involved in generating some of the examples can be discerned by the listeners. Thus the survey is a kind of musical Turing test. This perspective is interesting, because FSMC is based on no musical principle or theory other than establishing a functional relationship; if such a minimalist approach (guided by users' preferences) can generate plausible musical voices it suggests that the theory behind it is at least promising.

For this study, a total of 66 listeners, again all students from a diversity of majors at the University of Central Florida, participated in the study. The full survey, including the human compositions, is also available online at eplex.cs.ucf.edu/fsmc/cmj. Participants were asked to rate five different MIDI sequences by answering the following question:

> Based on your impression, how likely is it that any of the instrumental parts in the musical piece found at the following link . . . were composed by a computer? "Composed" means that the computer actually came up with the notes, i.e., both their pitch and duration, on its own (1 means very unlikely and 10 means very likely).

The participants rated a total of five MIDI sequences: (1) an obviously computer-generated control that was restricted to notes generated in the same key as the scaffold (which helped to establish that participants understood the question), (2) the version of "Nancy Whiskey" with a computer-generated additional voice, (3) a fully human-composed arrangement of "Chief Douglas' Daughter," (4) a fully human-composed version of "Kilgary Mountain," and (5) the version of "Bad Girl's Lament" with the computer-generated voice from the twelfth generation. Thus the main issue is whether participants judge the second and fifth pieces, which had additional voices evolved with FSMC, as distinguishable from the third and fourth pieces, which were entirely composed by humans.

The results of this study are shown in Table 3. On average, the participants judged the intentionally poor example as significantly more likely to be computer-generated than any other song in the survey ($p < 0.001$ according to Student's t test). This difference indicates that participants understood the survey.

Although the accompanied "Nancy Whiskey" was judged significantly more likely ($p < 0.05$) to be computer generated than the human-composed song "Chief Douglas' Daughter," it was not judged significantly more likely than "Kilgary Mountain" to be computer generated. This result indicates that the accompanied "Nancy Whiskey" can pass our musical Turing test, i.e., the participants could not distinguish it from a song that was entirely human-generated. The generated voice for "Bad Girl's Lament" was even more difficult for participants to distinguish from the completely human-composed pieces. It was not judged significantly more likely to be computer assisted than either of the pieces written by a human, i.e., "Chief Douglas' Daughter" or "Kilgary Mountain." In fact, on average, FSMC-accompanied "Bad Girl's Lament" scored as slightly less likely to be computer-generated than the entirely human-composed song "Kilgary Mountain."

Table 3. Survey Results

Name of MIDI Sequence	*Mean*	*Standard Deviation*
Control	7.82	2.15
"Nancy Whiskey" with Additional Voice.	5.45	2.65
"Chief Douglas' Daughter"	4.32	2.61
"Kilgary Mountain"	4.86	2.39
"Bad Girl's Lament" with Additional Voice.	4.82	2.44

Lower mean values indicate a more "human-like" result.

These results validate the premise that additional evolved voices are at least plausible enough to fool human listeners into confusing partly computer-generated compositions with fully human-composed ones.

Generating Polyphonic Voices

The experiments in this section are designed to show how users can generate multipart pieces from just a single monophonic melody with FSMC. A creative self-assessment from users of the program studies their experience of the process. The ability to generate convincing polyphonic pieces from just a simple monophonic initial concept would open up musical creativity to anyone who can compose a simple monophonic melody. Thus this experiment explores an important issue in establishing the breadth of potential applications of FSMC.

For this experiment, each of three undergraduate independent study students, Marie E. Norton, Trevor A. Brindle, and Zachary Merritt, composed a monophonic melody. To each of these user-composed melodies, each student added multiple generated voices using FSMC to create a polyphonic texture. Two other sets of multipart additional voices were generated by one of the students for the folk song "Early One Morning," illustrating that results even with the same scaffold are not deterministic. The most important point is that composing an entire polyphonic piece required nothing more than the skill to compose a single monophonic voice. Although results may sound consciously arranged, it is important to bear in mind that all the polyphony is entirely the output of FSMC.

Functional scaffolding provides significant freedom to the user in how to accumulate the layers of a multipart piece. In general, the user has the ability to decide from which parts to generate other parts. For example, from the original melody, five additional parts could be created at once by generating the output for all of them from both a single pitch and single rhythm CPPN. Or instead, the user might accumulate layers incrementally, feeding each new part into a new CPPN pair to evolve yet another layer. Some layers might depend on one previous layer, whereas others might depend on multiple previous layers. In effect, such decisions shape subtle structural relationships and, hence, the aesthetic of the final composition. For example, evolving all of the new parts from just the melody gives the melody a commanding influence over all of the generated voices, while incrementally training each layer from the last induces a more delicate and complex set of complementary partnerships. Overall, the student composers took advantage of this flexibility in a variety of ways.

Interestingly, the two versions of "Early One Morning" (Song 1) illustrate how a single user can generate different voices from the same initial monophonic melody, and how the initial melody exerts its influence, both rhythmically and harmonically, but in different ways. Songs 2, 3, and 4 exhibit a similar effect: rhythmic and harmonic influence from the original melody, yet distinctive and original generated voices. The result is that the overall arrangements sound composed, even though they are evolved through a breeding process.

A key motivation for these polyphonic experiments is that they reflect a likely common mode of usage for FSMC, in which users, who can only create a monophonic melody on their own, expand the initial melody into a full multipart piece with FSMC. To evaluate the effect of the program on their own creative self-expression, the three undergraduates who composed the polyphonic pieces in this section were asked several questions designed to investigate how FSMC affects the composition process of its users. Each of the three students also had experience composing without FSMC, providing a unique opportunity to learn their perspective on its contribution. The aim of this study is to provide a qualitative perspective on the experience of composing with FSMC.

Results indicate that the users were satisfied with ideas suggested by MaestroGenesis. For instance, when asked if "FSMC helped me explore a broader range of creative possibilities than I could before," each respondent indicated that MaestroGenesis helped them explore new areas of their creative search space. The students responded, "FSMC freed me from my normal stylistic tendencies," "I typically follow a sort of pattern when I compose, but FSMC expanded my thinking," and "specific parts of the output harmonies were very good, and I could see myself applying them in many places throughout the song."

When asked to describe the advantage of integrating FSMC into the subjects' own musical creativity process, responses ran from "It would provide as a great source of ideas and inspiration for any work. I could very easily input my composition, evolve it, and develop FSMC outputs to cater to my piece," through "a few of my stylistic elements will come through, other elements will surface," to "great for writer's block." Thus the innovations pushed users outside their normal musical boundaries, but tended to respect the musical direction that was intended.

There were several instances where users found FSMC more limiting than hoped. All three participants indicated that, although they liked the holistic motifs presented by FSMC, they would like more control over the form of the pieces. One respondent said, "I could not shape the harmony produced to suit my melody's form . . . I would need to input the harmony produced into Sibelius to make final corrections and changes." Although the functional representation ensures that the generated voice is based on the pitch and rhythmic patterns of the original piece in its entirety, sometimes different evolved functional relationships might be appropriate for different sections. That is, one function can be more appropriate for an introduction, another for the next section, and so on. This idea will be addressed in future work.

While the users wanted more features, they all indicated that they would generate ideas with FSMC in the future. One student summarized, "I often get writer's block, where nothing sounds how I want. By plugging my unfinished composition into FSMC, I would be able to find inspiration for new techniques, rhythms, or styles."

Discussion and Future Work

Whereas many approaches to automated composition focus on generating music through formalized musical theory (Temperley 2004; Keller and Morrison 2007; Keller et al. 2014) or statistical analysis of large corpora (Ponsford, Wiggins, and Mellish 1999; Rhodes, Lewis, and Müllensiefen 2007; Simon, Morris, and Basu 2008; Gillick, Tang, and Keller 2010; Kitani and Koike 2010), FSMC takes a different tack by starting with almost no rules or assumptions. By starting with so few assumptions, functional scaffolding facilitates exploration of both monophonically and polyphonically generated voices, while still maintaining musical plausibility. Most importantly, experimental results support the hypothesis that functional relationships alone are sufficient (in conjunction with human selection of candidates) to generate plausible musical voices, thereby suggesting a novel perspective on the nature of musical appreciation.

Implications for Musical Appreciation

Although experienced human composers draw on knowledge of musical rules and techniques, FSMC composition occurs only through functional transformations of a given scaffold (guided by the choices of a human, who need not have musical training). Such transformations are powerful, however, because they can generate a wide spectrum of meaningful relationships ranging from simple uniform transposition (e.g., from the key of C to that of D) to more complicated and subtle juxtapositions that elude traditional formalization.

An interesting aspect of FSMC is that the formal concepts corresponding to discovered transformations are never explicitly encoded in the representation. For example, a change in CPPN connection weights can mutate a perfect authentic cadence into a half or even plagal cadence, yet neither MaestroGenesis users nor the program's designers need to recognize cadence types, specify where they should occur, or even know what a cadence is.

In fact, because the emphasis is on generating plausible voices rather than conforming to musical rules, the search process has the potential to yield satisfying generated voices that, nevertheless, do not follow the rules. Interestingly, as illustrated by the studies in this article, the average listener can enjoy the generated musical voices even if they do not completely adhere to compositional tradition. This observation suggests FSMC may be exposing an important factor in musical appreciation that is typically not considered: that an implicit recognition of the functional relationships in music may be important for its appreciation. As Nicholas Cook wrote in *Music, Imagination, and Culture*,

> So it is not the enjoyment of the musical connoisseur who knows something about classical harmony and form that is perplexing: it is the degree of involvement that people who know nothing of these things feel in music, and their ability to respond to music in an appropriate and meaningful manner (Cook 1992, p. 2).

Because an essential aspect of appreciation may be functional relationships, listeners can potentially gain an appreciation for different genres and musical styles by studying the relationships that typify them. For instance, many musicians develop an appreciation for "art music" through their formal musical education. They study atonal works, analyze their compositional structure, and compose in such a style while working toward understanding and appreciating these types of pieces. Perhaps at an abstract level they are learning the functional relationships that relate parts of such music to each other. These functional relationships may, however, also partly explain how even the most educated musicians can appreciate a good riff from a popular song. We are inundated in our own culture with such simple, tonal relationships, from advertisement jingles to nursery rhymes and Christmas carols. FSMC thus hints at the possibility of a simple new approach to understanding the elusive nature of music appreciation.

Practical Applications

The experiments in this article suggest the possibility for humans to collaborate with FSMC to discover novel musical inspiration. Many approaches in this area are restricted by the representation of musical knowledge in the system. A successful composition in these approaches depends, in part, on the designer's ability to identify and reasonably apply key compositional rules (Marsden 2000). Although built-in rules may result in appealing musical pieces, they also constrain a full exploration of musical possibilities. In contrast, because FSMC requires almost no explicit encoding of musical knowledge, the space of generated voices can be theoretically expanded over evolution through the increasing complexity of CPPNs to represent almost any musical relationship.

The results only show a sampling of the possibilities in the folk-song genre, yet FSMC has the potential to help users compose additional musical voices for almost any style of music. Instead of first having the user specify a predefined style and then generating additional voices, FSMC-generated voices inherit style through the user-chosen scaffold. Future versions will also allow multiple simultaneous voices to be generated all in one step. Another idea is to allow FSMC not only to evolve the transforming function, but also to decide which inputs from the scaffold to include. Interesting future work also lies not only in developing filters to present the user with melodies that fit the constraints of a particular musical style, but in exploring the nature of the particular CPPN itself, by applying the same CPPNs to different scaffolds even as the search itself iterates. With all of these potential future extensions, an interesting experiment would be to explore how the FSMC-generated outputs and the user experience with MaestroGenesis compare to other established generative systems in the field.

In fact, the idea of functional scaffolding extends, in principle, beyond music. Inspired by FSMC, Clune, Chen, and Lipson (2013) recently showed that a three-dimensional model can act as a scaffold for related three-dimensional objects. In general, the complexity inherent in any preexisting artifact is a potential scaffold for a search that inherits such complexity from the start.

Daniel Levitin (2006, p. 194) points out that "the chasm between musical experts and everyday musicians has grown so wide in our culture" that people are easily discouraged from experiencing the satisfaction of creating their own performances or compositions. In this context, research efforts like FSMC and MaestroGenesis open the possibility of bringing the joy of making music back to people whose lack of expertise heretofore has forced them only to consume.

Conclusion

This article presented functional scaffolding for musical composition (FSMC), a method based on a simple mathematical principle and on little musical theory that can, nevertheless, generate plausible monophonic or polyphonic accompaniment from as little as a single, human-composed monophonic starting track or scaffold. The approach facilitates exploration by helping the user search candidate generated voices through interactive evolutionary computation. FSMC results in musical compositions that can sometimes be confused with works composed entirely by humans. FSMC is the first approach to explore the simple hypothesis that functional relationships may play a significant role in music appreciation.

Acknowledgments

This work was supported, in part, by the National Science Foundation (NSF) under grant IIS-1002507 and also by a NSF Graduate Research Fellowship. Special thanks to undergraduate students Marie E. Norton, Trevor A. Brindle, and Zachary Merritt, whose compositions were used in the study presented in this article. Special thanks also to Barry Taylor for granting permission to utilize his MIDI productions of folk music in this work. Barry Taylor originally sequenced "Scarborough Fair," "Nancy Whiskey," and "Bad Girl's Lament" (without additional generated voices), as well as "Kilgary Mountain" and "Chief Douglas' Daughter" (which were used as controls in Table 3). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Bäckman, K., and P. Dahlstedt. 2008. "A Generative Representation for the Evolution of Jazz Solos." In M. Giacobini, et al., eds. *Applications of Evolutionary Computing*. Berlin: Springer, pp. 371–380.

Biles, J. A. 1994. "GenJam: A Genetic Algorithm for Generating Jazz Solos." In *Proceedings of the International Computer Music Conference*, pp. 131–137.

Biles, J. A. 2007. "Improvising with Genetic Algorithms: GenJam." In E. R. Miranda and J. A. Biles, eds. *Evolutionary Computer Music*. Berlin: Springer, pp. 137–169.

Chuan, C.-H. 2009. "Supporting Compositional Creativity Using Automatic Style-specific Accompaniment." Paper presented at the Computational Creativity Support Workshop of the ACM Conference on Human Factors in Computing Systems (CHI), 4–9 April, Boston. Available online at computationalcreativity.org/cameraready/chuan.pdf. Accessed July 2014.

Clune, J., A. Chen, and H. Lipson. 2013. "Upload Any Object and Evolve It: Injecting Complex Geometric Patterns into CPPNs for Further Evolution." In *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 3395–3402.

Clune, J., and H. Lipson. 2011. "Evolving Three-Dimensional Objects with a Generative Encoding Inspired by Developmental Biology." In *Proceedings of the European Conference on Artificial Life*, pp. 144–148.

Collins, N. 2002. "Interactive Evolution of Breakbeat Cut Sequences." Paper presented at the Cybersonica Conference, 5–7 June, Institute of Contemporary Arts, London. Available online at www.sussex.ac.uk/Users/nc81/research/ieforbbcut.pdf. Accessed July 2014.

Conklin, D. 2003. "Music Generation from Statistical Models." In *Proceedings of the Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pp. 30–35.

Cook, N. 1992. *Music, Imagination, and Culture*. Oxford: Oxford University Press.

Cope, D. 1987. "An Expert System for Computer-Assisted Composition." *Computer Music Journal* 11(4):30–46.

Dawkins, R. 1986. *The Blind Watchmaker*. Essex, U.K.: Longman.

Gillick, J., K. Tang, and R. Keller. 2009. "Learning Jazz Grammars." In *Proceedings of the Sound and Music Computing Conference*, pp. 125–130.

Gillick, J., K. Tang, and R. Keller. 2010. "Machine Learning of Jazz Grammars." *Computer Music Journal* 34(3):56–66.

Holtzman, S. R. 1981. "Using Generative Grammars for Music Composition." *Computer Music Journal* 5(1):51–64.

Hoover, A. K., M. P. Rosario, and K. O. Stanley. 2008. "Scaffolding for Interactively Evolving Novel Drum Tracks for Existing Songs." In M. Giacobini, et al., eds. *Applications of Evolutionary Computing*. Berlin: Springer, pp. 412–422.

Hoover, A. K., and K. O. Stanley. 2009. "Exploiting Functional Relationships in Musical Composition." *Connection Science* 21(2):227–251.

Hoover, A. K., P. A. Szerlip, and K. O. Stanley. 2011a. "Generating Musical Accompaniment through Functional Scaffolding." In *Proceedings of the Sound and Music Computing Conference*, pp. 161–168.

Hoover, A. K., P. A. Szerlip, and K. O. Stanley. 2011b. "Interactively Evolving Harmonies through Functional Scaffolding." In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 387–394.

Hoover, A. K., P. A. Szerlip, and K. O. Stanley. 2012. "Generating a Complete Multipart Musical Composition from a Single Monophonic Melody with Functional Scaffolding." In *Proceedings of the Conference on Computational Creativity*, pp. 111–118.

Johanson, B., and R. Poli. 1998. "GP-Music: An Interactive Genetic Programming System for Music Generation with Automated Fitness Raters." *Proceedings of the Annual Conference on Genetic Programming*, pp. 181–186.

Keller, R., and D. R. Morrison. 2007. "A Grammatical Approach to Automatic Improvisation." In *Proceedings of the Sound and Music Computing Conference*, pp. 330–337. Available online at smc07.uoa.gr/SMC07%20Proceedings/SMC07%20Paper%2055.pdf. Accessed July 2014.

Keller, R., et al. 2014. "Automating the Explanation of Jazz Chord Progressions Using Idiomatic Analysis." *Computer Music Journal* 37(4):54–69.

Kippen, J., and B. Bel. 1992. "Modeling Music with Grammars: Formal Language Representation in the Bol Processor." In A. Marsden and A. Pople, eds. *Computer Representations and Models in Music*. London: Academic Press, pp. 207–238.

Kitani, K. M., and H. Koike. 2010. "ImprovGenerator: Online Grammatical Induction for On-the-Fly Improvisation Accompaniment." In *Proceedings of the Conference on New Interfaces for Musical Expression*, pp. 469–472.

Levitin, D. J. 2006. *This Is Your Brain on Music*. London: Penguin.

Marsden, A. 2000. "Music, Intelligence, and Artificiality." In E. R. Miranda, ed. *Readings in Music and Artificial Intelligence*. London: Routledge, pp. 15–28.

McCormack, J. 1996. "Grammar Based Music Composition." In R. Stocker, et al., eds. *Complex Systems 96: From Local Interactions to Global Phenomena*. Amsterdam: ISO Press, pp. 321–336.

Moroni, A., et al. 2000. "Vox Populi: An Interactive Evolutionary System for Algorithmic Music Composition." *Leonardo Music Journal* 10:49–54.

Nelson, G. L. 1993. "Sonomorphs: An Application of Genetic Algorithms to Growth and Development of Musical Organisms." In *Proceedings of the Fourth Biennial Art and Technology Symposium*, pp. 155–169. Available online at timara.con.oberlin.edu/gnelson/PapersPDF/morph93.pdf. Accessed July 2013.

Pachet, F. 2003. "The Continuator: Musical Interaction with Style." *Journal of New Music Research* 32(3):333–341.

Ponsford, D., G. Wiggins, and C. Mellish. 1999. "Statistical Learning of Harmonic Movement." *Journal of New Music Research* 28(2):150–177.

Putnam, J. B. 1994. "Genetic Programming of Music." Technical report. Socorro, New Mexico: New Mexico Institute of Mining and Technology.

Rhodes, C., D. Lewis, and D. Müllensiefen. 2007. "Bayesian Model Selection for Harmonic Labelling." In T. Klouche and T. Noll, eds. *Mathematics and Computation in Music*. Berlin: Springer, pp. 107–116.

Secretan, J., et al. 2011. "Picbreeder: A Case Study in Collaborative Evolutionary Exploration of Design Space." *Evolutionary Computation* 19(3):373–403.

Simon, I., D. Morris, and S. Basu. 2008. "MySong: Automatic Accompaniment Generation for Vocal Melodies." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 725–734.

Sims, K. 1991. "Artificial Evolution for Computer Graphics." *ACM SIGGRAPH Computer Graphics* 25(4):319–328.

Stanley, K. O. 2007. "Compositional Pattern Producing Networks: A Novel Abstraction of Development." *Genetic Programming and Evolvable Machines* 8(2):131–162.

Stanley, K. O., and R. Miikkulainen. 2002. "Evolving Neural Networks Through Augmenting Topologies." *Evolutionary Computation* 10(2):99–127.

Takagi, H. 2001. "Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation." *Proceedings of the IEEE* 89(9):1275–1296.

Temperley, D. 2004. *The Cognition of Basic Musical Structures*. Cambridge, Massachusetts: MIT Press.

Todd, P. M., and G. M. Werner. 1999. "Frankensteinian Methods for Evolutionary Music." In N. Griffith and P. M. Todd, eds. *Musical Networks: Parallel Distributed Perception and Performace*. Cambridge, Massachusetts: MIT Press, pp. 313–340.

Tokui, N., and H. Iba. 2000. "Music Composition with Interactive Evolutionary Computation." In *Proceedings of the International Conference on Generative Art*, pp. 215–226.

Zicarelli, D. 1987. "M and Jam Factory." *Computer Music Journal* 11(4):13–29.