Functional Scaffolding for Musical Composition: A New Approach in Computer-Assisted Music Composition

2014

Amy K. Hoover
University of Central Florida

Find similar works at: https://stars.library.ucf.edu/etd

University of Central Florida Libraries http://library.ucf.edu

Part of the Computer Sciences Commons, and the Engineering Commons

STARS Citation

https://stars.library.ucf.edu/etd/4559

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.
FUNCTIONAL SCAFFOLDING FOR MUSICAL COMPOSITION: A NEW APPROACH IN COMPUTER-ASSISTED MUSIC COMPOSITION

by

AMY K. HOOVER
B.S. University of Central Florida, 2009
M.S. University of Central Florida, 2014

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical Engineering and Computer Science in the College of Engineering and Computer Science at the University of Central Florida Orlando, Florida

Summer Term
2014

Major Professor: Kenneth O. Stanley
ABSTRACT

While it is important for systems intended to enhance musical creativity to define and explore musical ideas conceived by individual users, many limit musical freedom by focusing on maintaining musical structure, thereby impeding the user’s freedom to explore his or her individual style. This dissertation presents a comprehensive body of work that introduces a new musical representation that allows users to explore a space of musical rules that are created from their own melodies. 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 an additional generated voice. This relationship is encoded by a type of artificial neural network called a compositional pattern producing network (CPPN). A human user without any musical expertise can then explore how these additional generated voices should relate to the scaffold through an interactive evolutionary process akin to animal breeding. The utility of this insight is validated by two implementations of FSMC called NEAT Drummer and MaestroGenesis, that respectively help users tailor drum patterns and complete multipart arrangements from as little as a single original monophonic track.

The five major contributions of this work address the overarching hypothesis in this dissertation that functional relationships alone, rather than specialized music theory, are sufficient for generating plausible additional voices. First, to validate FSMC and determine whether plausible generated voices result from the human-composed scaffold or intrinsic properties of the CPPN, drum patterns are created with NEAT Drummer to accompany several different polyphonic pieces. Extending the FSMC approach to generate pitched voices, the second contribution reinforces the importance of functional transformations through quality assessments that indicate that some partially FSMC-
generated pieces are indistinguishable from those that are fully human. While the third contribution focuses on constructing and exploring a space of plausible voices with MaestroGenesis, the fourth presents results from a two-year study where students discuss their creative experience with the program. Finally, the fifth contribution is a plugin for MaestroGenesis called MaestroGenesis Voice (MG-V) that provides users a more natural way to incorporate MaestroGenesis in their creative endeavors by allowing scaffold creation through the human voice. Together, the chapters in this dissertation constitute a comprehensive approach to assisted music generation, enabling creativity without the need for musical expertise.
This dissertation is dedicated to “The Hoovers:”

Greg, Mary, Amy, and Stu.
ACKNOWLEDGMENTS

This dissertation started as an undergraduate research project with who was later to be my dissertation advisor, Professor Ken Stanley. Through all the successes and inevitable bumps along the way, his brilliance, enthusiasm, and optimism continued to motivate and reassure me. Thank you, Dr. Stanley for your seemingly infinite patience with me. This dissertation would not have been possible without you.

I also thank my committee members for their academic guidance through the process of completing my dissertation: Professor Annie Wu who also served on my undergraduate honors thesis committee, Professor Joe LaViola, and Professor Thad Anderson.

Thank you to Dave LaRue, my bass teacher and mentor whose faith in me is contagious. I thank him for not only teaching me crucial elements of practical music theory and bass technique, but for also inspiring my own mentoring and teaching strategies. I will miss the 1:00 pm Saturday lessons over at Bass Central.

Thank you to my Orlando family for your friendship and all of the good times: Juliet Norton, Sebastian Risi, Lisa Soros, Sarah Buchanan, Stephen Fulwider, Steven Zittrower, Joel Lehman, Ben Silvis, Dan DeBlasio, Paul Szerlip, EJ Samuel, Niki Ustjanowski, and Dave Ellis. Thank you to the rest of my family, especially Grammy Teabo, Uncle Jon and Aunt Karin Hoover, Aunt Debbye, Aunt Martha and Rob, Eryn Cook, Alison, Gabe, and Sue who remind me how proud they are of me.

Through CreativeIT funding, the MaestroGenesis project gained Paul Szerlip and several undergraduate team members. Thanks to Paul for his commitment to MaestroGenesis for being an incredible friend and research partner as we traveled the conference-circuit together these past five
years. Thank you also to Trevor Brindle, Zach Merritt, Marie Norton, and Jessica Sprague who were also dedicated to the project led to many of the results in Chapter 5 of this dissertation.

The folk MIDI's in Chapters 2 and 3 are all composed by Barry Taylor. I thank him for his permission to publish his MIDI's and for the kind words he spoke about our program.

Thank you to all past and current EPlex members not already mentioned including Greg Morse, Justin Pugh, and Drs. Erin Hastings, Jason Gauci, David D’Ambrosio, Phillip Verbancsics, Ben Jones, Brian Woolley, Andrea Soltoggio, and Jimmy Secretan for your friendship and valuable input over the years. I would also like to thank an honorary EPlex member, Dr. Charles ‘Ed’ Bailey, for generously supporting EPlex and the energy he brought to every meeting.

The work in this dissertation has been supported with a UCF Trustee’s Fellowship, a summer RA position for Mike Roberts, and the National Science Foundation through my Graduate Research Fellowship and CreativeIT grant no. IIS-1002507. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Finally, I thank my late parents for their love and support. By emphasizing my education, encouraging all of my different musical pursuits, and whatever oddball activities I wanted to try, they positioned me to succeed in life. Thanks Stu Hoover for your continued love and support and Uncle Robert and Susan Henry for supporting the Hoovers over the years. Thanks also to John Hargrove for convincing me to give computer science a shot, and supporting my progress over the years. I would especially like to thank my dad for his life wisdom, and for repeating himself so many times that I could never forget his ideas, advice, and life experiences.
# TABLE OF CONTENTS

**LIST OF FIGURES** ......................................................... xiii

**LIST OF TABLES** .......................................................... xxiv

**CHAPTER 1: INTRODUCTION** ............................................. 1

**CHAPTER 2: BACKGROUND** ............................................... 4

2.1 Previous Representations in Computer Music ....................... 4

2.2 Applications to Creativity Enhancement ............................. 5

2.3 Interactive Evolutionary Computation ............................... 7

2.4 Artificial Neural Networks ............................................. 8

2.5 CPPNs and NEAT ......................................................... 9

**CHAPTER 3: NEAT DRUMMER** ........................................... 13

3.1 The NEAT Drummer Approach ....................................... 16

3.1.1 CPPN Rhythm Generation ......................................... 16

3.1.2 Drum Pattern Interactive Evolution ............................ 22

3.1.3 Musical Instrument Digital Interface .......................... 24
3.2 Experimental Design ........................................ 24
  3.2.1 Testing the Power of Scaffolding .................. 25
  3.2.2 Comparisons ........................................... 25
3.3 Scaffolding Results: Drum Track Accompaniment .......... 27
  3.3.1 Inputting Instrument Tracks Alone .................. 28
  3.3.2 Inputting Instrument Tracks and Conductors .......... 31
  3.3.3 Evolved CPPNs ....................................... 34
3.4 Comparison Results ....................................... 36
  3.4.1 Interactive Evolution without Scaffolding .......... 36
  3.4.2 Targeted Evolution without Scaffolding ............ 37
3.5 Implications .............................................. 40
3.6 Conclusion ................................................ 41

CHAPTER 4: GENERATING MONOPHONIC ADDITIONAL VOICES FOR EXISTING COMPOSITIONS .......... 45
  4.1 Functional Relationship Representation ................. 46
  4.2 Choosing Scaffolds and Evolving Harmonies ............. 52
  4.3 Experiments ............................................. 55
4.3.1 Investigating the Evolution of Generated Voices ........................................... 56

4.3.2 Comparing FSMC-augmented Compositions to Fully Human Compositions 61

An Additional Voice for Nancy Whiskey ............................................................... 62

Musical Turing Test ............................................................................................... 65

4.4 Implications ....................................................................................................... 67

4.4.1 Simplicity of Functional Relationships ......................................................... 68

4.5 Conclusions ....................................................................................................... 69

CHAPTER 5: GENERATING POLYPHONIC ADDITIONAL VOICES FOR SINGLE MONOPHONIC STARTING MELODIES ................................................................. 70

5.1 Extending Functional Scaffolding for Music Composition ................................. 71

5.1.1 Experiments .................................................................................................... 73

Procedures for Polyphonic Generated Voices ....................................................... 73

Polyphonic Results ................................................................................................. 76

Assessing the Human Contribution ....................................................................... 77

Creative Self-Assessment ....................................................................................... 79

5.2 Implications ....................................................................................................... 81

5.3 Conclusion ......................................................................................................... 82
CHAPTER 6: MAESTROGENESIS VOICE ........................................ 83

6.1 Extending MaestroGenesis with MG-V ................................. 84

6.1.1 MeloTranscript ....................................................... 84

6.1.2 Chord Generation to Enrich the Scaffold ......................... 86

6.2 Experiments .................................................................. 88

6.3 Results ........................................................................ 89

6.4 Conclusion .................................................................... 90

CHAPTER 7: DISCUSSION AND FUTURE OPPORTUNITIES ............ 94

7.1 Implications for Musical Appreciation ................................. 94

7.2 Importance of Scaffolding .................................................. 96

7.3 Creativity Enhancement ..................................................... 97

CHAPTER 8: CONCLUSION ....................................................... 100

APPENDIX A: SURVEY QUESTIONS ........................................ 102

A.1 Perceived Quality Listener's Study in 4.3.1 ............................ 103

A.2 Musical Turing Test Study in 4.3.2 ..................................... 105

A.3 Perceived Quality Listener's Study in 5.1.1.3 ......................... 106
LIST OF REFERENCES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 108
## LIST OF FIGURES

2.1 **Example ANN for Robot Control.** Outputs $L_{\text{speed}}$ and $R_{\text{speed}}$ represent the speed of each wheel on a robot and are calculated by first finding each node’s activation sum which is then input to its activation function. The inputs to this ANN are the $(x, y)$ coordinates of the robot’s location and the radian value of its heading. ................................................................. 8

2.2 **CPPN Representation.** CPPNs like those shown in (a) generate pictures like those shown in (b) by querying each pixel position and outputting a coloring value. The functional relationship of the coloring results in visual patterns similar to musical patterns that create musical voices. .............................. 11

2.3 **CPPN-generated Patterns.** A collection of images from Picbreeder is shown that demonstrates the power of the CPPN representation. ........................................ 11

3.1 **CPPN Inputs and Outputs.** The user selects both a set of inputs from among the tracks in the MIDI file and a set of outputs corresponding to specific drums. 16

3.2 **NEAT Drummer Interface.** NEAT drummer presents an IEC interface where visual representations of drum patterns help the user to decide whether to listen to each candidate and then select their favorites. This approach, i.e. choosing inputs and outputs and selecting favorites, is designed to allow users to evolve compelling drum tracks without the need for musical expertise. 19
3.3 **Track Input Encoding.** Regardless of the instrument, each note in a sequence in any track input to the CPPN is encoded as a spike that decays over the duration of note. The pattern depicted in this figure shows how eighth notes decay faster than quarter notes, thereby conveying timing information to the CPPN, which samples this pattern at discrete ticks. The variable-intensity row of boxes under the spikes depicts the intensity of the spike sampled at discrete time steps (i.e. four per quarter note). The intensity at each timestep is represented by the darkness in its respective column, which indicates how the input track “sounds” to the CPPN at that moment.

3.4 **Potential NEAT Drummer Conductor Inputs.** Each figure depicts four measures of a conductor, which is a temporal coordinate frame optionally provided by the user to provide additional structure to the song. The simplest conductor (a) represents the current position in the song, suggesting a smooth transition across the entire song. Position-in-measure (b) allows the CPPN to know at every moment where it is within the current measure, which allows it to improvise patterns within measures and “understand” the measure structure of the song. Similarly, the time within each four-tick beat can be input as well (c). Conductors offer the user a subtle yet powerful means to influence the overall structure of the rhythm without the need for note-by-note specification.
3.5 **Johnny Cope Results.** Results are depicted from two different generations at two different parts of the song. The inputs from the original song, which are always the same, are shown at bottom. Note the relationship between the inputs and the outputs, and between the first generation and the eleventh, which elaborates on the former. The motif in measures three and four is typical of the first part of the song until measure 23, when it switches to a different motif in both generations. Thus, the figure gives a sense of the two predominant drum riffs exhibited in both generations. The main conclusion is that the output is a function of the input that inherits its underlying style and character. (These tracks are available at [http://eplex.cs.ucf.edu/fsmc/dissertation/](http://eplex.cs.ucf.edu/fsmc/dissertation/)).

3.6 **Oh! Susanna Outputs.** This pattern from measures three through six of Oh! Susanna is from generation 25 of evolution. The network evolved 15 hidden nodes and 41 connections. Near the end of measure four is a particularly improvisational riff in the snare that transitions to measure five. This riff is caused by variation in the piano and other inputs at the same time. As with Johnny Cope, the drum pattern sounds natural and styled correctly for this upbeat song. (This track is available at [http://eplex.cs.ucf.edu/fsmc/dissertation/](http://eplex.cs.ucf.edu/fsmc/dissertation/)).
3.7 **Oh! Dem Golden Slippers with and without Conductors.** Output drum patterns are shown for the song in one case when no conductor is input (a) and in the other where a position-in-song conductor is input (b). The difference in resultant drum patterns shows that the conductor imposes a temporal progression on the drum track that does not derive from the structure of the song itself, demonstrating the power of conductors to subtly shape the structure of music. Finally, when conductors indicating both position-in-song and position-in-measure are input simultaneously (c), progression is enhanced both throughout the song and within each measure.

3.8 **Complex Conductor.** The conductor, which follows the pattern quarter quarter half, is shown at bottom. Two three-part drum tracks that are functions of this conductor are shown above it. While both drum tracks are different, they are also both constrained by the underlying motif of the conductor.

3.9 **CPPN Drum Track Generators.** The evolved CPPNs that produce every drum track shown in Sections 3.2.1 and 3.2.2 are depicted. While the complexities vary, the quality of the output is similar because each produces a function of a preexisting song, thereby inheriting its qualities. Activation functions are denoted by $S$ for sigmoid, $M$ for multiplication, $G$ for Gaussian, and $L$ for linear.
3.10 **Typical Best Results from Interactive Evolution Without Scaffolding.**

Best results from both types of representations tested are depicted. The feed-forward network only inputs position-in-song and produces unremarkable gradient patterns that do not follow the structure of music nor the Johnny Cope song. The recurrent network produces repeating motifs, but they are not synchronized with the measure and they do not vary with the song. In this way, removing the scaffold removes a major advantage of NEAT Drummer.

3.11 **Typical Best Results from Target-based Evolution Without Scaffolding.**

Feedforward (a) and recurrent (b) results are depicted and the target pattern is shown in (c). The aim was to match the target. As this figure shows, neither variant successfully matched the target (which was evolved in NEAT Drummer in 11 generations) after 1,600 generations, although the recurrent variant evolved more complex patterns. This result confirms again the importance of the scaffold.

3.12 **Typical Result from Target-based Evolution with Position-in-song and Position-in-measure Inputs.** The improvement in pattern complexity, and the adherence to measure structure, are apparent in comparison to figure 3.11a. By providing position-in-measure as input, evolution can easily produce patterns that follow the timing of measures, demonstrating the power of conductor inputs. However, without the scaffolding inputs from Johnny Cope, the drum pattern still does not match the target despite its regular structure.
3.13 **Fitness Over Time of the Three Variants in Target-based Evolution.** The increase in fitness over 1,600 generations of evolving the three variant representations is shown, averaged over 20 runs for each. A perfect fitness of 1.0 would mean the target is matched perfectly. None of the variants exceed 0.7 fitness because the search is too unconstrained without the scaffold.

4.1 **How CPPNs Compute a Function of the Input Scaffold.** The rhythm CPPN in (a) and pitch CPPN in (b) together form the generated voices of FSMC. The inputs to the CPPNs are the scaffold rhythms and pitches for the respective networks and the outputs indicate the rhythms and pitches of the generated voices. Each rhythm network has two outputs for each instrument: OnOff and NewNote. The OnOff node controls volume and whether or not a note is played. The NewNote node indicates whether a note is re-voiced or sustained at the current tick. If OnOff indicates a rest, the NewNote node is ignored. The pitch CPPN output decides what pitch the generated voice should play at that particular tick. The internal topologies of these networks, which encode the functions they perform, change over evolution. The functions within each node depict that a CPPN can include more than one activation function, such as Gaussian and sigmoid functions. Two monophonic generated voice outputs are depicted, but the number of instruments a CPPN can output is unlimited. The number of input instruments also can vary.
4.2 **Pitch CPPNs over Two Generations** The pitch CPPNs in (a) and (b) illustrate how scaffolds are transformed to musical outputs. Each of the four identical CPPNs in (a) and the four identical CPPNs in (b) represent a calculation made at the four quarter-note-length ticks in this one measure scaffold. The CPPN in (a) is from the first generation and has yet to evolve hidden nodes, while the CPPN in (b) from the second generation has evolved an hidden node between the bias and output and mutated the existing connection weights.

4.3 **CPPN Input Representation.** The spike-decay representation for rhythms is shown in (a) and the pitch representation is shown in (b). Note that spike-decay model is identical to that seen in Chapter 3. Both such inputs are depicted in two ways: The first is a continuous-time graph that shows decaying spikes for rhythm and the pitch level for pitch. The second is a standard musical notation representation of the scaffold. In this way, this figure gives a sense of exactly what the CPPN “hears” (for each instrument in the scaffold) as it generates additional voices.

4.4 **Representing the Scaffold.** This generated voice for Scarborough Fair (top) is generated from the four instruments in the scaffold (bottom). Each of these instruments is input to both the rhythmic CPPN (middle left) and the pitch CPPN (middle right) using the input representation shown in figure 4.3. This example can be heard at [http://eplex.cs.ucf.edu/fsmc/dissertation/](http://eplex.cs.ucf.edu/fsmc/dissertation/).
4.5 **MaestroGenesis Graphical User Interface.** Generated voices in FSMC are presented both visually and sonically. Unlike image evolution, in which users can quickly evaluate the population [79], listening to MIDIs takes time. The visual representations help users decide which MIDIs are worth this extra time, thus speeding up evolution. There are ten individuals in a population, which are all displayed at once. The user rates individuals by giving them a thumbs-up or thumbs-down.

4.6 **Evolutionary Sequence of Generated Voices for Bad Girl’s Lament.** Three measures of the evolved steel guitar voice from generations 1, 6, and 12 of Bad Girl’s Lament is shown at top, followed by the pitch and rhythm inputs to the CPPN from the scaffold. In this experiment, the tick length or smallest rhythmic unit is a sixteenth note. Each generated voice can be heard at [http://eplex.cs.ucf.edu/fsmc/dissertation/](http://eplex.cs.ucf.edu/fsmc/dissertation/). The type of instrument played in the scaffold is noted at left (e.g. a harpsichord is one track in the rhythm scaffold). The increase in harmonic and rhythmic sophistication between generations 1 and 12 is apparent in the progression at top. In addition, the relationship (e.g. in note transition points) between the scaffold and generated voice can also be observed.
4.7 **FSMC-Generated Additional Voices for Nancy Whiskey.** Measures 5 and 6 of a generated voice evolved for Nancy Whiskey is shown. The top line shows the generated harpsichord output while the next three voices show the fiddle, steel guitar, and bass respectively. The generated voice is produced by CPPNs that relate the scaffold inputs to the generated outputs. The rhythm and pitch relationships between the scaffold and generated voice derive from the functional relationship between them. The 12th-generation additional voice evolved for Bad Girl’s Lament, also addressed in this section, was shown in figure 4.6. Both generated voices can be heard at [http://eplex.cs.ucf.edu/fsmc/dissertation/](http://eplex.cs.ucf.edu/fsmc/dissertation/). 63

4.8 **CPPNs Corresponding to Generated Voices.** CPPNs for generating additional voices for Nancy Whiskey (a and b) and Bad Girl’s Lament (BGL; c and d) are shown above. Each generated voice can be heard at [http://eplex.cs.ucf.edu/fsmc/dissertation/](http://eplex.cs.ucf.edu/fsmc/dissertation/). Line thickness is proportional to weight and gray lines are inhibitory. The simplicity of the CPPNs shows that the functional relationship between different musical parts is often plausible even if it is simple. 64

5.1 **Early One Morning Versions.** The first four measures of versions 1 and 2 of Early One Morning illustrate how a single user with the same monophonic starting melody can direct the generated voice in two different ways that nevertheless both relate to the initial melody. Because the generated voices share two of their layers, they sound related. However, through timbre selection and the evolution of two and three distinct layers in versions 1 and 2, respectively, the user imparts a different feel. 76
5.2 **Listener Study Results.** The average rating (by 129 participants) from one to ten of both the collaborative and FSMC-alone arrangements are shown side-by-side with the lines indicating a 5% error bound. The overall results for the listener study indicate that on average the collaborative arrangements are of significantly higher perceived quality than FSMC-alone.

6.1 **Singing into MaestroGenesis.** By pressing the microphone at the top of the screen in (a), users begin recording their song as represented by the waveform rectangle-visualization below it. The rectangles scroll as the song continues, minimize when the singer rests, and grow with the input volume. In (b) the MIDI transcription is shown with the waveform where a gray sliding bar indicates the current position in both the waveform and the MIDI throughout playback.

6.2 **First Fourteen Measures of Lyrics for For Today.** Users begin composing with MaestroGenesis by singing into MG-V. This scaffold is a song selected by a MaestroGenesis team member, undergraduate Jessica Sprague, called For Today by Jessica Lea Mayfield and is shown with her permission. Both the MIDI transcription and raw audio are available at http://eplex.cs.ucf.edu/fsmc/dissertation.

6.3 **First Fourteen Measures of Lyrics and Generated Chords for For Today.** In MG-V, users choose the chords to accompany their audio. The raw audio, its MIDI transcription, and the generated chord sequence are available at http://eplex.cs.ucf.edu/fsmc/dissertation.
6.4 **Generated Voices from Monophonic Scaffold.** Measures sixth through nine in (a), (b), and (c) typify three response types articulated by the generated voice when encountering the measure-long rest in the scaffold (at measure seven). In (a) the additional voice rests, in (b) the note generated for the tonic of the key (B Major) is the only pitch available and sustained through the rest, and in (c) note generated for the tonic of the key is articulated throughout the measure.

6.5 **Additional Generated Voice with Chord Progression.** In measure 7, a rest is encountered in the scaffold that would typically leave little room for variation in the additional generated voice. However, because of the chord progression in the scaffold, at measure 7 the additional voice is able to play a different note than the one previously played before the rest began. This result with an additional generated voice is available at [http://eplex.cs.ucf.edu/fsmc/dissertation](http://eplex.cs.ucf.edu/fsmc/dissertation).
LIST OF TABLES

4.1 **Perceived Quality by Survey Participants.** This table shows the average ratings and the mean and standard deviation for the control and four Bad Girl’s Lament (BGL) MIDIs. The MIDI names are on the left while the average ratings are on the right. .......................................................... 60

4.2 **Survey Results (lower means more human-like).** The average ratings and standard deviations for the samples show that FSMC generated voices can sound human. The Bad Girl’s Lament MIDI, which is partly computer-generated, ranks less likely to be computer-generated than the fully human-composed song, Kilgary Mountain, although this difference is not significant. 66

6.1 **Chord Transitions.** Because certain chords sound better following others, a transition table of valid movements is established through music theory. The left hand side represents the current chord, while each list on the right contains possible choices for the next. Because MG-V requires that a measure must share a chord tone with the suggested chord, the chords shown are a proper subset of those available in the program. Transition probabilities are equally weighted. .......................................................... 87
CHAPTER 1: INTRODUCTION

An important aspect of any creativity-enhancing algorithm for musical composition is the ability to facilitate exploring beyond the user’s usual boundaries. While there exist programs that promote compositional creativity, many focus on producing a particular feel or style at the cost of restricting creative exploration. Because a desired structure is emphasized in such programs, the musical spaces that they explore are defined by formalized rules that constrain the results to particular styles and genres [90, 13, 20].

Because music is inherently grammatical (i.e. hierarchical; [48, 61, 53, 78, 15]), such musical structure is often computationally imposed by representing music through a grammar [19, 45, 54, 92]. Temporal relationships are then often captured through Markov Models, Hidden Markov Models (HMMs), or n-grams [17, 93, 60, 12]. However, the success of these approaches depends on statistical rules and relationships that researchers extract from carefully crafted databases.

In contrast to these approaches, this dissertation introduces a new representation for computer-assisted musical composition that includes almost no built-in knowledge of musical structure yet still helps users compose starting from as little as a single, preexisting monophonic melody, or scaffold. This approach, called functional scaffolding for musical composition (FSMC), exploits two rarely explored mathematical properties of music: that music can be represented as a function of time [63] and that multiple parts in a coherent piece must be functionally related to each other [33, 36, 34, 35, 31, 32]. 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 a computer-generated voice through a neural-network-like representation called a compositional pattern producing network (CPPN), which produces an additional computer-generated voice for the scaffold as an output [77]. 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 complement of the other.

To implement this idea in practice, I co-created two programs: NEAT Drummer, a program introduced to allow users to explore a space of drum pattern accompaniments composed by CPPN-based transformations [31, 32], and its successor MaestroGenesis (freely available at http://maestrogenesis.org), which helps users explore the space of rhythmic and melodic additional generated voices [33, 36, 35, 34]. Both programs help users navigate the space of possible transformations (i.e. generated voices) by having the user sing a melody or input a MIDI and then presenting a set of candidate computer-generated voices to be played simultaneously with the input. Users then choose the best additional voices that new set of candidate computer-generated voices then inherit some of the appealing traits of the chosen generated voices. This process, called interactive evolutionary computation (IEC) [79], 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), through which music-generating CPPNs can gradually increase in complexity [75], thereby granting users the ability to explore the necessary complexity of the relationship between scaffold and additional voice.

The overarching hypothesis for this work is that establishing a functional relationship alone is sufficient to generate plausible musical voices and to complete incomplete songs, even without any other musical principles or theory assumed by the system. The hypothesis is addressed through several major contributions. First (1), an important first step in validating FSMC was to determine the importance of the scaffold. That is, the question is whether CPPNs create musical voices by transforming the essential human essence from existing compositions or whether they can function on their own. Results with NEAT Drummer showed that the contribution of the scaffold is indeed essential to FSMC [31].
Because FSMC (which steps beyond NEAT Drummer to full harmonization) emphasizes the importance of functional relationships between parts of a song, it is important that MaestroGenesis can create high-quality additional generated voices through functional transformations. Indeed, results reported here illustrate that listeners without special musical expertise cannot distinguish between fully human-composed pieces and those with additional voices generated by MaestroGenesis [34, 35]. The high quality of these additional computer-generated voices suggests that not only can music be generated through functional transformations, but also that functional relationships may be an important implicit component of human music composition and appreciation.

Through FSMC (3), MaestroGenesis not only helps users compose music but can also facilitate their creativity. A further set of results indicates that with MaestroGenesis, users can easily construct and explore a space of plausible computer-generated musical voices that increase in quality over generations. An additional study (4) also explores the creativity-enhancing effects that amateur musicians experience with MaestroGenesis and examines how users integrated MaestroGenesis into their own compositions [36].

Finally (5), to further facilitate creativity enhancement, a MaestroGenesis plugin is developed that allows users to elaborate melodies composed from a user’s own lyrics and voice, thereby eliminating the need for initial MIDI composition. Initial results illustrate the feasibility of incorporating the human voice.

The next chapter provides relevant background while Chapters 3, 4, and 5 focus on the FSMC method, experiments, and results with NEAT Drummer and MaestroGenesis. The MaestroGenesis plugin MaestroGenesis Voice (MG-V) is described in Chapter 6 along with experiments showcasing polyphonic pieces that can be composed from raw audio of a human voice. Implications the completed work are discussed in Chapter 7 and the dissertation concludes in Chapter 8.
CHAPTER 2: BACKGROUND

Much of the expressive potential of computer-generated music derives from the power of chosen musical representation. This chapter discusses prior approaches and representations in computer-generated music.

2.1 Previous Representations in Computer Music

Many musical representations have been proposed before FSMC, although their focus is not necessarily on representing the functional relationship between parts. For example, from long before FSMC, Holtzman [30] describes a musical grammar that generates harp solos based on the physical limitations imposed on harp performers. Similarly, Cope [20] derives grammars from the linguistic principles of haiku to generate music in a particular style. These examples and other grammar-based systems are predicated on the idea that music follows grammatical rules and thus by modeling musical relationships as grammars, they are capturing the important structures of music [66, 54]. While grammars can produce a natural sound, deciding which aspects of musical structure should be represented by them is often difficult and ad hoc [46, 52].

An alternative to manually constructing grammars is to discover important musical relationships through statistical analyses of musical corpora that then guide decision-making. For example, Ponsford, Wiggins, and Mellish [62] build generative grammars by deducing important harmonic movement rules from a carefully constructed and annotated corpus of 84 seventeenth-century dance pieces. While this method could potentially generate music in a variety of musical styles, database requirements restrict its extensibility. For example, the dance music was chosen for its particular properties: (1) well-defined rules for composition, (2) moderate simplicity, and (3) natural focus
on notating harmonic movement (i.e. the style leaves the melody composition to the performer). However, most musical genres are not so well defined. Gillick, Tang, and Keller [28] emulate jazz musicians by manipulating and clustering the training data itself. From as little as a single song, they abstract contour shapes, group measures with similar properties, and build probability matrices to determine the order of measures in a given piece. While these operations generate cohesive solos, the lack of global structure in the generated pieces confines music to less structured genres. Nevertheless, these works represent significant contributions toward understanding how music can be generated by computers and illustrate the research community’s convergence towards grammatical representations. The next section discusses applications of these representations to creativity enhancement.

2.2 Applications to Creativity Enhancement

There have been many different approaches to incorporating computer-generated music into creativity-enhancing software. The grammar-based creativity-enhancing program Impro-Visor helps users create monophonic jazz solos by automatically composing any number of measures in the style of famous jazz artists [45]. Styles are represented as grammars that the user can invoke to complete compositions. Creativity-enhancement in Impro-Visor occurs through the interaction of the user’s own writing and the program’s suggestions [45]. When users have difficulty elaborating musical concepts, they can access predictions of how famous musicians would approach the problem within the context of the current composition. By first learning different professional compositional techniques, students can then begin developing their own personal styles. While Impro-Visor is an innovative tool for teaching jazz styles to experienced musicians, it focuses on *emulating* prior musicians over exploration of a new sound.

MySong generates chord-based accompaniment for a vocal piece from hidden Markov models

5
Users select any vocal piece and MySong outputs accompaniment based on a transition table, a weighting factor that permits greater deviation from the table, and musical style (e.g. rock, big band). MySong thus allows users to create accompaniment for their own melodies in a variety of different predefined styles from which users cannot deviate. While these models yield notable results, they require a significant database of specific examples that must be carefully constructed by the programmers.

Some programs offload the responsibility of rule and database construction to the user. Zicarelli [94] describes an early interactive composition program, Jam Factory, that improvises on human-provided MIDI inputs from rules represented in transition tables. Users manipulate the output in several ways including changing the probability distributions of eight different transition tables; there are four each for both rhythm and pitch. Users can gain more creative control by designing and consulting the transition tables, but the increased flexibility may result in unnatural outputs that thereby limit the utility of the main algorithms [95]. The approach described by Chuan [13] balances user control by training transition tables based on only a few user-provided examples. The tables then reflect the “style” inherent in the examples and can generate chord-based accompaniment for a user’s own piece.

In contrast to works that depend upon specific rules or trained transition tables, the aim in this dissertation is to exploit very general, high-level principles that can be applied across a broad range of compositions and styles. At the same time, the user will register his or her own preferences through the process described next that is designed to extract the intent of non-experts without the need for musical expertise.
2.3 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 79, 40, 57, 6, 16, 31, 85). The idea is that humans, rather than hard-coded rules, can rate candidate computer-generated voices in place of an explicit fitness function. IEC originated in Richard Dawkins’ book, The Blind Watchmaker, in which he described a simple program called Biomorphs that is meant to illustrate evolutionary principles [25]. The program displays a set of several pictures (called Biomorphs) on the screen at one time. The user then selects from among those pictures (called the population) his or her favorite. From that selection, a new generation of offspring is spawned that replace 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.

IEC can encompass a variety of digital media [68, 79], including images [51, 91, 29], movies [87], three-dimensional models [58], and music [40, 57, 6, 16, 31, 85]. In GenJam a human player and computer “trade fours,” a process whereby the human plays four measures and the computer “answers” them with four measures of its own [5]. Musical propositions are mutated and combined into candidates that the user rates as good or bad. Similarly, Jacob [38] introduces a system in which human users rate, combine, and explore musical candidates at three different levels of the composition process and Ralley [64] generates melodies by creating a population from mutations of a provided user input. Finally, CACIE creates atonal pieces by concatenating musical phrases as they are generated over time [1]. Each phrase is represented as a tree structure that users can interactively evolve or directly manipulate. Most such systems impose explicit musical rules conceived by the developer to constrain the search spaces of possible musical voices, thereby narrowing the potential for discovery. In contrast, the aim in this dissertation is to explore less
2.4 Artificial Neural Networks

The representation that underlies generated music in this dissertation is a variant of artificial neural networks (ANNs), which are therefore reviewed in this section. Because the human brain is the working model that many machine learning approaches aim to emulate, many abstractions of it exist [9, 82]. ANNs abstract the neural connectivity of the brain can also theoretically approximate any continuous function [22]. They have achieved successful results in a variety of domains including robot control [50, 86, 55], speech recognition [88, 10, 23], and classification [65, 14].
ANNs contain a variable number of sensor nodes, potential hidden nodes often represented by sigmoid functions, and output nodes. The sensors collect information from the environment that is then sent to hidden nodes (if any) and subsequent output nodes over connection weights. Figure 2.1 illustrates by example how ANNs can determine behavior.

In this example sensory information is derived from the robot’s current location determined by the \((x,y)\) coordinates in the plane and from the robot’s heading for a total of three inputs. The two outputs \(L_{\text{speed}}\) and \(R_{\text{speed}}\) correspond to the requested speed of each wheel, which determines its speed and heading. The hidden nodes are \(h_1\), \(h_2\), and \(h_3\). Weights between node \(i\) and node \(j\) are written as \(w_{i,j}\). To calculate the output values, it is first necessary to determine the values of the hidden nodes and their activation levels. For example, the activation level for \(h_1\) is \(a_1 = x w_{x,h_1} + y w_{y,h_1} + \theta_{in} w_{\theta_{in},h_1}\). The output for \(h_1\) is then computed by the sigmoid function \(\frac{1}{1+e^{-a_1}}\).

Outputs for each of the nodes are calculated in this way until final outputs are determined and converted to their respective actions. Thus potentially complex behaviors and functions can be determined by a single ANN.

2.5 CPPNs and NEAT

FSMC creates computer-generated voices by “listening” to the original parts in a MIDI composition simultaneously and transforming these parts into an additional musical voice that follows the contours of the original song. This transformation occurs through a variant of an ANN called a compositional pattern producing network (figure 2.2a).

The CPPN is a network of interconnected nodes similar to ANNs. However, unlike traditional ANNs, each node in a CPPN can compute a different type of function (e.g. Gaussian, sigmoid, linear, sine, multiplicative, etc.), thereby biasing the search space toward results with particular
regularities. For instance, Gaussians tend to generate bilateral symmetry while sine hidden nodes suggest repeating patterns. These nodes are arranged within an arbitrary topology and, as in an ANN, the nodes send their outputs over outgoing connections by multiplying their outputs by the connection weights. In effect the CPPN is like an ANN with multiple possible activation functions that can be expressed within the same network.

CPPNs are chosen for FSMC precisely because they are suited to producing patterns with regularities. They 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). While CPPNs generate music in this dissertation, their pattern-generating capabilities were originally demonstrated through generating images (i.e. spatial patterns). To give an intuitive idea of how a CPPN can generate such a pattern, figure 2.2 shows how CPPNs transform pixel positions into a collection of shadings that paint images. The \((x,y)\) position of each pixel on the canvas in figure 2.2b is input to the CPPN in figure 2.2a to produce a shading. Figure 2.2b shows how these shadings are collected over the canvas to create complete pictures. As shown in figure 2.2b, the resultant patterns exhibit regularities, as should musical sequences as well.

Just as music represented by CPPNs will be interactively evolved in this dissertation, images input by CPPNs can also be interactively evolved as demonstrated by the Picbreeder website [39]. In this approach, users are presented with a collection of images and select those with the most appealing traits. These images are then combined and mutated with an algorithm called NeuroEvolution of Augmenting Topologies (NEAT; [75]) to produce a new generation of offspring images resembling the parents. NEAT incrementally evolves CPPNs by gradually adding nodes and connections. Figure 2.3 shows a collection of images created through this process that exhibit the breadth of pattern-generating capabilities of this representation.

The next chapters introduce NEAT Drummer [31] and MaestroGenesis, which show that such a
Figure 2.2: **CPPN Representation.** CPPNs like those shown in (a) generate pictures like those shown in (b) by querying each pixel position and outputting a coloring value. The functional relationship of the coloring results in visual patterns similar to musical patterns that create musical voices.

Figure 2.3: **CPPN-generated Patterns.** A collection of images from Picbreeder is shown that demonstrates the power of the CPPN representation.
generic pattern-generating capability can indeed be applied to musical patterns. Users of these programs select the generated voices with the most appealing musical qualities from a set of candidates and evolve the pieces until the voice is satisfactory. NEAT Drummer demonstrates this idea with percussion and MaestroGenesis with full harmonization.
A most intriguing capability of human composers is that they can often quickly conceive multiple instrumental parts simultaneously during the creative process. For example, Mozart could hear complex multipart pieces form “in his head,” suggesting a powerful creative mechanism for generating additional musical voices [69, 26, 37]. Relatedly, rock guitarists in jam sessions and jazz musicians can improvise together while simultaneously perfectly respecting the interdependencies of their separate parts [3, 4, 44, 59, 70, 89]. Thus although intuition may suggest that complex interdependent constructions should require care and labor to devise, in fact such constructions in music appear almost effortless. Thus it is likely that no explicit serial reasoning is involved in the creative construction of computer-generated instrumental tracks. What kind of mechanism then is responsible for such a capability?

This chapter suggests a possible high-level answer to this question. The key idea is that different instrumental parts are functionally related, which means that one can be expressed as a function of another. Furthermore, although we may perceive the interplay between two or more simultaneous instruments as rich and complex, in fact the function that relates one to the other can be quite simple. Thus, in this view, once a single track such as a melody is created, it can serve as a scaffold, i.e. an existing support structure, upon which other tracks are generated. In this way, while composers may seem to improvise entire harmonies and drum tracks one note at a time, fundamentally they need only construct a simple function for each part that transforms the scaffold. In fact, because the scaffold already in effect embodies the intrinsic contours and complexities of the song, any transformation of the scaffold inherits these features and thereby embodies the same thematic elements automatically. Thus the space of possible transforming functions is highly forgiving, in part explaining why improvising musical voices can appear effortless. As long as the generated voice is expressed as a function of the scaffold, it is difficult to go significantly wrong.
While this idea of functions relating one pattern to another is difficult (though not necessarily impossible) to confirm at a neurological level, it does suggest a promising model for computer-generated music. This chapter describes the implementation of such a model focused on generating percussion and presents its results. The goal is to generate drum tracks to accompany existing songs. Because rhythm is simpler than melody or harmony, rhythm generation is an appealing stepping stone to full blown harmonization. It effectively highlights the advantages of the functional perspective in clear and simple terms that do not require musical expertise to appreciate.

The idea is implemented in this chapter in a program called NEAT Drummer, which automatically generates drum tracks for existing songs. It accepts existing human compositions as input to CPPNs and outputs drum patterns to accompany the instruments. The inputs to NEAT Drummer are specific parts of a Musical Instrument Digital Interface (MIDI) file (e.g. the lead guitar, bass guitar, and vocals) and the outputs are drum tracks that are played along with the original MIDI file. That way, outputs are a function of the original MIDI file inputs, forcing synchronization with the MIDI. To take into account the user’s own inclinations, NEAT Drummer allows the user to interactively evolve rhythms from an initial population of drum tracks with the NeuroEvolution of Augmenting Topologies (NEAT) algorithm [75], which evolves increasingly complex CPPN-encoded patterns.

The main results are drum tracks for existing songs that tightly follow the contours and idiosyncrasies of individual pieces, yet elaborate and elucidate them in creative and unexpected ways. Even when major transitions occur, because the drum tracks are a function of the music, the drums perfectly follow the transitions.

This functional model of musical composition is then further extended to allow human users to add their own functional influences to create variational motifs outside the confines of the provided song. For example, users can provide a monotonically increasing function (i.e. time), which
suggests change over time even if the underlying scaffold is repetitive. The result is that the drum track can be made to vary exactly as the user requests, while still seamlessly interweaving with the song. These user-provided functions are called *conductors* in a loose analogy with orchestral conductors, who describe functional contours with their hands that the orchestra follows. The conductors further highlight the simplicity and relative ease of creating subtle overlapping textures through simple functional relationships.

To highlight the importance of scaffolding and conductors, several variants of NEAT Drummer without such facilities are compared with NEAT Drummer with its full functionality intact. The result is that the consequent capabilities are significantly impoverished, demonstrating the critical role that scaffolding plays in generating drum pattern accompaniment.

The main contribution of this chapter is to introduce the key idea of functional scaffolding in a relatively simple context while the high-level insight into the functional relationship essential to improvisational musical voices provides a clue to how such mechanisms may work in the brain. Because NEAT Drummer can generate natural sounding rhythms, this chapter lays the foundation for Functional Scaffolding for Musical Composition (FSMC), which facilitates creation of rhythmic and pitched additional generated voices.

The research in this chapter was originally published in [32] and [31]. All of the folk MIDIs in this chapter and Chapter 4 were sequenced without drum pattern accompaniment by musician Barry Taylor and are provided with his permission: (1) Johnny Cope, (2) Oh! Susanna, and (3) Oh! Dem Golden Slippers, (4) Nancy Whiskey, (5) Bad Girl’s Lament, (6) Kilgary Mountain, (7) Chief Douglas’ Daughter.
Figure 3.1: **CPPN Inputs and Outputs.** The user selects both a set of inputs from among the tracks in the MIDI file and a set of outputs corresponding to specific drums.

3.1 The NEAT Drummer Approach

The main idea in NEAT Drummer is that the temporal patterns of the instrumental parts of a song can be inherited by the drums by making the drums a function of the other instruments. This section begins by explaining how CPPNs encode rhythm and then details how they are evolved interactively.

### 3.1.1 CPPN Rhythm Generation

NEAT Drummer begins by generating an initial set of original drum tracks for a provided song. To initiate this first generation, the user must first specify the inputs and outputs of the CPPN (figure 3.1).

The *inputs* are individual instrumental tracks from the chosen song and the *outputs* are a set of drums that together produce the entire drum accompaniment.

From the inputs the CPPN derives its original patterns, which are therefore *functions* of the original
song (i.e. the scaffold) and its structure. In other words, NEAT Drummer generates a rhythm that is a function of these inputs. Thus, it is important to choose instruments that play salient motifs in the song so that the drum pattern can be derived from the richest structures available. Further texture can be achieved by inputting more than one MIDI track, e.g. bass and guitar.

Thus the user selects any combination of tracks representing individual instrumental parts from a MIDI file to be input into the CPPN. In this way, NEAT Drummer generates rhythms from any MIDI file.

The user also chooses the percussion instruments that will play the rhythm. Each such instrument is represented by a single output on the CPPN. For example, one output may be a bass drum, one a snare, and the final a hi-hat. Any number of drums, and hence any number of outputs, are permissible.

To produce the initial patterns, a set of random initial CPPNs with a minimal initial topology (following the NEAT approach) and the chosen inputs and outputs are generated. The number of inputs in these initial topologies corresponds to the number of instrument tracks in the scaffold (e.g. guitar, bass, etc.) plus a bias node. The relationship between the initial topology and the original song is thus established through these inputs, which feed information from the scaffold directly into the network. The number of outputs equals the number of drums in the drum ensemble. The initial minimal topology has random connectivity yet always contains exactly one hidden node. This single hidden node ensures that initial patterns sound more interesting than perceptrons, but are still relatively simple. Note that the internal topology is thus unrelated to the scaffold except insofar as it is affected by the number of inputs. Thus the apparent “knowledge” of the provided song in the pattern output by the network is entirely a result of computing a function of the scaffold.

NEAT Drummer then inputs the selected tracks into the CPPN over the course of the song in sequential order and records the consequent outputs, which represent drums being struck.
ically, from time $t = 0$ to $t = l$, where $l$ is the length of the song, the inputs are provided and outputs of the CPPN are sampled at discrete subintervals (i.e. ticks) up to $l$.

Individual notes input into the CPPN from selected MIDI tracks are represented over time as spikes that begin high and decrease (i.e. decay) linearly (figure 3.3). The period of decay is equivalent to the duration of the note. That way, the CPPN “hears” the timing information from the supplied tracks while in effect ignoring pitch, which is unnecessary to appreciate rhythm (pitch is considered later by MaestroGenesis). By allowing the spikes to decay over their duration, each note becomes a kind of temporal coordinate frame. That is, the CPPN in effect knows at any time where it is within the duration of a note by observing the stage of its decay. That information allows it to create drum patterns that vary over the course of each note.

Figure 3.2 shows the NEAT Drummer interface that visually represents generated drum patterns and input tracks from MIDI scaffolds. Each individual is presented as a set of rectangles displaying both the inputs (e.g. bias, track 1) and the outputs (e.g. bass drum, hi-hat, snare). Each rectangle represents a quantization called a tick, which can be adjusted from quarter notes to eighth notes. These ticks are concatenated together to form the measures that comprise the piece. The darkness of each tick indicates volume.

Interestingly, it is potentially useful also to input temporal patterns that are not part of the song itself. Such patterns can provide additional structure to the drums by situating them within coordinate frames that describe how the user wants the song to vary at a meta-level. For example, inputting a simple linear function of time that indicates the position-in-song at each tick (figure 3.4a) in addition to the instrument tracks means that the output is a function of both the song itself and the position-in-song. That way, the CPPN can produce a drum track that shifts gradually from one motif to another over the course of the song.
Figure 3.2: **NEAT Drummer Interface.** NEAT drummer presents an IEC interface where visual representations of drum patterns help the user to decide whether to listen to each candidate and then select their favorites. This approach, i.e. choosing inputs and outputs and selecting favorites, is designed to allow users to evolve compelling drum tracks without the need for musical expertise.
Figure 3.3: **Track Input Encoding.** Regardless of the instrument, each note in a sequence in any track input to the CPPN is encoded as a spike that decays over the duration of note. The pattern depicted in this figure shows how eighth notes decay faster than quarter notes, thereby conveying timing information to the CPPN, which samples this pattern at discrete ticks. The variable-intensity row of boxes under the spikes depicts the intensity of the spike sampled at discrete time steps (i.e. four per quarter note). The intensity at each timestep is represented by the darkness in its respective column, which indicates how the input track “sounds” to the CPPN at that moment.

Similarly, by inputting *position-in-measure* (figure 3.4b) or *position-in-beat* (figure 3.4c), the user can bias the output towards progressions across each measure or beat.

In this dissertation, these additional inputs are called *conductors* to make a metaphor with the silent patterns expressed to an orchestra by its conductor. Additional inputs that represent desired hidden contours beyond the pattern of the instruments themselves give the user an unprecedented level of control over the nuances of the global output pattern.

In fact, any arbitrary sequence can be input as a conductor, which in effect simply means a set of note spikes that are never actually heard. Thus the pattern in figure 3.3, while introduced as an instrumental pattern, could also be a complex conductor pattern that suggests a particular underlying motif that the drums should elaborate. Note that in NEAT Drummer, by convention, conductor inputs that represent time are spikes that start low and *attack*, which conveys the idea of a timing
Figure 3.4: **Potential NEAT Drummer Conductor Inputs.** Each figure depicts four measures of a conductor, which is a temporal coordinate frame optionally provided by the user to provide additional structure to the song. The simplest conductor (a) represents the current position in the song, suggesting a smooth transition across the entire song. Position-in-measure (b) allows the CPPN to know at every moment where it is within the current measure, which allows it to improvise patterns within measures and “understand” the measure structure of the song. Similarly, the time within each four-tick beat can be input as well (c). Conductors offer the user a subtle yet powerful means to influence the overall structure of the rhythm without the need for note-by-note specification.

Unlike CPPN inputs, the level of each CPPN output is interpreted as the *volume* (i.e. strength) of each drum strike. That way, NEAT Drummer can produce highly nuanced effects through varying softness. Two consecutive drum strikes one tick after another are interpreted as two separate drum strikes (as opposed to one long strike). To produce a pause between strikes, the CPPN must output an inaudible value for some number of intervening ticks. Because the CPPN has one output for each drum, the end result of activating the network over $t$ ticks is a drum sequence for each drum in the ensemble.

An interesting aspect of this representation is that it does not make explicit use of recurrent con-
nections. While recurrent networks are often noted for their ability to encode temporal patterns [27, 84, 11], it is easier to simply express music as a function of an existing temporal pattern (i.e. the melody and harmony) and thereby affix one pattern to another without needing to learn the temporal synchronization itself. Thus, while recurrence is well suited to temporal problems in which the inputs are not known a priori, because music is deterministic, recurrence is unnecessary; because the inputs are always the same, the outputs can simply be expressed as a function of the inputs. Thus the CPPN can potentially represent that function without recurrence.

NEAT Drummer generates each of the individuals in the initial population with the same set of inputs and outputs. However, the initial CPPN weights and activation functions for each member of the population are decided randomly. In particular, every input is connected to every output with a randomized weight within \([-2, 2]\). The activation function of each node is assigned randomly from among the following options: sigmoid, binary threshold, Gaussian, linear, multiplication, and sine. To encourage interesting patterns in the initial generation, a single hidden node with a random activation function is also connected into the network by splitting a randomly chosen connection. Each song is divided into ticks (four per beat). At each tick, the vector of note spike values at that discrete moment of time for all the instruments is input. The CPPN is fully activated and the value of each drum output is recorded so that all the generated drum tracks can be visualized or played instantaneously to facilitate interactive evolution, as explained in the next section.

3.1.2 Drum Pattern Interactive Evolution

It is important to note that unlike in many evolutionary experiments, patterns in the initial generation already sound appropriate. This initial high quality underscores the contribution of the scaffold (i.e. the existing tracks) to the structure of the generated patterns. Thus many appealing patterns already exist in the first generation, demonstrating how quickly appropriate accompani-
ment can be generated as a function of the source tracks. In this way, a major contribution of this research is in showing how rich context can be leveraged by a connectionist system to successfully constrain output to appropriate patterns.

The aim of evolution is thus to elaborate on such patterns. The user can choose to listen to any of the displayed patterns. When listening, the user can listen to the drum track alone or the drum track with its associated song. The visual presentation allows the user to quickly identify unappealing patterns without wasting time listening to them (e.g. wherein the bass is hit over and over again without pause).

Then either the user rates the individual patterns from which NEAT Drummer chooses parents or the user selects a single parent of the next generation. Further rounds of selecting and breeding continue until the user is satisfied. In this way, drum tracks evolve interactively. Because of complexification in NEAT, they can become increasingly elaborate as evolution progresses.

To encourage rapid elaboration over generations, the probability of adding a connection or node in NEAT was 90%. While this high probability would be deleterious in typical NEAT experiments [75, 76], because drum tracks tend to follow song structure, this domain supports adding structure quickly. The mutation power, i.e. the maximum magnitude of weight change, was 0.1 and the probability of mutating the weight of an individual connection was 90%.

Finally, it is also important to note that in principle, the idea of representing musical structure in a connectionist system through scaffolding and conductors could be combined with a different evolutionary algorithm, or even a different training mechanism. Thus while NEAT is a robust algorithm from which to demonstrate the power of scaffolding, the benefit of the scaffolding approach is likely compatible with other connectionist training approaches as well.
3.1.3 Musical Instrument Digital Interface

NEAT Drummer reads its input tracks from Musical Instrument Digital Interface (MIDI) files. Standard MIDI format (SMF) is the most common MIDI filetype. SMF format includes at most 16 channels with any number of tracks, each containing a sequence of instrumental events. Each track contains events that tell a particular instrument when and how loudly to play. According to the specification, most of the instrument sounds can occur in any of the 16 channels with the exception of percussion, for which channel 10 is reserved.

NEAT Drummer can input any combination tracks into the CPPN. That is, given a MIDI song, NEAT Drummer generates a drum pattern as a function of any subset of the preexisting bass, guitar, vocals, etc. The resulting drum patterns are all explicitly functions of the inputs, so if part of a MIDI is input to the ANN, the percussion follows the structure of that part. In this way, NEAT Drummer can generate percussion for MIDI songs based on any subset of the preexisting instrument parts.

3.2 Experimental Design

This chapter includes two sets of experiments. The first set focuses on the capability of the scaffolding approach to generate drum tracks. The second set compares several other approaches with the scaffolding approach, both interactive and supervised, to provide an objective validation of the methodology.

Also, because music appreciation is largely subjective and auditory, the results of NEAT Drummer should be judged in part on that basis. Therefore, MIDI files for every experiment reported in this section are available online at http://eplex.cs.ucf.edu/fsmc/dissertation/. The reader is invited to listen to the recordings and judge the natural quality of the percussion tracks.
3.2.1 Testing the Power of Scaffolding

The first set of experiments aim to determine whether drum tracks generated for particular songs are appropriate and nontrivial. The hope is that they respect the structure and transitions of the song yet do not mimic its instruments superficially. Such sophisticated correspondence can confirm the capacity of functional relationships to generate plausible, human-like drum patterns.

Specifically, the first two experiments in this set investigate what happens when salient instrument tracks are input alone to the CPPN, which generates drum tracks for the folk songs Johnny Cope and Oh! Susanna. A follow-up experiment explores the consequences of inputting both instrument tracks and conductors for the folk song Oh! Dem Golden Slippers. The question is whether the conductors add a dimension of variation that is seamlessly combined with the structure of the original song in the resultant drum track. A complex conductor is then input by itself into a CPPN to isolate its effects and easily discern the functional relationship between the conductor and its outputs.

3.2.2 Comparisons

The second set of experiments are designed to scrutinize the power of scaffolding via input from the original song by attempting to achieve comparable output drum tracks without providing the original song as input. The aim is to illustrate the contribution of such scaffolding by investigating how other approaches fare without it. To control specifically for the contribution of the scaffold, each such attempt is still a variant of NEAT. That way, differences in performance are attributable to representation and scaffolding.

In this spirit, first, ten 30-generation attempts are made to interactively evolve accompanying drums to Johnny Cope with NEAT without the drum tracks from the song input into the network. Instead,
in the first five attempts, the network is recurrent and inputs only a bias. These attempts compare the capabilities of a recurrent network without any scaffolding to those of the scaffolded networks. In the last five attempts, the network is feedforward and provided only position-in-song as input. Typical best results are presented.

Second, three target-based experiments form a more objective comparison. In these target-based runs, the aim is to reproduce a specific drum track that was previously evolved with NEAT Drummer (i.e. with the scaffold provided) as an accompaniment to Johnny Cope. This drum track is set as the target for the target-based experiments, which do not have access to the scaffold.

Target-based runs rely on the same NEAT algorithm as NEAT Drummer; however, the computer performs selection instead of a human user. Selection is performed as in regular NEAT, wherein each individual in the population is assigned a fitness based on the sum-squared error between the target pattern and the attempted output:

\[
f = \sqrt{\frac{\sum_{t=0}^{l} M^2 - \sum_{t=0}^{n} (x_t - y_t)^2}{lM^2}},
\]

where \( M \) is the maximum possible error at any tick \( t \), \( l \) is the number of ticks, \( x_t \) is the target note value at tick \( t \), and \( y_t \) is the output value of the network at tick \( t \). Note that if there are multiple output tracks, this expression is applied to each and the fitness is the average. This fitness function is designed to approach 1.0 the better the output matches the target.

The main question is how hard it will be for NEAT to evolve the very same rhythm that it evolved with the scaffold. Three alternative representations are tested in this way:

- recurrent neural networks with only a bias input,
- feedforward networks with only a position-in-song conductor input, and
- feedforward networks with both a position-in-song conductor and a position-in-measure conductor input.

In these target-based experiments, NEAT is run with typical successful parameter settings for regular non-interactive evolution [75, 76]. In particular, the population size was 100 and probability of adding a connection or node in NEAT was 3% and 5%, respectively. The mutation power, i.e. the maximum magnitude of weight change, was 0.1 and the probability of mutating an individual connection was 80%. The compatibility coefficients for determining to which species individuals belong [75] were $c_1 = 1.0$, $c_2 = 1.0$, and $c_3 = 0.4$. The compatibility threshold $C_t$ was adjusted dynamically in increments of 0.5 to maintain a stable equilibrium of eight species.

If it turns out that any of these variants can evolve the target drum track, it will show that the scaffold is not necessary to provide a context. On the other hand, if none of the representation can evolve the target, it shows the critical contribution of the scaffold.

In summary, experimental results are divided into two parts: First, the power of scaffolding is tested through interactive evolution; second, scaffolding is compared to several variants of NEAT Drummer without scaffolding. The next section details the results of evolving interactively with the scaffold.

3.3 Scaffolding Results: Drum Track Accompaniment

While NEAT Drummer can theoretically input a drum track from a MIDI file and thereby generate variations of the percussion, this section focuses on drum tracks generated from inputting non-percussion instruments, like guitars and bass. Thus the MIDI songs input in this section do not include drums in their original form.
Results in this section are reported through figures that are designed to demonstrate the relationship between the CPPN inputs and outputs as the song progresses over time. In the figures that follow, the inputs are arranged in rows at the bottom of each depiction and the outputs are the rows above. Time moves from left to right and each discrete column represents a tick of the clock. No instrument can play at a rate faster than the clock ticks. There are four ticks per beat in all songs tested. A slightly thicker dividing line between columns denotes a measure break. While all drum tracks include bass, snare, and hi-hat outputs, the number and types of drum outputs is unlimited in principle as long as the right sounds are available.

Recall that inputs are spikes in the figures, their decays are depicted as decreasing darkness. In contrast, outputs represent volumes, wherein darker shading indicates higher volume. The main difference between inputs and outputs is that a single note in the input may straddle several columns during its decay. Outputs on the other hand are played as separate notes for every solid column. For an output drum to last for more than a single tick before the next drum attack, it must be followed by white (empty) columns.

3.3.1 Inputting Instrument Tracks Alone

Figure 3.5 shows individuals from generations one and 11 generated for the folk song Johnny Cope. The relationship between the bass, hi-hat, and snare and the three input tracks is complex because each drum is related to all three inputs. Note however that the instrumental patterns in measures three and four are highly related though not identical. Slight differences exist between the piano pattern in measure three and measure four; this difference is reflected in the snare in both generations one and 11, which both slightly differ between the early parts of measures three and four. Thus, the drum pattern’s subtle variation is correlated to the music because of their coupling, which evokes a subjective sense of appropriate style.
Figure 3.5: **Johnny Cope Results.** Results are depicted from two different generations at two different parts of the song. The inputs from the original song, which are always the same, are shown at bottom. Note the relationship between the inputs and the outputs, and between the first generation and the eleventh, which elaborates on the former. The motif in measures three and four is typical of the first part of the song until measure 23, when it switches to a different motif in both generations. Thus, the figure gives a sense of the two predominant drum riffs exhibited in both generations. The main conclusion is that the output is a function of the input that inherits its underlying style and character. (These tracks are available at http://eplex.cs.ucf.edu/fsmc/dissertation/)

At measure 23, the song changes sharply by eliminating the piano part. Consequently, the CPPN outputs also diverge from their previous consistent motifs. This strongly coupled divergence that is carried both in the tune and in the drums creates a sense of purposeful coordination, again yielding a natural, sophisticated feel. In this way, the functional relationship represented by the CPPN causes the drums to follow the contours of the music seamlessly.

Generation 11, which evolved 12 additional connections and six additional nodes, reacts particularly strongly to the elimination of the piano by significantly altering its overall pattern. In genera-
Figure 3.6: **Oh! Susanna Outputs**. This pattern from measures three through six of Oh! Susanna is from generation 25 of evolution. The network evolved 15 hidden nodes and 41 connections. Near the end of measure four is a particularly improvisational riff in the snare that transitions to measure five. This riff is caused by variation in the piano and other inputs at the same time. As with Johnny Cope, the drum pattern sounds natural and styled correctly for this upbeat song. (This track is available at http://eplex.cs.ucf.edu/fsmc/dissertation/)

Results from generation 25 of *Oh! Susanna* are shown in figure 3.6. NEAT Drummer produces similarly natural and style-appropriate rhythms for this song as well, suggesting its generality. Because style is inherent in the original song’s tracks, it transfers seamlessly to the drum track without any need for explicit stylistic rules. The result is an entertaining sound that could be attached to the original instrumental tracks without raising suspicion.

It is interesting to listen to the songs with their generated drum tracks, which makes it possible to judge their subjective quality (a critical aspect of musical appreciation). In the authors’ experience (which the reader can also judge), the generated tracks sound natural and lack the usual “mechani-
cal” quality of computer-generated music. Rather than repeating stock patterns, core motifs subtly vary and are interspersed with occasional unique flourishes. The personality of these variations is a byproduct of the personality that is implicit in the song itself, simply functionally transformed into a different local motif. This result further demonstrates that it is possible to inherit the natural character of one pattern by deriving another from it. Of course, the evolved song also in part reflects the tastes of the human user.

### 3.3.2 Inputting Instrument Tracks and Conductors

Figure 3.7 highlights the effect of a conductor input on drum tracks produced for the song Oh! Dem Golden Slippers, which has a very similar beginning and end; all the measures in these parts are similar. Thus the question is whether a conductor can introduce a sense of progression into the drums even though the song itself undergoes little discernible progression between the start and finish.

Figure 3.7a shows example drum output for this song without any additional conductor. Thus, with only the song’s tracks as inputs, the resultant drum pattern is also highly repetitive; the pattern in measures two and three is largely preserved much later in measures 38 and 39 (figure 3.7a). Yet when a position-in-song conductor (figure 3.4a) is added as an input, the difference in drum patterns between measures two and three and measures 38 and 39 is dramatic, showing the powerful effect of the simple conductor (figure 3.7b). Nevertheless, even though the drum pattern exhibits a sharply different motif at these two similar parts of the song, it sounds appropriate and sophisticated in both parts because it is a function of both the conductor and the instrument tracks. Thus it is a seamless variation on both influences simultaneously.

It is also possible to combine multiple conductors to affect the structure of the output in more than one way. Figure 3.7c shows the impact of inputting both the time in the song (figure 3.4a)
Figure 3.7: **Oh! Dem Golden Slippers with and without Conductors.** Output drum patterns are shown for the song in one case when no conductor is input (a) and in the other where a position-in-song conductor is input (b). The difference in resultant drum patterns shows that the conductor imposes a temporal progression on the drum track that does not derive from the structure of the song itself, demonstrating the power of conductors to subtly shape the structure of music. Finally, when conductors indicating both position-in-song and position-in-measure are input simultaneously (c), progression is enhanced both throughout the song and within each measure.
Figure 3.8: **Complex Conductor.** The conductor, which follows the pattern quarter quarter half, is shown at bottom. Two three-part drum tracks that are functions of this conductor are shown above it. While both drum tracks are different, they are also both constrained by the underlying motif of the conductor.

and the time in the measure (figure 3.4b) together. The result is that not only do the later drum patterns differ from the earlier ones, but the interior of each measure varies in part independently of the instrumental scaffold. This effect is subtle because there are five instrumental tracks already influencing the pattern in each measure. However, closely comparing the drum measure patterns in 3.7c to 3.7a and 3.7b does reveal a discernible difference.

Finally, figure 3.8 isolates the effect of a single complex conductor. The aim is to show explicitly how the output of the CPPN is influenced by the incoming conductor, which expresses the same quarter quarter half pattern as in figure 3.3 (from the approach). Thus, the outputs of two CPPNs that both take the same conductor are displayed for comparison. The main result is that the patterns of the two three-part tracks are both closely tied to the pattern of the conductor, wherein two short
events are always followed by a long one. Yet, within that framework, the patterns nevertheless differ significantly, illustrating the idea that a conductor is an implicit guide above which the pattern is realized, even if there is no explicit song at all.

The next section exhibits the evolved CPPNs that produce the drum tracks in this section.

### 3.3.3 Evolved CPPNs

Figure 3.9 shows the CPPNs that were evolved for each of the evolved drum tracks in Sections 3.3.1 and 3.3.2. These networks range in complexity from 1 to 15 hidden nodes. Interestingly, the subjective quality of the drum patterns does not seem to correlate to the complexity of the network. This perception makes sense because the functional relationship to the original instrument tracks guarantees a tight coordination between drums and instruments. Thus creating a plausible coordination does not require significant complexity. Furthermore, if the underlying instrument tracks themselves embody complex motifs and progressions, then the drums inherit the same complexity even if the CPPN that relates them is not itself complex.

What CPPN complexity affords, rather, is a more complex relationship that is realized through more elaborate covariation. This subjective effect is subtle yet perceptible, suggesting that more sophisticated compositions may suggest to the human ear the complexity of the function relating their parts.

Yet the most important conclusion is that complexity is not essential to the CPPN that relates one part to another because the complexity need only exist originally in the preexisting parts. To a large extent, that original complexity is transferred through the CPPN to any affiliated drum pattern.

The next section presents the results from the comparative experiments.
Figure 3.9: CPPN Drum Track Generators. The evolved CPPNs that produce every drum track shown in Sections 3.2.1 and 3.2.2 are depicted. While the complexities vary, the quality of the output is similar because each produces a function of a preexisting song, thereby inheriting its qualities. Activation functions are denoted by $S$ for sigmoid, $M$ for multiplication, $G$ for Gaussian, and $L$ for linear.
3.4 Comparison Results

While the results in Section 3.3 establish the quality of the tracks produced through scaffolding and the power of conductors to shape the output pattern, the question remains what is lost if the scaffold is not provided, as in prior automated music generation techniques. Can similar drum patterns be produced without a scaffold? This section validates the role of the scaffold by answering this question.

Typical results reported in this section can be heard at http://eplex.cs.ucf.edu/fsmc/dissertation/.

3.4.1 Interactive Evolution without Scaffolding

As described in Section 3.2.2, in the first set of comparisons, recurrent ANNs with only a bias input and feedforward ANNs with position-in-song as input were evolved interactively with no other inputs to accompany Johnny Cope. Five 30-generation runs of both configurations were completed.

Figure 3.10 shows typical best results from these runs. The best results from each set reveal a distinct difference between feedforward functions of position-in-song and evolved recurrent networks. The feedforward patterns never develop beyond monotonous unbroken gradients that gradually vary from loud to soft, and sometimes back again (figure 3.10a). These gradients often span the length of the song, or a large extent of it, and do not respect the measure structure. Thus, overall, position-in-song alone is not enough to allow interactive evolution to compete with evolving networks with a better scaffold. This result makes sense because the only input is the position in the song, so the only way to develop a significant number of changes is to add many hidden nodes, which would take far longer than 30 generations. Also, because CPPNs have no
knowledge of when measures begin or end, the changes that do occur are difficult to evolve to align with the measure structure.

In contrast, patterns interactively evolved with recurrent networks do display more complexity and more frequent changes over time (figure 3.10b). Because feedback can lead to complex oscillations without the need for many hidden nodes, recurrent networks are better suited to producing complex variation early in evolution. However, the drum patterns are difficult to evolve to align with the contours of the music itself because the recurrent network is unaware of the music without the scaffold. Furthermore, these networks suffer from the same problem with measure structure as the feedforward networks: Even though some motifs repeat, they repeat at haphazard times relative to each measure, producing a disorganized aesthetic. For example, the bass drum in figure 3.10 b is hit several times at the start of the first measure, but by the fourth measure, this motif has moved to the middle of the measure.

The overall result is that while the recurrent networks produce more complexity, the patterns evolved by both networks are not synchronized with Johnny Cope and therefore sound disjointed, highlighting the critical role of the scaffold in tethering the accompaniment to the song itself.

### 3.4.2 Targeted Evolution without Scaffolding

In this experiment, the same two types of networks as in the previous section, i.e. one recurrent and one feedforward with position-in-song input, were evolved to match a target (figure 3.11c), which is a rhythm for Johnny Cope output by NEAT Drummer with the scaffold in the 11th generation. Clearly, the scaffold provides an advantage, but the question addressed by this experiment is how hard it is without the scaffold to approximate the same output even when the precise target rhythm is known a priori. Because it is also known that the target rhythm took exactly 11 generations to evolve when the scaffold was provided, the number of generations it takes these variant networks
to produce the same output can be compared. Each variant attempted to match the target in 20 separate runs.

Figure 3.11 shows typical best results from these runs after 1; 600 NEAT generations with a population size of 100. It turns out that the feedforward network suffered similar problems breaking away from simple gradients as in interactive evolution. While such a network theoretically could approximate the target pattern, it always became trapped on a local optimum because it is easy to reach a fairly high fitness simply by approximating the general loudness of drums over large contiguous periods of time. In other words, instead of attempting to discover every individual beat, it discovers their average energy and paints that energy level across large swaths of time. Thus this type of network is demonstrably ill-suited to producing such temporal patterns, either through interactive evolution or target-based evolution.

However, interestingly, unlike in the 30-generation interactive experiment, after 1600 generations the recurrent network typically produces a repeating pattern that does synchronize with the measure structure. Thus one conclusion is that recurrent networks can learn musical structure given sufficient time. However, unlike the target pattern, which displays distinct variations (e.g. in the latter half of figure 3.11c), the pattern produced by the best recurrent networks tend to repeat the same measure pattern throughout the song after the first generation (figure 3.11b). Also, this repeating motif is only vaguely reminiscent of the target, which is likely because the recurrent network has trouble producing the subtle repetition with variation that the target inherited from its scaffold when it was evolved.

Because the feedforward results were disappointing, a third set of 20 runs was attempted with a feedforward network that receives both position-in-song and position-in-measure conductors as input. The idea is to relieve the network of the need to discover the measure structure of music on its own, exploiting the power of conductors. In fact, as figure 3.12 shows, providing position-in-
measure typically dramatically improves the complexity of the output and allows it to break out of the local optima that trap such networks without position-in-measure. Indeed, the feedforward output resembles the output of the recurrent network and respects measure structure, demonstrating the contribution of the additional conductor. Yet because even such conductors do not contain the same song-specific information as the scaffold, like the recurrent network, the output pattern is only marginally reminiscent of the actual target pattern, and is also more repetitive.

Thus, after 1,600 generations, none of the variant networks are able to successfully match a target pattern that was discovered in only 11 generations. Figure 3.13 summarizes this result by depicting fitness over time in the three variants, each averaged over 20 runs. Whereas a fitness of 1:0 denotes a perfect match, none of the variants reach a fitness of even 0:7. Interestingly, despite the apparent aesthetic inferiority of the feedforward network with only position-in-song, on average its fitness approaches that of the other two variants, demonstrating why local optima characterized by smooth volume gradients attract it.

Although their final fitness levels are not far apart, the differences between some of the variants are significant. In particular, recurrent networks produce significantly higher fitness than position-in-song alone throughout the run ($p < 0.05$), and feedforward with the position-in-measure input outperforms feedforward without it ($p < 0.05$). However, interestingly, by the end of each run, recurrent networks on average are not significantly better than feedforward with position-in-measure, showing that feedforward networks can match the performance of recurrent networks on this task when provided information on the structure of music. However, most significantly, none of the variants can produce a pattern close to the target within 1,600 generations.

The main conclusion from both interactive and target-based comparisons is thus that the scaffold provides critical infrastructure. In effect, it constrains the search to patterns that relate to the original song. If accompaniment is to be evolved for an existing song, inputs from that song
should be provided as a scaffold. Without such context, the accompaniment becomes decoupled from the song regardless of the representation.

A further result is that conductors make it easier to discover patterns that respect musical structure. While the recurrent network does eventually discover a measure-synchronized motif in the target-based runs, it takes hundreds of generations to achieve such synchrony. On the other hand, when time-in-measure is provided as a conductor, measure structure is respected from the start. Overall, this set of experiments confirms the contribution of the scaffold and suggests that it should be a standard facet of any network-based attempt to generate additional musical voices for an existing piece.

3.5 Implications

NEAT Drummer exploits the fact that the different parts of a musical piece are functionally related, which is what allows the output to sound natural and song-appropriate. The implications of this functional relationship are significant because it means that many parts of a song are less complex than they seem to be when put in the context of the rest of the instruments.

Formally, any pattern over time can be described as a function of time, $f(t)$. However, a good pattern for a particular drum alone may be highly complex with respect to $t$, making its discovery prohibitive. Yet given another part $g(t)$ that varies over time, because the parts of a song are related to each other, it may be easier to discover $h(f(t))$ than $g(t)$ even though they produce the same output. In effect, $f(t)$ provides a scaffold upon which other parts can be attached.

In this way, the drums are a function of the melodic and the harmonic rhythm patterns, and, interestingly, those patterns are also an inverse function of the drums. The important implication is that as long as something is human-generated, the rest will naturally inherit its intangible natural
quality on their own. Furthermore, they will automatically inherit the style intrinsic in the scaffold, removing the need for style-specific considerations.

Finally, interactive evolution allows the user to refine and elaborate the initial track in an unlimited variety of ways. The next chapter explores the functional relationships between melodic and harmonic rhythms and pitches in different music by generating pitched additional voices for existing pieces.

3.6 Conclusion

This chapter introduced NEAT Drummer, a program that can generate novel drum tracks for songs sequenced in MIDI. The main idea is to input instrumental tracks directly into a kind of neural network called a compositional pattern producing network (CPPN), which produces a pattern that is a function of its input. In this way, the drum sequences output by the CPPN are related to the song inputs, which thereby supply a scaffold for the drums. Human users can then evolve and thereby elaborate the track to satisfy their specific tastes. Drum tracks generated for two popular folk songs exhibit a natural style appropriate to each song that lacks the familiar computerized feel of computer-generated music. This work in rhythm generation suggests the general power of functional relationships in music, which leads to the Functional Scaffolding for Musical Composition theory discussed in the next chapter.
Figure 3.10: Typical Best Results from Interactive Evolution Without Scaffolding. Best results from both types of representations tested are depicted. The feedforward network only inputs position-in-song and produces unremarkable gradient patterns that do not follow the structure of music nor the Johnny Cope song. The recurrent network produces repeating motifs, but they are not synchronized with the measure and they do not vary with the song. In this way, removing the scaffold removes a major advantage of NEAT Drummer.
Figure 3.11: Typical Best Results from Target-based Evolution Without Scaffolding. Feedforward (a) and recurrent (b) results are depicted and the target pattern is shown in (c). The aim was to match the target. As this figure shows, neither variant successfully matched the target (which was evolved in NEAT Drummer in 11 generations) after 1,600 generations, although the recurrent variant evolved more complex patterns. This result confirms again the importance of the scaffold.
Figure 3.12: **Typical Result from Target-based Evolution with Position-in-song and Position-in-measure Inputs.** The improvement in pattern complexity, and the adherence to measure structure, are apparent in comparison to figure 3.11a. By providing position-in-measure as input, evolution can easily produce patterns that follow the timing of measures, demonstrating the power of conductor inputs. However, without the scaffolding inputs from Johnny Cope, the drum pattern still does not match the target despite its regular structure.

Figure 3.13: **Fitness Over Time of the Three Variants in Target-based Evolution.** The increase in fitness over 1,600 generations of evolving the three variant representations is shown, averaged over 20 runs for each. A perfect fitness of 1.0 would mean the target is matched perfectly. None of the variants exceed 0.7 fitness because the search is too unconstrained without the scaffold.
CHAPTER 4: GENERATING MONOPHONIC ADDITIONAL VOICES FOR EXISTING COMPOSITIONS

This chapter extends the idea of exploiting functional relationships in musical pieces that was introduced by NEAT Drummer to a more comprehensive approach called *functional scaffolding for musical composition* (FSMC). The FSMC approach is implemented in a program called MaestroGenesis (available at http://maestrogenesis.org) that generates complete melodies and harmonies from existing compositions. These compositions form the scaffold from which computer-generated voices are built. However, unlike in NEAT Drummer, these scaffolds include rhythmic information *and* pitch information, thereby providing the foundation for harmonization.

Pitch patterns are an important aspect of most musical styles. Western musicians often constrain pieces to a particular *key* (i.e. a subset of all possible pitches) to ensure that the notes sound plausible together. However, even within a key, the emotions that a piece conveys can differ dramatically; songs in major *modes* may sound triumphant and happy while their counterparts in relative minor modes can sound eerie and sad even though they are combinations of the same sets of pitches. Furthermore, context is essential for determining the plausibility of any two simultaneously sounded notes, or *interval*. Dissonant intervals that precede resolving intervals sound satisfying, while unresolved intervals sound unnatural. Recalling that NEAT Drummer extracts rhythms from a spike-decay model introduced in 3.3 and evolves these rhythms through a CPPN representation, how can this approach be extended to exploit important regularities in pitch patterns of existing musical pieces?

To understand the idea behind FSMC consider again that if different instrumental parts in the same composition were not related to each other at all, they would sound inappropriate together. This principle should hold for pitched instruments just as it does for percussion. Recall that there
is some relationship between different parts in the same piece. In effect, 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. If that function is simple then the relationship between the parts is more easily discernible than if the function is complex. Yet in any case, the important point is that there is some function that relates these parts to each other. The idea in FSMC and thus MaestroGenesis is to exploit this fact similarly to NEAT Drummer by evolving the function that relates one part to another, this time with pitched instruments. That way, instead of searching for a sequence of notes, MaestroGenesis can help users search for a transforming function that bootstraps off the existing parts (i.e. the scaffold) to generate the complete computer-generated voice. In effect, FSMC is the hidden function that relates different parts of a composition to each other.

FSMC thus represents computer-generated voices as a function that transforms pitches and rhythms from the scaffold into a temporal pattern interpreted as the generated voice. In particular, this transforming function is encoded in FSMC by an expanded CPPN representation, as explained in the next section. The CPPN representation is extended from that in Chapter 3, which only encoded drum patterns. Outputs from CPPNs are interpreted as the additional generated voices that thereby follow contours of the original song. Users then interactively explore the space of such functions for personalized generated voices through IEC. The research in this chapter was originally published in [35] and [34].

4.1 Functional Relationship Representation

FSMC divides each musical part into a pitch pattern and a rhythm pattern, both of which are represented by separate CPPNs (figure 4.1). While CPPNs themselves are not essential to FSMC, they serve as convenient representations for exploiting the functional relationships between parts of
a piece. The particular idea of separating pitch and rhythm follows a tradition in other approaches to music generation [21, 41]. The rhythm network, which extends the CPPN representation in NEAT Drummer, is shown in figure 4.1a. Unlike in NEAT Drummer, it must also contend with the fact that pitched instruments can be voiced over multiple time steps. It has a set of scaffold inputs from the original composition (i.e. before a generated voice is added) and two output nodes for each instrument in the generated voice: \textit{OnOff} and \textit{NewNote}. OnOff decides volume and whether or not the note will play. If the OnOff output returns a value below a given threshold, the generated voice rests at that tick. If OnOff indicates that a note is to be played, NewNote decides whether the note will be re-struck or sustained. In partnership with the rhythm CPPN, the pitch CPPN (which was not proposed in NEAT Drummer) in figure 4.1b sees the pitches of instruments in the scaffold and decides the pitch of the generated voice with a single output. Viable pitches are discretized into bins that correspond to the given key and the network thereby plays the pitch closest to its output. The CPPNs in figure 4.1 act just like ANNs with weighted connections and hidden neurons that transform the scaffold input at the current timestep into rhythm and pitch generated voices.

The CPPN representations in figure 4.1 thus in effect implement the idea of functional scaffolding for a pitched generated voice. The CPPN is itself just a formalism for specifying a function that can be artificially evolved. The inputs to the CPPNs are the pattern of notes and durations within the scaffold and the outputs are the additional generated voice. In this way, the CPPN is literally a function of the scaffold that transforms it into a functionally-related computer-generated pattern.

The hidden nodes in the CPPNs depicted in figure 4.1 are added by mutations that occur over the evolutionary process. They in effect increase the complexity of the transforming function by adding intervening nonlinearities. For example, the Gaussian function (depicted as a “G”) introduces symmetry (i.e. such as the same sequence of notes ascending and then descending) and the sigmoid (depicted as “S”) is nonlinear yet asymmetric. As in a neural network, the connections are weights (i.e. coefficients) that are multiplied by their inputs. By accumulating such transfor-
Figure 4.1: **How CPPNs Compute a Function of the Input Scaffold.** The rhythm CPPN in (a) and pitch CPPN in (b) together form the generated voices of FSMC. The inputs to the CPPNs are the scaffold rhythms and pitches for the respective networks and the outputs indicate the rhythms and pitches of the generated voices. Each rhythm network has two outputs for each instrument: OnOff and NewNote. The OnOff node controls volume and whether or not a note is played. The NewNote node indicates whether a note is re-voiced or sustained at the current tick. If OnOff indicates a rest, the NewNote node is ignored. The pitch CPPN output decides what pitch the generated voice should play at that particular tick. The internal topologies of these networks, which encode the functions they perform, change over evolution. The functions within each node depict that a CPPN can include more than one activation function, such as Gaussian and sigmoid functions. Two monophonic generated voice outputs are depicted, but the number of instruments a CPPN can output is unlimited. The number of input instruments also can vary.

In evolutions, the relationship between scaffold and generated voice can become more complex.

To help illustrate intuitively how CPPNs work to encode functional transformations in FSMC, figure 4.2 shows how pitch outputs are calculated at each tick and how CPPN mutations can affect the generated output. The sequence in figure 4.2 is a simple example of both how CPPNs calculate their outputs and also how mutations to the CPPN in figure 4.2a alter the output it produces for the same scaffold (shown in figure 4.2b). While this example focuses on the pitch CPPN, the rhythm CPPN computes its transformations in an analogous manner. Each of the four identical pitch CPPNs in figure 4.2a and the four identical CPPNs in figure 4.2b represent a calculation made at a particular tick from both a bias (which is just a constant input) and scaffold input. To calculate
Figure 4.2: **Pitch CPPNs over Two Generations** The pitch CPPNs in (a) and (b) illustrate how scaffolds are transformed to musical outputs. Each of the four identical CPPNs in (a) and the four identical CPPNs in (b) represent a calculation made at the four quarter-note-length ticks in this one measure scaffold. The CPPN in (a) is from the first generation and has yet to evolve hidden nodes, while the CPPN in (b) from the second generation has evolved an hidden node between the bias and output and mutated the existing connection weights.
the output value for the simple CPPN in figure 4.2a (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 $\text{sigmoid}(\text{ActivationSum}) = \frac{1}{1 + e^{-2 \cdot \text{ActivationSum}}}$ is a real number between $[0, 1]$ that is then mapped to one of fifteen notes in a two octaves of a diatonic key set by the user. In figure 4.2, the first note output 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 $\frac{9}{15}$, or 9 diatonic steps above the starting pitch of C below Middle C. While the output of MaestroGenesis is constrained to standard diatonic keys, scaffold values can be input as 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 with its nodes that determine what it outputs for a particular scaffold input.

Unlike the network in figure 4.2a, the CPPN in figure 4.2b 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 input to its sigmoid activation function such that the output (activation level) for the hidden node is $\text{sigmoid}(1.1 \cdot 1.0)$. 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, $\text{normalize(midi)} \cdot -0.62$. The final output depends on the current tick, but is represented by the function, $\text{output} = \text{sigmoid}(\text{sigmoid}(1.1 \cdot 1.0) \cdot -0.19 + (\text{normalize(midi)} \cdot -0.98))$. While the generated melody in 4.2a is transposed a diatonic third from the scaffold pitches, the additional hidden function and corresponding weight mutations in figure 4.2b in effect transpose and mirror invert the melody generated in figure 4.2a down a diatonic second.
Figure 4.3: CPPN Input Representation. The spike-decay representation for rhythms is shown in (a) and the pitch representation is shown in (b). Note that spike-decay model is identical to that seen in Chapter 3. Both such inputs are depicted in two ways: The first is a continuous-time graph that shows decaying spikes for rhythm and the pitch level for pitch. The second is a standard musical notation representation of the scaffold. In this way, this figure gives a sense of exactly what the CPPN “hears” (for each instrument in the scaffold) as it generates additional voices.

In total, the example in figure 4.2 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 4.3a shows an example of a temporal pattern that is input into the rhythm CPPN. This scaffold is four measures of a repeating quarter-quarter-half note motif. To impart a sense of time within a note, when a note begins, an attack spike is sent to the network for that particular instant in time. This spike decays linearly over time for the duration of the scaffold note. This spike-decay representation of time ensures that the position within the particular note is known to the rhythm network at any given time, thereby providing rhythmic context from the scaffold to the generated voice. Thus it can output patterns based on the rhythmic information in the scaffold. Simultaneously, the pitch from the scaffold at each discrete instant in time is sent modulo 12 to the pitch CPPN (figure 4.3b), whose output is converted to one of eight pitches in the specified key.

Figure 4.4 illustrates through an example how MaestroGenesis interprets the pitch and rhythmic 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 guitar, clavinet I, and clavinet II, is input to both the rhythm and pitch CPPNs to generate the output additional voices for Scarborough Fair.

In effect, the CPPN and its inputs provide the functional scaffolding in FSMC. The next section explains how such preferences are conveyed through the evolutionary process.

4.2 Choosing Scaffolds and Evolving Harmonies

Users select additional generated voices through IEC. After choosing initial network configurations, i.e. inputs for the rhythm and pitch networks and instrument outputs, FSMC presents users with a population of possible generated voices to the scaffold (figure 4.5). These candidates are then rated by the user. The user rates any number of pieces in the current population, all of which influence the next generation of generated voices. If mutation rates are high, the character between
Figure 4.4: **Representing the Scaffold.** This generated voice for Scarborough Fair (top) is generated from the four instruments in the scaffold (bottom). Each of these instruments is input to both the rhythmic CPPN (middle left) and the pitch CPPN (middle right) using the input representation shown in figure 4.3. This example can be heard at [http://eplex.cs.ucf.edu/fsmc/dissertation/](http://eplex.cs.ucf.edu/fsmc/dissertation/).
Figure 4.5: MaestroGenesis Graphical User Interface. Generated voices in FSMC are presented both visually and sonically. Unlike image evolution, in which users can quickly evaluate the population [79], listening to MIDIs takes time. The visual representations help users decide which MIDIs are worth this extra time, thus speeding up evolution. There are ten individuals in a population, which are all displayed at once. The user rates individuals by giving them a thumbs-up or thumbs-down.

generations varies greatly. Otherwise, with low mutation rates, the user evolves pieces with similar character. Mutations in CPPNs generally modify connection weights or add or remove connections and nodes, following the NEAT method [75].

While IEC has previously been applied to pitched music generation [57, 56, 2, 6], the hope in FSMC is that the CPPN representation of the functional relationship between scaffold and generated voice will allow a holistic evolution of song-wide patterns. Instead of manipulating single notes or features of a composition, FSMC evolves entire functional relationships, thereby ensuring that the search space at least only considers generated voices with some relationship to the scaffold.

Selecting the scaffold is itself an important task. It requires choosing to which instrument tracks
the rhythm and pitch networks should listen. While beginners can easily choose appropriate inputs to create appealing generated voices, refined choices can significantly influence the piece. For example, whether or not the rhythm network listens to a fast-changing instrument can impact the complexity of the corresponding output generated voice. In fact, chosen tracks do not have to be the same for each network (e.g. the rhythm network can have a piano and guitar input while the pitch network only has a bass guitar input).

Because the parts of the scaffold themselves are human-composed and thereby sound appealing, generated voices built from any combination of such tracks ends up following the contours of the original song. However, depending on the specific inputs selected and the internal network structure, the relationship between selected inputs and the generated voices may be of varying complexity. NEAT [75] (the underlying evolutionary algorithm that evolves CPPNs), which occasionally adds new structure to CPPNs (and can also remove it in the implementation in this chapter) allows such complexity to increase or decrease following the preferences of the IEC user.

4.3 Experiments

FSMC 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. 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 are asked to rate the quality of the voices at the beginning, middle, and end of evolution, thereby revealing whether FSMC helps uncover a plausible area of musical space through IEC.

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 computer composed? To answer this question, the next experiment tests whether listeners can distinguish between two partially computer-composed and fully human-composed pieces. It also explores the internal structure of each generated voice to identify the properties of CPPNs that compose more appealing generated voices.

4.3.1 Investigating the Evolution of Generated Voices

To begin to study the capabilities of FSMC, it is helpful to analyze in detail a representative evolutionary progression of generated voices. Such an analysis, coupled with a user study of perceived quality over generations, helps to illuminate how generated voices are evolved and the contribution of interactive evolution to the results. This understanding will provide context for later experiments that concentrate more on the final output of user sessions.

In this experiment, the focus is on the evolution of the generated voice. Therefore, the scaffold, i.e. music for which the generated voice will be evolved, is chosen to meet an established level of quality. That way, it is possible to determine whether the generated voice can maintain and complement the original quality in the scaffold. For this purpose, the well-known folk song Bad Girl’s Lament is chosen.

The interactive evolutionary process for the example piece was guided by MaestroGenesis team members. No musical knowledge was applied 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 generated 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. While user input is an important aspect of
this process, no session lasted more than 12 generations (i.e. no more than 12 preference decisions were ever made), highlighting the overriding importance of the FSMC relationship to constraining generated voices to a reasonable set of candidates. Thus, interestingly, in contrast to data-intensive approaches, the only knowledge needed to generate voices through this approach is imparted in ten to 15 clicks of IEC.

To explore the space created by FSMC, an evolutionary progression of an instrumental generated voices for Bad Girl’s Lament between generations 1 and 12 is studied by highlighting important milestones at generations 1, 6, and 12. This 12-generation progression took about thirty minutes in total for the user to complete; most of the time was spent listening to candidate generated voices. Inputs to the rhythm and pitch CPPNs are the piano and harpsichord tracks from the scaffold (i.e. from the original Bad Girl’s Lament MIDI).

While the particular evolutionary progression of Bad Girl’s Lament chosen for analysis is anecdotal, most results with other pieces exhibit similar features and dynamics. Thus the hope in this section is to provide deeper insight into what exactly FSMC does when combined with IEC by focusing on the details of a particular progression. Later experiments will identify more general elements of FSMC, but without the level of individual detail possible here.

Generated voices are evolved through NEAT with a CPPN mutation rate and crossover rate of 0.3. The NewNote threshold is also 0.3. Furthermore, when the OnOff output in the rhythm network (which also indicates volume) falls below 0.3, no note is played. Population size was 10 per generation. The next generation was generated through mutation and recombination of solely the choices of the user.

To appreciate the results for Bad Girl’s Lament, it is important to experience the generated tracks. All the selections discussed in this chapter can be heard at http://eplex.cs.ucf.edu/fsmc/dissertation/.
Figure 4.6 shows measures 17, 18, and 19 of the generated voices for Bad Girl’s Lament in generations 1, 6, and 12. The pitches in measures 17 and 18 of the first generation differ from those created for generations 6 and 12. Pitches in generation 1 ascend across notes A and B in measure 17 followed by C# and B in measure 18. However, in generations 6 and 12, the pattern more closely follows the harpsichord input from the scaffold with notes B and D occurring at beats one and two and a half in measure 17 and 18, demonstrating the influence of the functional relationship to the harpsichord on the evolved progressions. However, in the third measure, generation 12 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 the nineteenth measure of generations 1 and 6.

While the three depicted generations in Bad Girl’s Lament exhibit some similar characteristics, they progressively change over evolutionary time. For example, while generations 6 and 12 are rhythmically similar, generation 1 sounds significantly shorter notes. The pitch evolution progresses similarly to rhythm. From generation 1 to 6 many pitches change, but generations 6 and 12 differ in pitch by only a few choice notes.

To understand the effect of evolution on subjective appreciation, a total of 60 listeners, all of whom are 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 is which, they listened to (1) an intentionally poor-quality control with an inappropriate generated voice (which helps to establish that participants indeed generally agree on something subjective), (2) the original Bad Girl's Lament without an additional generated voice, (3) the song with a generated voice selected from the first generation of IEC, (4) the song with a generated voice selected from the sixth generation of IEC, and (5) the final selected song with a generated voice from generation 12. For each of these variants, the listener was asked:
Figure 4.6: **Evolutionary Sequence of Generated Voices for Bad Girl’s Lament.** Three measures of the evolved steel guitar voice from generations 1, 6, and 12 of Bad Girl's Lament is shown at top, followed by the pitch and rhythm inputs to the CPPN from the scaffold. In this experiment, the tick length or smallest rhythmic unit is a sixteenth note. Each generated voice can be heard at [http://eplex.cs.ucf.edu/fsmc/dissertation/](http://eplex.cs.ucf.edu/fsmc/dissertation/). The type of instrument played in the scaffold is noted at left (e.g. a harpsichord is one track in the rhythm scaffold). The increase in harmonic and rhythmic sophistication between generations 1 and 12 is apparent in the progression at top. In addition, the relationship (e.g. in note transition points) between the scaffold and generated voice can also be observed.

Rate MIDI $i$ on a scale of one to ten. (1 is the worst and 10 is the best),

where $i$ refers to one of the five variants, which are available for listening online at [http://eplex.cs.ucf.edu/fsmc/dissertation/](http://eplex.cs.ucf.edu/fsmc/dissertation/).

By establishing the perceived quality of a respected composition, it becomes possible to estimate
Table 4.1: **Perceived Quality by Survey Participants.** This table shows the average ratings and the mean and standard deviation for the control and four Bad Girl’s Lament (BGL) MIDIs. The MIDI names are on the left while the average ratings are on the right.

<table>
<thead>
<tr>
<th>MIDI Name</th>
<th>Mean</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poor Control</td>
<td>4.35</td>
<td>1.93</td>
</tr>
<tr>
<td>BGL without an Add. Voice</td>
<td>7.30</td>
<td>1.85</td>
</tr>
<tr>
<td>BGL, Generation 1</td>
<td>5.15</td>
<td>2.20</td>
</tr>
<tr>
<td>BGL, Generation 6</td>
<td>6.07</td>
<td>1.96</td>
</tr>
<tr>
<td>BGL, Generation 12</td>
<td>6.83</td>
<td>1.98</td>
</tr>
</tbody>
</table>

how well evolution can maintain that professional standard even though FSMC with IEC incorporates *no prior musical knowledge or expertise*. In fact, the important deeper aim in this experiment is to suggest that through FSMC, evolutionary-assisted music generation can reach a high level of quality even with almost no musical theory whatsoever.

The results from the 60-person listener study, which focused on the same IEC-evolved generated voices for Bad Girl’s Lament from the previous section, are shown in table 4.1. The first entry is the control example, which sounds intentionally artificial. As expected, it is rated significantly worse than every other example in the survey (at least \( p < 0.001, t(67) \geq 2.50 \) for all pair-wise comparisons with a Student’s t-test). This result establishes that listeners likely understood the questions in the survey.

Importantly, generation 6 is judged significantly higher quality than generation 1 (\( p < 0.05, t(67) = 3.50 \)) and generation 12 is judged significantly better than generation 6 (\( p < 0.05, t(67) = 2.49 \)). To verify the Student’s t-test finding, differences between the first and sixth generations and sixth and twelfth are tested with a Friedman test and then a Wilcoxon signed rank test; differences with these tests are also found to be significant with \( p < .001, Z = 3.40 \) and \( p < .002, Z = 2.86 \) respectively. Thus the participants felt that the progression from generation 1
to 12 indeed exhibits continual improvement in quality.

Furthermore, although the original MIDI without an additional voice is judged significantly better than generation 6 \( (p < 0.001, t(67) = 4.10; \) Student’s t-test), it is not judged significantly better than generation 12 with Student’s t-test \( (p < 0.05, t(67) = 2.20) \). On the other hand, given that a Wilcoxon test shows significance with \( p < 0.05, Z = 2.04 \), the original MIDI can be considered close in quality yet slightly better than generation 12. Thus evolution guided by the human user eventually achieves in a short number of generations a level of quality close to the quality of the original. This result hints that FSMC-generated parts can meet an acceptable level of quality through evolution. The next section explores whether such results can pass for a natural human composition.

### 4.3.2 Comparing FSMC-augmented Compositions to Fully Human Compositions

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

It is important to note that these pieces are chosen for this experiment because they exemplify entirely human compositions that meet a minimum standard of recognizable quality. By starting with pieces that are convincing as complete compositions, it is possible to discern whether the generated voices reduce the human plausibility of the work, or whether they complement it successfully, as would be hoped for such an approach.
An Additional Voice for Nancy Whiskey

Like the experiment in Section 4.3.1, the interactive evolutionary process for Nancy Whiskey was guided by MaestroGenesis team members. No musical knowledge was applied beyond simply choosing which candidates sounded best. The same experimental settings were applied as in the previous section.

Figure 4.7 shows results after only two generations of evolving additional generated voices for Nancy Whiskey, including both the inputs and the output generated voice. 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. A key issue in understanding the results is the functional relationship between scaffold inputs and CPPN output over time, which gives a sense of the implication of linking these parts functionally. Figure 4.7 shows two measures, numbered 5, and 6, of a generated voice for the MIDI scaffold, Nancy Whiskey. The fiddle, steel guitar, and bass from Nancy Whiskey are input to both the pitch and rhythm networks; the output is a harpsichord voice that inherits pitch and rhythm relationships from the scaffold. One salient such relationship is at the ending of measure six where a G is played in the fiddle part. Although the generated voice does not always follow the fiddle, at this point the output generated voice also plays G at the same time, although an octave lower. However, while it follows the pitch, the generated voice varies the rhythm at this part, borrowing rhythmic elements from the other instruments in the scaffold, thereby further differentiating itself from the fiddle. In totality, the generated voice incorporates pitch and rhythmic elements from all three scaffold instruments while also varying and combining them in novel ways, yielding an original pattern that complements the whole.

Figure 4.8 shows the internal structure of the evolved CPPNs that produce these generated voices and the final generated voice from Bad Girl's Lament in the previous section. In the Nancy Whiskey rhythm CPPN (figure 4.8a), each input is connected to the two outputs with different
Figure 4.7: **FSMC-Generated Additional Voices for Nancy Whiskey.** Measures 5 and 6 of a generated voice evolved for Nancy Whiskey is shown. The top line shows the generated harpsichord output while the next three voices show the fiddle, steel guitar, and bass respectively. The generated voice is produced by CPPNs that relate the scaffold inputs to the generated outputs. The rhythm and pitch relationships between the scaffold and generated voice derive from the functional relationship between them. The 12th-generation additional voice evolved for Bad Girl’s Lament, also addressed in this section, was shown in figure 4.6. Both generated voices can be heard at [http://eplex.cs.ucf.edu/fsmc/dissertation/](http://eplex.cs.ucf.edu/fsmc/dissertation/).

weights. In the pitch network, each input is directly connected to the pitch output with the exception of the second steel guitar input, which connects through a hidden node with a Gaussian function. Recall that output nodes compute a sigmoid function of their input.

Interestingly, the CPPNs for generating Bad Girl’s Lament voices are even simpler and did not even evolve any hidden nodes (i.e. additional nonlinearities beyond the sigmoid output functions). The simplest CPPN of all, which is the pitch network in figure 4.8d, has only a single connection. Such relationships encoded by the CPPNs can also be written mathematically. For example, the Nancy Whiskey rhythm CPPN (figure 4.8a) computes $\text{OnOff} = \sigma(-.35n_1 + 1.34n_2 - 1.76n_3 + 1.46n_4)$ and $\text{NewNote} = \sigma(1.01n_1 + 1.70n_2 - .37n_3 + .51n_4)$, where $\sigma(x) = \frac{1}{1+e^{-1.1x}}$, and $n_i$ is input
Figure 4.8: **CPPNs Corresponding to Generated Voices.** CPPNs for generating additional voices for Nancy Whiskey (a and b) and Bad Girl’s Lament (BGL; c and d) are shown above. Each generated voice can be heard at [http://eplex.cs.ucf.edu/fsmc/dissertation/](http://eplex.cs.ucf.edu/fsmc/dissertation/). Line thickness is proportional to weight and gray lines are inhibitory. The simplicity of the CPPNs shows that the functional relationship between different musical parts is often plausible even if it is simple.

number $i$. In this way, musical relationships really are being encoded as functions. It is important to understand that the simplicity of these relationships resulted from a process of human selection through IEC that ended when the human was satisfied, which means it reflects the human user’s implicit preferences. These results show that simple relationships can be appealing and convincing. In this way, this kind of application can tell us something about the nature of the implicit musical relationships that we appreciate.
Musical Turing Test

In this second listener study, anonymous participants were asked to 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 can generate plausible additional voices it suggests that the theory behind it is at least promising.

For this study, a total of 66 listeners, all of whom are students in a diversity of majors at the University of Central Florida, participated in the study. The full survey, including the human compositions, is provided at http://eplex.cs.ucf.edu/fsmc/dissertation/. The aim is to discover whether the generated voices sound either natural or computer-generated. Participants are asked to rate five different MIDIs 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, <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 MIDIs: (1) an obviously computer-generated control (which helps to establish that participants understand the question), (2) the version of Nancy Whiskey with a generated voice, (3) fully human-composed Chief Douglas’ Daughter, (4) fully human-composed Kilgary Mountain, and (5) the version of Bad Girl’s Lament with a generated voice. Thus the main issue is whether participants judge piece 2 and piece 5, which have generated voices evolved with FSMC, as distinguishable from piece 3 and piece 4, which are entirely composed by humans.
Table 4.2: **Survey Results (lower means more human-like).** The average ratings and standard deviations for the samples show that FSMC generated voices can sound human. The Bad Girl’s Lament MIDI, which is partly computer-generated, ranks less likely to be computer-generated than the fully human-composed song, Kilgary Mountain, although this difference is not significant.

The complete results of this study are shown in table 4.2. On average, the 66 participants judge the intentionally-poor example as significantly more likely to be computer-generated than any other song in the survey ($p < 0.001$, $t(72) \geq 6.42$ according to Student’s t-test and confirmed through a Wilcoxon signed-rank test, $p < 0.001$, $Z \geq 5.80$). This difference indicates that participants understand the survey.

Although the Nancy Whiskey with an additional generated voice is judged significantly more likely ($p < 0.05$, $t(72) = 4.41$ with Student’s t-test and $p < .05$, $Z = 4.02$ with a Wilcoxon signed-rank test) to be computerized than the human song Chief Douglas’ Daughter, it is not judged significantly more likely (with either test) than Kilgary Mountain to be computerized. This result indicates that the Nancy Whiskey with an additional generated voice can pass the musical Turing Test, i.e. people cannot distinguish it from a song that is entirely human-generated.

The Bad Girl’s Lament with an additional generated voice is even more difficult for participants to differentiate. It is not judged significantly more likely to be computer-assisted than either of the human pieces, i.e. Chief Douglas’ Daughter or Kilgary Mountain. In fact, on average, FSMC-generated Bad Girl’s Lament scored slightly *less* likely to be computerized than the entirely human song Kilgary Mountain.

<table>
<thead>
<tr>
<th>Table 4.2: Survey Results (lower means more human-like).</th>
<th>Mean</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MIDI Name</strong></td>
<td><strong>Mean</strong></td>
<td><strong>Std. Dev.</strong></td>
</tr>
<tr>
<td>Control</td>
<td>7.82</td>
<td>2.15</td>
</tr>
<tr>
<td><em>Nancy Whiskey with Add. Voice.</em></td>
<td>5.45</td>
<td>2.65</td>
</tr>
<tr>
<td>Chief Douglas’ Daughter</td>
<td>4.32</td>
<td>2.61</td>
</tr>
<tr>
<td>Kilgary Mountain</td>
<td>4.86</td>
<td>2.39</td>
</tr>
<tr>
<td><em>Bad Girl’s Lament with Add. Voice.</em></td>
<td>4.82</td>
<td>2.44</td>
</tr>
</tbody>
</table>
These results validate that evolved generated voices are at least plausible enough to fool human listeners into confusing partly computer-generated compositions with fully human-composed ones, even though FSMC has almost no a priori musical knowledge programmed into it. The next section discusses the implications of evolving generated voices and listener studies in this chapter.

4.4 Implications

From the results of the listener studies in Sections 4.3.1 and 4.3.2, it is apparent that not only can FSMC produce human quality results, but that IEC can help users effectively navigate the search space induced by FSMC. In the study to determine perceived quality, the average ratings (shown in table 4.1) from generation 1 and generation 12 significantly improved while the quality of the original piece is indistinguishable from that in generation 12, demonstrating that IEC yielded a significant subjective improvement that ultimately re-approached the quality of the original song, yet now with a new additional generated voice.

However, although the musical Turing Test study (shown in table 4.2) suggests that FSMC created generated voices of indistinguishable quality from a human composition, FSMC is not ultimately intended for generating additional voices from complete songs that are already good. The results in this chapter were voices generated from the already-full compositions Nancy Whiskey and Bad Girl’s Lament because they met a threshold of quality recognized in folk music. Thus the quality of the generated voice could be assessed relative to the original benchmark in different stages of development. Yet once this capability is established (through the perceived quality study and others), eventually the aim will be to show that the quality of incomplete songs is actually improved by adding evolved generated voices. Of course, it was difficult to improve the compositions in these studies because they were already complete and well-regarded.
Because the perceived quality study shows that it can take as few as 12 generations to reach an area of the search space of viable generated voices, it is plausible to infer that the FSMC method is not forcing the user to search the entire space of possible generated voices, which would be onerous. Both the CPPN representation and the scaffold help to capture the human essence of a pre-existing song, which the generated voice can transform to sound plausible. In effect, FSMC “steals” the quality inherent in the scaffold and then manipulates it to create something new, thereby feeding on the skill of the human originator. Therefore, the FSMC method potentially can help users to find appealing generated voices faster than could be found without such a method. This principle should even work with more complex musical pieces because the complexity in the scaffold will still be reflected in the output of the CPPN that transforms it.

4.4.1 Simplicity of Functional Relationships

While most generation methods require an extensive corpus to analyze or prior music knowledge, these studies show that simple functional relationships are central to musical composition and therefore possible to exploit. The Musical Turing Test study confirms that such generated voices can be indistinguishable from fully-human compositions by average listeners.

Not only is the insight simple that a functional relationship is foundational to the idea of generated voices, but the evolved relationships between the generated voices and the scaffold themselves also turned out to be simple. Nancy Whiskey contains a single hidden node in its rhythm and pitch CPPNs, indicating at most two function compositions. Generating voices for Bad Girl’s Lament is even simpler.

Of course, these results are anecdotal and do not imply that complex relationships would not be appealing in some cases, yet they do raise the intriguing hypothesis that many such relationships could be simpler than they sound intuitively.
4.5 Conclusions

This chapter provides insight into the relationship between musical parts through a new theory called functional scaffolding for musical composition (FSMC). By representing the relationship between existing parts and a new computer-generated voice functionally, plausible additional voices are generated. Elaborating on these patterns through IEC gives inexperienced composers the opportunity to explore a space of additional computer-generated voices for their own compositions. The resulting pieces and additional generated voices are even sometimes indistinguishable from fully-human compositions. The main conclusion is that FSMC provides an alternative to data-intensive approaches to music generation and analysis that nevertheless promises a different kind of insight into the nature of generated musical voices.

While the next chapter naturally focuses on improving the method further, FSMC in effect opens up a new direction in research on evolutionary music generation by providing a succinct and effective theory based on a simple principle from which to build, i.e. that the different parts of a song are functionally related.
CHAPTER 5: GENERATING POLYPHONIC ADDITIONAL VOICES FOR SINGLE MONOPHONIC STARTING MELODIES

Among the most important functions of any approach to enhancing human creativity is what Boden [7] terms *transformational creativity*. That is, key creative obstacles faced by human artists and musicians are the implicit constraints acquired over a lifetime that shape search space structure. By offering an instance of the search space (e.g. of computer-generated musical voices) with a radically different structure, a creativity-enhancing program can potentially liberate the human to discover unrealized possibilities. In effect, the familiar space of the human artist is *transformed* into a new structure intrinsic to the program. Once the user is exposed to this new world, as a practical matter the program must provide to the user the ability to *explore* and *combine* concepts within the newly-conceived search space, which corresponds to Boden’s *combinatorial* and *exploratory* classes of creativity [7]. That way, the user experiences a rich and complete creative process within a space that was heretofore inconceivable.

The danger with transformational creativity in computational settings is that breaking hard-learned rules may feel unnatural and thereby unsatisfying [8]. Any attempt to facilitate transformational creativity should respect the relationships between key artistic elements even as they are presented in a new light. Thus for a given domain, such as computer-generated musical voices, a delicate balance must be struck between unfettered novelty and respect for essential structure.

Many approaches to generating music focus on producing a natural sound at the cost of restricting creative exploration. Because structure is emphasized, the musical space is defined by rules that constrain the results to different styles and genres [90, 13, 20]. The necessity for a priori rules potentially facilitates the combination of musical structures or exploration of the defined space, but precludes transformational outcomes.
In contrast, musical structures in FSMC are defined as the very functions that relate one part of a piece to another, thereby enabling satisfying transformational creativity [34, 35]. The user-guided creative exploration itself is facilitated by an interactive evolutionary technique that in effect allows the user to breed the key functional relationships that yield computer-generated voices, which supports both combinatorial and exploratory creativity [7] through the crossover and mutation operators present in evolutionary algorithms. By representing music as relationships between parts of a multipart composition, FSMC creates a new formalism for a musical space that transforms its structure for the user while still respecting its fundamental constraints.

The previous chapter showed that FSMC can produce musical voices that are indistinguishable by listeners from fully human-composed pieces. However, the voices in these studies was only a single monophonic instrument, leaving open the key question of whether a user with little or no musical expertise can perhaps generate an entire multipart arrangement with this technology from just a single-instrument monophonic starting melody. If that were possible, then anyone with only the ability to conceive a single, monophonic melody could in principle expand it into a complete multilayered musical product, thereby enhancing the creative potential of millions of amateur musicians who possess inspiration but not the expertise to realize it. Originally published in [36], this chapter demonstrates that FSMC indeed makes such achievement possible.

5.1 Extending Functional Scaffolding for Music Composition

This section extends the original FSMC approach described in Chapter 4, which only evolved a single monophonic additional voice [34, 35]. It explains the core principles of the approach and how they are applied to producing multipart computer-generated voices.

A crucial aspect of any creativity-enhancing approach for music composition is first to define the
musical space. Users can help define this space in FSMC by first selecting a musical starting point, i.e. the monophonic melody or *scaffold*. Initial scaffolds can be composed in any style and if they are only single monophonic parts as in this chapter, they 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 computer-generated voices can be created with only this initial scaffold. Because of the relationship of different musical voices to the scaffold and therefore to each other, the space is easily created and explored.

This chapter introduces a new layering technique whereby generated voices from previous generations can serve as inputs to new CPPNs that then generate more layers of harmony. The result is the ability to spawn an entire multi-layered piece from a single monophonic starting melody.

One such layering approach is performed by generating one new monophonic voice at a time. The first layer is the monophonic melody composed by the human user. The second layer is generated through FSMC from the first. The third layer is then generated through FSMC by now inputting into the CPPNs the first and second layers, and so on. All of the layers are finally combined to create an entire set of generated voices, resulting in voices that are functionally related to both the initial melody and previous generated voices. In this way, each additional voice is slightly more removed from the original melody and subsequent generated voices are based functionally on both the scaffold and previously-generated lines.

To create additional voices more closely related to the original melody, another layering technique is for users to generate all layers from only the single monophonic starting point. For this purpose, the CPPNs are given enough outputs to represent all the instruments in the generated voices at the same time. Because the melody and the generated voices are functionally related, any generated voice will follow the contours of the melodic starting point. However, in this case, the only influence on each voice is this starting point itself, yielding a subtly different feel.
With either of these approaches or a combination of them users can further influence their generated voices by holding constant the rhythm CPPN or pitch CPPN while letting the other evolve. Interestingly, when two generated voices share the same rhythm network but differ in the pitch network slightly, the two monophonic instruments effectively combine to create the sound of a polyphonic instrument. Similarly, the pitch networks can be shared while the rhythm networks are evolved separately, creating a different sound. Notice that this approach requires no musical expertise to generate multipart compositions.

### 5.1.1 Experiments

The experiments in this section are designed to show how users can generate multipart pieces from just a single monophonic melody with FSMC. They are divided into music generation and two studies: a listener study that establishes the quality of the compositions and a creative self-assessment from users of the program. 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.

**Procedures for Polyphonic Generated Voices**

For this experiment, three undergraduate independent study students composed in total three monophonic melodies. From each of these user-composed melodies, a multiple generated voices to be played simultaneously were created through FSMC by the author of the originating melody. Two other collections of generated voices were generated by one of the students for the folk song, *Early One Morning*. The most important point is that no musical expertise was necessary to apply to the final creations beyond that needed to compose the initial monophonic melody in MIDI format.
Thus, although results may sound consciously arranged it is important to bear in mind that all the polyphony you hear is entirely the output of FSMC. The original melodies, generated voices, and CPPNs are available at http://eplex.cs.ucf.edu/fsmc/dissertation/.

FSMC 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 generated at once by outputting 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, while others might depend on multiple previous layers. In effect, such decisions shape the subtle structural relationships and hence 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 harmonious partnerships. As the remainder of this section describes, the student composers took advantage of this latitude in a variety of ways.

Early One Morning, (Song 1) versions 1 and 2 with four- and five-part generated voices were started by undergraduate, Marie E. Norton, from an initial monophonic melody transcribed from the traditional, human-composed folk song. The second layer is identical in both versions and was evolved from Early One Morning itself. The third, fourth, and fifth parts of version 1 were all evolved from the second layer. The third, fourth, fifth, and sixth parts of version 2 were evolved from the pitch network of the second layer of version 1, and the rhythm network from the original Early One Morning monophonic melody. This experiment illustrates that the results with FSMC given the same starting melody are not deterministic and in fact do provide creative latitude to the user even without the need for traditional composition techniques.
Song 2 started from an original monophonic melody composed by undergraduate Marie E. Norton. The second layer was added by inputting this melody into the rhythm and pitch networks of the subsequent generated voice populations. This second layer then served as input to the pitch and rhythmic CPPNs for layers 3 and 4. The pitch CPPN for layer 5 took the output from layer 2 as input, but the rhythm network only had a bias (i.e. constant) input. Finally, the inputs for the pitch network for layer 6 were layers 3, 4, and 5, while the inputs to the rhythm CPPN were layer 4 and a measure timing signal first introduced for NEAT Drummer by Hoover and Stanley [31] that gives the network a sense of where the song is within the measure. All of the layers finally combined to create a single, multipart piece in which each line is functionally related to the others. Each layer took as few as three to as many as five generations to evolve.

For Song 3, Zachary Merritt first created a layer that influences most of the other layers, but is not heard in the final track. The fourth layer was generated from the third, which is influenced by the monophonic melody and the unheard layer. The fifth layer was generated from the population of the fourth layer with the rhythm network held constant to create a chordal feel. The sixth layer was generated from only the initial starting melody and another special timing signal that imparts a sense of the position in the overall piece [31]. Similarly, the seventh layer is generated from only the initial starting melody, but adds a separate timing input, \(\sin(\pi x)\), where \(x\) is the time in measure. Although there are seven layers described in this procedure, only six were selected to be heard, which means that there is a five-part collection of generated voices.

Finally, undergraduate Trevor A. Brindle created an initial piece and evolved all five generated voices for Song 4 directly from it. Instead of inputting results from previous generations, he started new runs for each voice from the same scaffold, giving a strong influence to the melody.

Notice that the key decisions made by the users are in general from which tracks to generate more tracks. Of course the users also performed the IEC selection operations to breed each new layer.
Figure 5.1: **Early One Morning Versions.** The first four measures of versions 1 and 2 of Early One Morning illustrate how a single user with the same monophonic starting melody can direct the generated voice in two different ways that nevertheless both relate to the initial melody. Because the generated voices share two of their layers, they sound related. However, through timbre selection and the evolution of two and three distinct layers in versions 1 and 2, respectively, the user imparts a different feel.

Importantly, none such decisions require musical expertise.

*Polyphonic Results*

Samples of the scores for the two arrangements created to accompany Early One Morning are shown in figure 5.1. The layers are shown in order from top to bottom in both versions (layer 1 is the original melody). Layer 2, which is the same in both versions, is heard as violin II in version 1 and viola in version 2.

An important observation is that the violoncello part in version 1 follows the rhythm of the initial starting melody very closely while the pitch contour differs only slightly. While the viola and double-bass parts differ in both pitch and rhythm over the course of the song, both end phrases and subphrases on the tonic note, F, in many places over the course of the piece, including measure 4
in figure 5.1a. Version 2, on the other hand, contains many rhythmic similarities (i.e. the eighth note patterns contained in the keyboard I, viola, keyboard II, and the violin II parts), but illustrates distinct pitch contours. Together, the two versions illustrate how a single user can generate different generated voices from the same initial monophonic starting melody and how the initial melody exerts its influence both rhythmically and harmonically.

Songs 2, 3, and 4 (where scores and audio are available at the accompanying website) exhibit a similar effect: rhythmic and harmonic influence from the original melody, yet distinctive and original generated voices nevertheless. The result is that the overall arrangements sound composed even though they are evolved through a breeding process. The next section provides evidence that impartial listeners also appreciate the contribution of the human user.

Assessing the Human Contribution

The contribution of users to the quality of the generated works and accordingly the effectiveness of the creativity enhancement is evaluated through a listener study.

The listener study consists of five surveys, one for each generated arrangement. The surveys present two MP3s to the listener, who is asked to rate the quality of both. The first MP3, called the collaborative arrangement, is an arrangement resulting from the collaboration of the author with the program (i.e. the two versions from Early One Morning or Songs 2, 3, or 4). The second, called the FSMC-alone arrangement, is generated by the program alone. That is, a random pitch CPPN and a random rhythm CPPN are provided the same monophonic starting melody as the collaborative arrangement and their output is taken as the FSMC-alone arrangement. Thus the factor that is isolated is the involvement of the human user, who is not involved in the FSMC-alone arrangement. However, it is important to note that the FSMC-alone arrangements do not actually sound random because even if the CPPNs are generated randomly, they are still functions
of the same scaffold, which tends even in the random case to yield outputs that sound at least coherent (which is the motivation for FSMC in the first place). Thus this study investigates whether the human user is really able to make a creative contribution by leveraging FSMC. That is, any perceived difference in quality between the FSMC-alone and collaborative arrangements can be attributed to the human users, showing that their efforts are perceived by listeners.

A total of 129 students participated as listeners in the study. The full survey is available at http://eplex.cs.ucf.edu/fsmc/dissertation/, but note that in the administered surveys, the order of the MP3s was random to avoid any bias. The users were asked to rate each piece with the following question:

Rate MIDI \( i \) on a scale of one to ten. (1 is the worst and 10 is the best),

where \( i \) refers to one of the ten generated works. The idea is that if the user-created arrangements are rated higher than those generated by FSMC-alone, the user’s own input likely positively influenced the outcome.

The results of the listener study in figure 5.2 indicate that all of the collaborative arrangements are rated higher than those generated with FSMC alone, with three out of five (Song 1 version 2, Song 3, and Song 4) displaying significant difference (\( p < 0.05, t(30) = 4.33, t(29) = 5.96, \) and \( t(20) = 2.39 \) respectively with Student’s paired t-test and Wilcoxon signed-rank, \( Z \geq 2.14 \)). Taken all together, the collaborative arrangements sound highly significantly more appealing than those generated with FSMC alone (\( p < 0.001; \) Student’s paired t-test and Wilcoxon signed-rank). These results indicate that not only does FSMC provide a structurally plausible search space, but that it is possible to explore such a space without applying musical expertise. That is, the results suggest that the user input significantly improves the perceived quality of the generated compositions.
Figure 5.2: Listener Study Results. The average rating (by 129 participants) from one to ten of both the collaborative and FSMC-alone arrangements are shown side-by-side with the lines indicating a 5% error bound. The overall results for the listener study indicate that on average the collaborative arrangements are of significantly higher perceived quality than FSMC-alone.

Creative Self-Assessment

An important aspect of a program aiming to facilitate creativity is how users perceive the effect of the program on their own creative self-expression. For this purpose, 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. The survey questions are available at http://eplex.cs.ucf.edu/fsmc/dissertation/.

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. In fact, one student claimed that “FSMC freed me from my normal stylistic tendencies;” while another indicated that “I typically follow a sort of pattern when I compose, but FSMC expanded my thinking.” Said another, “Specific parts of the output harmonies were very good, and I could see myself applying them in many places throughout the song.”

Furthermore, when asked to describe the advantage of integrating FSMC into the respondents’ own musical creativity process, one student replied, “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.” Another said, “a few of my stylistic elements will come through,” but that “other elements will surface” that had not been considered. The third student claimed that FSMC was “great for writer’s block.” Thus the innovations pushed users outside of 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 expected. 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 user said, “I could not shape the harmony produced to suit my melody’s form. I would need to input the harmony produced into Sibelius [a sequencing program from Avid Technology, Inc.] to make final corrections and changes.” Although the functional representation ensures that the generated voice follows the pitch and rhythmic contours 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, an issue that is being addressed in future MaestroGenesis improvements.

While the users wanted more from the technology, 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.”

5.2 Implications

A key feature of figure 5.1 is that the collaborative arrangements generated by users with the assistance of FSMC follow the melodic and rhythmic contours of the original scaffold. Furthermore, the listener study suggests that FSMC helps the user establish and explore musical search spaces that may otherwise have been inaccessible.

While the users search this space through IEC, which facilitates the combination of musical ideas and the exploration of the space itself, an interesting property of this search space is its robustness; even FSMC-alone arrangements, which are created without the benefit of human, subjective evaluation, can sound plausible. However, when coupled with the human user, this approach in effect transforms the user’s own internal search space of possible generated voices to one constrained by functional scaffolding.

While the quantitative data suggests the merit of collaborative arrangements, music is inherently subjective. Therefore, it is important for the reader to judge the results for his or herself at http://eplex.cs.ucf.edu/fsmc/dissertation/ to fully appreciate the potential of the FSMC method.

Perhaps most importantly, with only a single, monophonic melody, users could compose entire multipart pieces without the need for explicit musical knowledge. Even if not at the master level, such a capability opens to the novice an entirely new realm of exploration.
5.3 Conclusion

This chapter presented an extension to functional scaffolding for musical composition (FSMC) that facilitates a human user’s creativity by generating polyphonic compositions from a single, human-composed monophonic starting track. The technique enables creative exploration by helping the user construct and then navigate a search space of candidate generated voices through a breeding process akin to animal breeding called interactive evolutionary computation (IEC). These collaborative arrangements bred by users were judged by listeners against those composed only through FSMC. Overall, listeners liked collaborative arrangements more than the FSMC-alone arrangements. Most importantly, a promising potential for creativity enhancement in AI is to open up the world of the amateur to the domain once only accessible to the expert. The approach in this dissertation is a step in this direction.
CHAPTER 6: MAESTROGENESIS VOICE

As interest in creativity support tools increases [71, 42], coders and composers alike are increasingly building new tools to augment their individual creative processes [43, 24, 81]. For instance, popular musician David Bowie famously co-developed a program called Verbasizer with Ty Roberts for lyrical inspiration [83, 67] while David Cope’s EMI, which generates music in the style of famous composers, was developed to overcome Cope’s own musical “writer’s block” [21]. These programs offer their originators a chance to generate specialized musical and lyrical ideas while integrating with their natural and established processes of composition. However, because many commercially available programs often address singular aspects of composition, the endpoint in the creative landscape (i.e. an artifact) is left for the user to discover in isolation.

While results in Chapter 4 illustrate that MaestroGenesis can spark creativity in amateur musicians, users may lack the expertise or time to transcribe a musical idea to MIDI (for input to MaestroGenesis) before even attempting to generate additional voices. The aim in this chapter is to facilitate the process by which users interact with MaestroGenesis through the development of a new plugin called MaestroGenesis-Voice (MG-V) that enables composition of the scaffold through raw audio singing. Like SongSmith [73] and LaDiDa[74], MaestroGenesis users can sing into the program and generate additional voices; however, in contrast FSMC offers more control over generated output, a common complaint amongst users [74].

Unlike the continuous scaffold MIDIs in Chapter 5, human singing that is intended to include instrumental accompaniment often contain rests where the additional instrumentation is expected. Because MaestroGenesis CPPNs are queried and output information at the current position in the song or tick, the additional generated voice would hear only rest at such points in the scaffold. By convention, the value of a rest is set to the tonic or first note of the key, which is an arbitrary
selection for the pitch of silence. The resulting generated voices have little choice but to rest or output a rhythmic pattern with the pitch generated by the CPPN at the tonic for the duration of the rest in the scaffold. To encourage more variability and initial musical structure, MG-V therefore also helps users choose chord progressions that are then combined with the original melody and input to MaestroGenesis.

The experiments and results in this chapter are designed to qualitatively compare voices generated from raw audio alone, and those with raw audio and an MG-V chord accompaniment. By simplifying the composition process with MG-V, the hope is to provide an enjoyable user experience such that amateur musicians can experience a sense of ownership upon hearing their pieces.

6.1 Extending MaestroGenesis with MG-V

A natural place to begin facilitating music composition is through human vocalization. This section describes an extension to the approach developed in Chapter 5 that is implemented in MG-V. It describes both how the plugin transcribes human voices to MIDI and how MG-V chord progressions are generated to build a strong scaffold from which to generate additional instrumental voices in MaestroGenesis.

6.1.1 MeloTranscript

Because MaestroGenesis relies on accurate MIDI input, a crucial aspect of MG-V is to transcribe the human voice accurately to MIDI before generating additional voices. Through a library called MeloTranscript [80] that is designed to convert monophonic melodies to MIDI, users sing into MG-V through an interfaced designed and implemented by Paul Szerlip (a member of the MaestroGenesis team) where their raw audio is converted to MIDI (as shown in figure 6.1).
Figure 6.1: **Singing into MaestroGenesis.** By pressing the microphone at the top of the screen in (a), users begin recording their song as represented by the waveform rectangle-visualization below it. The rectangles scroll as the song continues, minimize when the singer rests, and grow with the input volume. In (b) the MIDI transcription is shown with the waveform where a gray sliding bar indicates the current position in both the waveform and the MIDI throughout playback.

While singing, users see a waveform visualization of their voice (figure 6.1a) that can help gauge volume consistency and indicate rests and breathing patterns. Upon pressing the stop button, the completed melody and MIDI transcription are played simultaneously and displayed as shown in figure 6.1(b). MIDI notes are represented as bars whose height and length correspond to the note’s pitch (i.e. higher notes indicate a higher pitch) and length indicates duration. Pitches can be altered
by dragging the bars up or down while its position in the song can be adjusted by dragging the bars left or right. In this way, users can adjust the MIDI file to correct for imperfections before it is input into MaestroGenesis.

6.1.2 Chord Generation to Enrich the Scaffold

In most popular music, chord progressions form the basis upon which a song is built [47]. Specific chords and the order in which they are played are important for determining stylistic characteristics like whether a song has a blues or fifties feel and also enforcing that genre in music created by generative systems. For experienced musicians, these chords also suggest how to approach composition through theories that suggest that certain melodic notes sound better with particular chords and positions within a measure. While the previous two chapters focus on reducing information contained in the scaffold, initial results with MG-V illustrate that even simple chord progressions when added to the scaffold can increase musical quality of additional generated voices created from a human voice.

To overcome the issue that voiced monophonic melodies intended for instrument accompaniment often contains rests, I implemented a chord generation interface to help users develop chords for their original monophonic melodies. Each chord is a triad (i.e. three simultaneously played notes) that starts on one of the seven notes of a user-specified diatonic scale and contains every other note until three notes are chosen. For example, in the key of C major with notes C, D, E, F, G, A, and B, the triad that starts on C contains notes C, E, and G. Because the interval from one note to the next in a diatonic scale can be uneven (e.g. D to E is a whole step in tone whereas E to F is a half step), triads have different sounds. Those starting on the first, fourth, and fifth notes of the scale are major or perfect (notated as I, IV, V), while those on the second, third, and sixth notes (notated as ii, iii, and vi) are minor; triads on the seventh are diminished (vii°).
Table 6.1: **Chord Transitions.** Because certain chords sound better following others, a transition table of valid movements is established through music theory. The left hand side represents the current chord, while each list on the right contains possible choices for the next. Because MG-V requires that a measure must share a chord tone with the suggested chord, the chords shown are a proper subset of those available in the program. Transition probabilities are equally weighted.

<table>
<thead>
<tr>
<th>Chord</th>
<th>Next Chord</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I, ii, iii, IV, V, vi, vii°</td>
</tr>
<tr>
<td>ii</td>
<td>V, vii°</td>
</tr>
<tr>
<td>iii</td>
<td>iv, vi</td>
</tr>
<tr>
<td>IV</td>
<td>I, ii, V, vii°</td>
</tr>
<tr>
<td>V</td>
<td>I, vi</td>
</tr>
<tr>
<td>vi</td>
<td>ii, IV</td>
</tr>
<tr>
<td>vii°</td>
<td>I</td>
</tr>
</tbody>
</table>

One method for determining chord appropriateness for a particular measure in MG-V is assessing whether shared tones exist between the melody and chord (i.e. chord tones). The most liberal strategy is to find only a single matching note in the melody and to suggest an appropriate corresponding chord. While there can be many different chord choices for each measure, before shown to the user, they are first gathered and judged against the transition table in 6.1 to establish whether the sequence is musically sensible. Starting from the first measure of the melody, users are then prompted, measure-by-measure, with chord choices that match both the input melody and follow transition rules for creating appealing chord progressions (shown in table 6.1). These suggestions can be played with the original scaffold melody to gain an intuitive understanding of the choice, and once finalized are loaded into MaestroGenesis for conventional FSMC.

The next section describes experiments on generating additional voices with both the MeloTranscript and the chord generator components of MG-V.
6.2 Experiments

The experiments in this chapter are designed to uncover qualities of generated voices that can be created from melodies sung into MG-V, and whether enriching the scaffold has a noticeable effect on the output.

For the first experiment, three additional voices are generated from only the MIDI transcription of a melody sung by a user. The melody selected for this experiment is the song For Today, originally by Jessica Lea Mayfield and shown with her permission. It is sung by the user into MG-V and the MIDI transcription is shown in figure 6.2. The aim is to determine the base level of quality possible when generating from monophonic raw audio.

Because lyrical lines can sustain long periods of rest and amateur musicians may be unable themselves to augment the scaffold with additional chords, the second experiment examines generated voices evolved from both the original lyrical line in figure 6.2 and a chord progression generated by MG-V (shown together in figure 6.3). Three additional voices are generated from the original lyrical line (in the key of B major: B, C#, D#, E, F#, G#, A#). The goal is to investigate the added benefit of generating music from scaffolds with chord progressions.

Because many amateur musicians compose music through singing or on their chosen instruments, this new approach to generating additional voices presents a more natural composition process than composing an initial scaffold in MIDI. This study demonstrates to potential users how MaestroGenesis can enhance their own recordings by simultaneously playing both the original track and the generated additional voices, potentially increasing the impact of MaestroGenesis in the amateur musician community.
I was walking with your left hand in my back pocket and I stared at the sky as you kissed me.

You were running away away from everything so I stayed behind so you'd miss me.

---

Figure 6.2: **First Fourteen Measures of Lyrics for For Today.** Users begin composing with MaestroGenesis by singing into MG-V. This scaffold is a song selected by a MaestroGenesis team member, undergraduate Jessica Sprague, called For Today by Jessica Lea Mayfield and is shown with her permission. Both the MIDI transcription and raw audio are available at http://eplex.cs.ucf.edu/fsmc/dissertation

6.3 Results

Results from the first experiment in figure 6.4 show measures six through nine from the three evolved voices for the MIDI shown in figure 6.2. As expected from MaestroGenesis, each generated voice follows the contours of the original melody. However, when the scaffold rests in the seventh measure through to the eighth, each voice can only indicate a rest or rhythmic variations of the pitch generated for the tonic of the key. For example, figure 6.4b sustains an F# during the rest in measure 7. Typically in human compositions this silence in the melody would be an opportunity for other instrumentalists to showcase their own talents by introducing new pitch and
rhythmic contour. However, when only “listening” to the scaffold silences, the generated voices are harmonically static.

To show the effect of adding chords to the voice-based scaffold, the previous scaffold is externally enriched by incorporating a simple chord progression generated by MG-V. Figure 6.5 shows one of the three additional voices generated from both the combined melody and chord progression. Unlike the results in figure 6.4, the additional generated voice is free to switch pitches when a rest is encountered in the scaffold, thereby illustrating the additional variety that can be achieved through added chord progressions.

6.4 Conclusion

The new MaestroGenesis plugin MG-V allows users to create additional voices from scaffolds sung in their own voice. Because lyric lines often contain sustained rests through transitions (e.g. transitions from verses to the chorus), MG-V helps users determine appropriate chord choices for each measure of the scaffold. Songs are presented that have been created from both the user’s voice and a combined scaffold including chords generated with MG-V. The result is that chord generation can ease potential problems encountered when singing a monophonic melody for scaffold input to MaestroGenesis.

Moreover, the pipeline from recoding raw audio, to transcription, to chord generation, to evolved accompaniment represents the first such working sequence of algorithms to be introduced. Each step in this pipeline presents its own significant technical challenges and solutions, and each must feed seamlessly into the next. The results in this chapter represent such a working sequence and thereby signify that systems requiring no musical expertise beyond the ability to hum a tune are now a feasible research direction for computer-assisted music generation.
Figure 6.3: First Fourteen Measures of Lyrics and Generated Chords for For Today. In MG-V, users choose the chords to accompany their audio. The raw audio, its MIDI transcription, and the generated chord sequence are available at http://eplex.cs.ucf.edu/fsmc/dissertation
Figure 6.4: **Generated Voices from Monophonic Scaffold.** Measures sixth through nine in (a), (b), and (c) typify three response types articulated by the generated voice when encountering the measure-long rest in the scaffold (at measure seven). In (a) the additional voice rests, in (b) the note generated for the tonic of the key (B Major) is the only pitch available and sustained through the rest, and in (c) note generated for the tonic of the key is articulated throughout the measure.
Figure 6.5: **Additional Generated Voice with Chord Progression.** In measure 7, a rest is encountered in the scaffold that would typically leave little room for variation in the additional generated voice. However, because of the chord progression in the scaffold, at measure 7 the additional voice is able to play a different note than the one previously played before the rest began. This result with an additional generated voice is available at [http://eplex.cs.ucf.edu/fsmc/dissertation](http://eplex.cs.ucf.edu/fsmc/dissertation).
CHAPTER 7: DISCUSSION AND FUTURE OPPORTUNITIES

FSMC follows the idea that music can be represented as a function of time and that additional musical voices can be generated from preexisting pieces through functional scaffolding. With almost no musical knowledge, complete polyphonic pieces can be created from as little as a monophonic starting melody. The next section discusses what this minimal approach to music generation implies for human music appreciation. The second then explores the importance of scaffolds in FSMC and the third discusses its creativity-enhancing effects.

7.1 Implications for Musical Appreciation

While experienced human composers often synthesize a substantial knowledge of musical rules and techniques, FSMC composition occurs only through functional transformations of a given scaffold. However, such transformation 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.

By encompassing all possible transformations, FSMC induces an entire space of generated musical voices that acknowledge and respect the contours of the provided scaffold. For example, the generated voice for Nancy Whiskey (Section 4.3.2) demonstrates that quick discovery of plausible generated voices is possible. However, perhaps more importantly, the space of transformations can be searched intuitively to find increasingly appealing generated voices. The family of voices generated for Bad Girl’s Lament confirm that through each generation, the voices, and the musical rules that they represent, can be directed to increasingly satisfying areas of the space.

An interesting aspect of FSMC is that the formal concepts that correspond to discovered trans-
formations 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, NEAT Drummer users, nor their own designers need to recognize cadence types, specify where they should occur, or even know what a cadence is. The CPPN representation instead can produce holistic transformations that can affect many smaller aspects of composition simultaneously. Users simply listen to generated musical voices and select those that conform to their own musical tastes, regardless of the particular musical rules that are obeyed, mutated, or even creatively broken.

In fact, because the emphasis is on generating plausible voices rather than on conforming to musical rules, the search process has the potential to yield satisfying additional voices that nevertheless do not follow the rules. Interestingly, as illustrated by the generated voices in this dissertation, the average listener can enjoy musical voices even if they do not completely adhere to compositional tradition. This observation suggests that 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 is key to its appreciation. As Nicholas Cook wrote in *Music, Imagination, and Culture* [18],

> 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.

While many factors influence the subjective evaluation of a musical piece such as the composer’s reputation and previous works, or his or her goals and motivations, the results in Chapters 3, 4, and 5 were arguably appreciated in the absence of such external context (i.e. in blind quality assessments). Users appreciated FSMC-generated voices based solely on their own personal taste.
What the perceived quality of these pieces illustrates is again that one of the overlooked features of music appreciation is that it seems to depend on the ability to recognize the functional relationships between parts of a piece.

Furthermore, because an essential aspect of appreciation may be these 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 spend countless hours listening to e.g. atonal works, analyzing their composition structures, and recreating their own pieces in such a style while working toward understanding and appreciating these types of pieces. Perhaps at some level they are learning the functional relationships that relate parts of such music to each other. However, these functional relationships may also in part 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 a simple new approach to understanding the elusive nature of music appreciation.

7.2 Importance of Scaffolding

The idea of scaffolding, i.e. deriving several parts from a preexisting pattern, means that the most profound effort in musical creativity can be largely concentrated on a relatively small part of the overall composition. From the comparison experiments in Chapter 3, it is clear that something must form the scaffold from which all other generated voices are derived.

Even with a scaffold, different styles of generated voices may generally be more or less difficult to discover. For example, can convincing jazz walking bass be generated even in the context of
other jazz instrumental tracks? Certainly it is possible that the interactive evolution process can discover a function that expresses a particular style, yet the likelihood of such a discovery depends on to what extent the style is already embodied by the existing scaffold. The extent to which the scaffold contains essential stylistic cues, combined with the complexity of the function that would create the right style in the absence of such cues, determines the difficulty of the discovery. Thus this work does not diminish the considerable human effort required to acquire specialized styles of composition.

However, how much information must the scaffold contain to generate plausible voices? Experiments in Chapters 3 and 4 scaffold drum pattern and pitch voices with complete polyphonic compositions while Chapter 5 produces compelling pieces from as little as a single monophonic starting melody. Thus the range of musical information that a scaffold must contain to compose plausibly can vary.

However, even the scaffolds in previous chapters do not solve the fundamental problem of generating the scaffold itself. What kind of process can create the initial pattern? Interestingly, it is possible that even an individual instrumental part can be generated from an even more abstract underlying scaffold, i.e. one that is never actually heard, like the conductors in Chapter 3. These abstract patterns represent musical structures below the level of the explicit notes and pauses. Rather, they are the shape of the fabric upon which such notes are woven.

7.3 Creativity Enhancement

A promising application of this new musical discovery process is in creativity enhancement. Many approaches in this area are restricted by the representation of musical knowledge in the system; a successful composition in such approaches depends in part on the designer’s ability to identify
and reasonably apply key compositional rules [52]. However, decades of research illustrates the complexity of this task. While built-in rules may result in appealing musical pieces, they constrain a full exploration of musical possibilities. In contrast, because FSMC requires 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.

Interestingly, even with the same scaffold, users are able to direct their generated voices toward more than one outcome of their liking, as shown in the two versions of Early One Morning in figure 5.1, the two versions of Johnny Cope in figure 3.5, and the three versions of Oh! Dem Golden Slippers in figure 3.7. Although both versions of Early One Morning begin with the same monophonic melody, the same user generated entirely different directions for each piece. Even more changes might be observed if evolved by a different individual with his or her own concepts of musical style and taste. Thus MaestroGenesis and NEAT Drummer have the potential to help users explore their own creative space.

Results in this dissertation show that MaestroGenesis composes musical ideas that users may not have otherwise considered, but users’ experience with MaestroGenesis suggest that composing MIDI scaffolds is more challenging and less personalized than simply singing or humming an original melody. To further explore the bounds of creativity enhancement, MG-V allows users to sing the scaffold and hear it played simultaneously with candidate additional voices generated in MaestroGenesis. By also helping users build chord progressions, MG-V promises to provide more opportunity for creativity enhancement to nonmusical users.

In his best-selling book, Levitin [49] 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.
CHAPTER 8: CONCLUSION

This dissertation introduced functional scaffolding for musical composition (FSMC), a method that augments human creativity by generating rhythmic and melodic generated voices to help users complete their existing compositions. From as little as a single monophonic starting melody that users can sing, called a scaffold, FSMC helped users add any number of generated voices to their original works. In contrast to other music-generating systems, FSMC creates tailored parts that inherently follow the rhythmic, melodic, and harmonic contours of the user-provided scaffold.

These FSMC-generated parts each result from user interaction with one of two programs that implement FSMC: NEAT Drummer, a program that composes drum patterns, and MaestroGenesis, which generates melodic and harmonic compositions. Both programs help users construct a search space of possible additional generated voices that they can then navigate through a process akin to animal breeding called interactive evolutionary computation (IEC). In this process, users read or hear a set of FSMC-proposed generated voices and select those that sound most appealing. Then the next set of candidates, which are based on the previously selected “parent” voice, inherit some of its appealing traits. Thus, by iteratively selecting the most appealing generated voices, users can converge on personalized and plausible generated voices to complete their compositions.

Furthermore, initial results illustrate that FSMC users break through their writer’s block to generate musical compositions that sometimes can be confused with fully human-composed works. While many approaches in automated composition focus on generating music through well known musical theories and formulas, FSMC is the first to explore the simple hypothesis that functional relationships alone are enough to account for much of musical appreciation.

In sum, FSMC was demonstrated in this dissertation through five major contributions. The first illustrates through drum patterns that complement and elaborate a preexisting piece that the scaf-
fold provides a human element to the generated pieces. The second establishes the quality of the additional voices with a listener study that shows that some partially FSMC-composed pieces are indistinguishable from those that are fully human-composed. The third contribution explores how MaestroGenesis helps users construct and explore a creative space of additional generated voices, while the fourth contribution covers a two-year study on its creativity enhancing affects. Finally, through a plugin called MaestroGenesis-Voice (MG-V), the fifth contribution lets users generated additional voices from their own singing.

An important aspect of NEAT Drummer and MaestroGenesis is that because they require human participation, a session with NEAT Drummer or MaestroGenesis reflects the creativity of both the user and the algorithm. It is an interesting question whether the user can be entirely eliminated, allowing the computer to compose completely on its own. However, while neither NEAT Drummer nor MaestroGenesis eliminate the need for human input, what they do eliminate is the need for human expertise by shifting the creative focus from composition to opinion (i.e. what sounds the best). In this way, a significant obstacle to widespread, high-quality musical creativity is removed. Thus the promise of this work is that it opens an intriguing new avenue to computer-generated music that raises interesting questions about how music is encoded and generated by humans.
APPENDIX A: SURVEY QUESTIONS
A.1 Perceived Quality Listener’s Study in 4.3.1

The six questions appear as survey participants saw them for the listener study described in Chapter 4, Section 3.1, Investigating the Evolution of Generated Voices. The accompaniment tracks in MIDIs 3, 4, 5 were generated by functional scaffolding for musical composition (FSMC). MIDI 1 is a computer-generated control, and MIDI 2 is the original Bad Girl’s Lament without accompaniment. Special thanks to Barry Taylor for granting special permission to utilize his own MIDI productions of folk music in this work. Barry Taylor originally sequenced both Nancy Whiskey and Bad Girl’s Lament without accompaniment.

Survey

1. Have you heard of NEAT Drummer or of Functional Scaffolding for Musical Composition?
   No
   Yes

2. Based on your impression, how likely is it that any of the instrumental parts in the musical piece found at the following link, MIDI 1, 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)
   1  2  3  4  5  6  7  8  9  10

3. Based on your impression, how likely is it that any of the instrumental parts in the musical piece found at the following link, MIDI 2, 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.
4. Based on your impression, how likely is it that any of the instrumental parts in the musical piece found at the following link, MIDI 3, 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)

1 2 3 4 5 6 7 8 9 10

5. Based on your impression, how likely is it that any of the instrumental parts in the musical piece found at the following link, MIDI 4, 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)

1 2 3 4 5 6 7 8 9 10

6. Based on your impression, how likely is it that any of the instrumental parts in the musical piece found at the following link, MIDI 5, 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)

1 2 3 4 5 6 7 8 9 10
A.2 Musical Turing Test Study in 4.3.2

The six questions below appear as survey participants saw them for the listener study described in Chapter 4 Section 3.2, Comparing FSMC-augmented Compositions to Fully Human Compositions. The accompaniment tracks in MIDIs 3, 4, and 5 were generated by FSMC. MIDI 1 is a computer-generated control that is intentionally poor and MIDI 2 is the original Bad Girl’s Lament without accompaniment. Special thanks to Barry Taylor for granting permission to utilize his own MIDI productions of folk music in this work. Barry Taylor originally sequenced both Bad Girl’s Lament and Scarborough Fair without accompaniment.

Survey

MIDI 1
MIDI 2
MIDI 3
MIDI 4
MIDI 5

1. Have you heard of NEAT Drummer or of Functional Scaffolding for Musical Composition?
   No
   Yes

2. Rate MIDI 1 on a scale of one to ten.
   (1 is the worst and 10 is the best)
   1 2 3 4 5 6 7 8 9 10
3. Rate MIDI 2 on a scale of one to ten.

   (1 is the worst and 10 is the best)

   1 2 3 4 5 6 7 8 9 10

4. Rate MIDI 3 on a scale of one to ten.

   (1 is the worst and 10 is the best)

   1 2 3 4 5 6 7 8 9 10

5. Rate MIDI 4 on a scale of one to ten.

   (1 is the worst and 10 is the best)

   1 2 3 4 5 6 7 8 9 10

6. Rate MIDI 5 on a scale of one to ten.

   (1 is the worst and 10 is the best)

   1 2 3 4 5 6 7 8 9 10

A.3 Perceived Quality Listener’s Study in 5.1.1.3

The four questions below appear as survey participants saw them for the listener study described Chapter 5, Section 1.1.3, Assessing the Human Contribution. Participants only saw one of five possible pairs of MP3s. One MP3 in the pair was generated with the assistance of FSMC while the other was generated by FSMC alone to serve as a control. The order participants saw the MP3s was randomized.
Survey

MP3 1

MP3 2

1. Have you heard of NEAT Drummer or of Functional Scaffolding for Musical Composition?
   No
   Yes

2. Rate your level of musical expertise.
   (1 is no musical experience and 10 is much higher than average experience with music)
   1  2  3  4  5  6  7  8  9  10

3. Rate MP3 1 on a scale of one to ten.
   (1 is the worst and 10 is the best)
   1  2  3  4  5  6  7  8  9  10

4. Rate MP3 2 on a scale of one to ten.
   (1 is the worst and 10 is the best)
   1  2  3  4  5  6  7  8  9  10
LIST OF REFERENCES


