MUME 2016 - The Fourth International Workshop on Musical Metacreation, ISBN #978-0-86491-397-5

Active Trading with Impro-Visor

Zachary Kondak, Mikayla Konst, Carli Lessard, David Siah, Robert M. Keller

Vassar College, University of Rochester, Harvey Mudd College, and Villanova University, USA

zakondak@vassar.edu, mkonst@u.rochester.edu, clessard@hmc.edu, dsiah@villanova.edu, keller@hmc.edu

Abstract

Trading is a common form of jazz improvisation in which one performer exchanges improvisations with others, usually in four- or eight-bar segments. We describe and demonstrate a new feature of Impro-Visor (short for *Improvisation Advisor*, a program designed to help musicians develop improvisational skills) called *active trading*, which significantly extends its former automated, but *passive*, grammar-based trading capabilities. Because Impro-Visor's active trading can be based on a variety of different response models, it can be viewed as a *meta* capability, providing for future extensions simply by plugging in code for other trading modules.

Introduction

Impro-Visor is a program designed to help musicians learn to improvise or to improve their improvisation proficiency. It is free, open-source software, with a user community of around 8,800 subscribers. The central focus of an Impro-Visor document is an editable lead sheet, which displays chord symbols and a monophonic melody. The original purpose was to have a user construct a solo by point-and-click on the lead sheet, providing various forms of advice and feedback on choices being made. An example of advice occurs in the form of a large repertoire of melodic fragments ("cells", "idioms", "licks", and "quotes") keyed to chords in the progression. An example of visual feedback occurs in the form of note coloration (Keller and Morrison, 2007), to indicate whether a given note is a chord tone, color tone (tone outside the chord but sonorous with it), approach tone (chromatic half-step away from a chord or color tone), or none of the above (typically an "outside" sounding note). These categories are provided in a vocabulary file, which the user is free to modify or augment. There are many editing tools, one of which is *rectifying* a melody so that it consists only of tones that are not outside the harmony.

Through several years of development, the authors (along with many other undergraduate students) have broadened and deepened the educational purposes by adding capabilities such as input from MIDI instruments to the lead sheet, including real-time input, automatic improvisation using grammars, off-line grammar and style learning (from transcribed solos and MIDI performances respectively), and harmonic analysis in the form of idiomatic harmonic "bricks" (Keller, *et al*. 2013). Performance aspects are done in the context of automatically generated backing (typically piano, bass, and percussion) for both improvisations and user input.

Until recently, the only form of trading available is what we now refer to as *passive* trading, where the production of melody by the grammar alternates with space left for user input (Keller, 2012). While passive trading is helpful for practice, it is not truly reactive, in that the system is not really listening to what the user played. *Active* trading is a new feature, in which the program can listen to the user and base its improvisations on what the user plays. Input takes the form of MIDI, such as can be created from a MIDI keyboard or other MIDI instrument. We also provide a companion converter called Pitch Tracker, written in the *SuperCollider* framework (SuperCollider), which converts monophonic audio to MIDI for input to the program.

Trading Modes

We have implemented a basic capability for inputting a MIDI stream in real-time while the program is also playing melody and accompaniment. While implementing these real-time functions in Java, the language in which the entire system is coded, presented significant technical challenges, in this paper we mainly address the issues of *how* the program might respond to the user's input and give examples of how it currently responds. Because the program is intended to provide various tutorial functions, one can conceive of several possible trading modes:

- The program responds with melodic lines directly reflective of what the user played.
- The program responds with melodic lines that are more complex than what the user played. For example, the responses could include shorter note values, a wider range of pitches, or more harmonic tension.
- The program responds with melodic lines that are simpler than what the user played. This could be useful to illustrate to the user how to create better-structured melodies.

Thus in the first mode, the program might be considered to be serving as a companion (Keller, *et al*., 2012), while in the second and third modes, the program serves as more of a tutor.

At present, we have not labeled the modes to indicate the relative tutorial functions described above, although this is a future objective. Instead the actual modes provided reflect technological approaches, as described next. Most of these modes were chosen because they exploit infrastructure already present in Impro-Visor for other purposes. However, we plan to explore other modes in the future.

Repeat and Rectify Mode

In repeat and rectify mode, the program simply rectifies what the user has played before responding. Rectification is a simple transformation that pulls melody notes into line with the underlying chord, by moving each note that is not a chord tone nor a color tone to the nearest such tone. Thus rectification guarantees that the response will sound "inside" the chord progression, unlike what might happen with repeat mode. A second use of this mode is to inform the user how the line might look and sound if user pitch errors, as perceived by the program, are corrected.

Modify and Rectify Mode

In modify and rectify mode, the program performs simple modifications based on compositional techniques, such as inversion, reversal, and time dilation or compression, then rectifies the result. The purpose of this mode is to provide modest, but recognizable uses of the melody in responding. In the future, other types of responses will be implemented.

Abstract Melody Mode

The concept of *abstract melody* is described in (Gillick, *et al*., 2012). There the concrete notes of a melody are replaced with indicators of their categories (chord tones, color tones, etc.) while retaining durations. These abstract notes are then instantiated to concrete notes based on the underlying harmony. An additional part of the abstract melody specification includes *slopes*, which are concatenated to form *contours* to produce lines shaped like the original, but not necessarily identical to it. The original use of abstract melody was in improvisations from learned grammars. A big advantage of an abstract melody is that it can be used over any chord progression, including ones substantially different from the original source. This trading mode entails converting the user's input to an abstract melody, then uses the abstract melody to create a new melody over the chord sequence for the response, which may differ from the chord sequence over which the user plays.

Transform Mode

Transform mode applies one of an open-ended set of transformations to the user's input. These transformations are specified by transformational grammars, as described in (Putman and Keller, 2015). Examples of transformations include interpolating one or more passing tones, adding turns, neighboring tones, and many others. Impro-Visor provides automation for learning these transformations from solo transcriptions, and there are several different transformational grammars derived from well-known artists available.

Grammar Solo Mode

This mode is closest to passive trading, in that a generative grammar is used to provide the response, but the user's input is not used. This mode is useful for having the user respond to the program, rather than the other way around. The user can select from many different generative grammars.

Chop and Memorize Mode

This mode memorizes selected segments the user's melody for reuse later in the session. Other transformations can be applied when the melody is reused.

Other Modes

Development of other modes is the subject of on-going research, as are methods for assessing quality of responses.

Other Controls

In addition to the modes as described in the preceding section, Impro-Visor's active trading provides the following controls:

- *Who starts*: Indicates whether the program or the user will play first in trading.
- *Length of trade*: The length of traded melodies can be set to any integral number of bars, from a single bar on up.
- *Looping*: In loop mode, the user can continue trading over repeated choruses of a tune as long as desired.

Examples

Figure 1 shows an example of 2-bar trading using Modify and Rectify Mode. The user played the first two bars in real-time (160 beats per minute) on a MIDI keyboard, and the system responded with the next two bars. Trading continued on, but we only show a small segment. The 2-bar pattern continues on the second line. Selection of the modifications is random. The first trade happens to use inversion, while the second uses an identity transformation but rectifies. Figure 2 shows the active trading control panel for this particular session.

Figure 3 shows a continuation of the above chord progression, a 12-bar blues doubled to 24 bars. This figure

illustrates 4-bar trading, using the Transform mode, with a transformational grammar learned from a Bill Evans solo.

Figure 4 shows 4-bar trading over a different tune, *Hot House* by Tadd Dameron using the grammar that was learned from a more complex Wes Montgomery solo. Compared to Figure 3, we see the kinds of complexity variations possible with different transformational grammars.

Related Work

Our work is independent of, but preceded by, several other programs that provide modes of interactive trading between a jazz musician and computer. Space limitations only permit us to list representative papers, relying on the reader to consult the references for further information. *The Continuator* (Pachet, 2002) is a performance vehicle that learns the style of a performing musician and then continues playing in that style, for example to complete or enhance an initial melody. *GenJam* (Biles, 2013) is a performance-oriented program that can trade with its human developer by producing responses based on genetic operators. As such, it contrasts with the grammar-based approaches used in our system.

Other work includes *Monterey Mirror* (Manaris, *et al*. 2015), which is also based on genetic operators and Zipf's law, and *CHIME* (Franklin, 2002), which used a recurrent neural network and reinforcement learning. *Clap-along* (Young and Bown, 2010) also mentions several modes of mirroring the user's rhythmic patterns.

We are in no sense claiming superiority of our approach over these. The rationale for our presentation here is the addition of active trading of a specific free and open-source notational tool having an appreciable pre-existing user community.

Conclusion and Initial Evaluation

We have reported on the status of on-going work that introduces active trading into Impro-Visor, a notation-based tool focused on individual education more than public performance. With active trading, a user can trade melodies with the program as it plays the background of chords, bass, and percussion. Both the user and computer parts are recorded in notation. As this feature was only made available one month before this publication, user experience is limited so far. However, the last author can attest that trading actively with Impro-Visor has revealed some insights about his own playing, such as a tendency to reuse certain ideas in playing over a given tune. The ideal effect of such a realization should result in a broadening of the set of ideas in the player's repertoire. The ability to record these ideas on the staff as they are played can provide a record and assessment of such broadening.

Acknowledgment

This work was supported in part by NSF CISE REU award #1359170 to Harvey Mudd College. We thank Alex Putman, who developed the original transformational grammar feature, on which the Transform trading mode is based. We also thank the referees for several helpful comments.

References

Biles, J.A. *Straight-Ahead Jazz with GenJam: A Quick Demonstration*. MUME (Musical Metacreation Workshop), pp. 20-23, 2013.

Franklin, J.A. *Multi-phase learning for jazz improvisation and interaction*. Proc. Eighth Biennial Symposium on Arts and Technology, Connecticut College, 2001.

Gillick, J., Tang, K., and Keller, R.M. 2010. *Machine Learning of Jazz Grammars*. Computer Music Journal, September 2010.

Impro-Visor. 2016. http://www.impro-visor.com/ (last consulted 10 May 2016).

Keller, R.M. and Morrison, D. *A Grammatical Approach to Automatic Improvisation*, Proceedings SMC'07, 4th Sound and Music Computing Conference, Lefkada, Greece, pp. 330-337, July 2007.

Keller, R.M., Toman-Yih, A. Schofield, A. and Merritt, Z. *A Creative Improvisational Companion Based on Idiomatic Harmonic Bricks*, Proc. Third International Conference on Computational Creativity, Dublin, pp. 155-159, 2012.

Keller, R.M. *Continuous Trading with Impro-Visor*, Proc. Third International Conference on Computational Creativity, Dublin, pp. 222, 2012.

Keller, R.M., Schofield, A., Toman-Yih, A., Merritt, Z., and Elliott, J. *Automating the Explanation of Jazz Chord Progressions Using Idiomatic Analysis*. Computer Music Journal, Vol. 37, No. 4, pp. 54-69, Winter, 2013.

Manaris, B., Hughes, D., and Vassilandonakis Y., *Monterey Mirror: an experiment in interactive music performance combining evolutionary computation and Zipf's law*. Evolutionary Intelligence 8(1), Springer Verlag, pp. 23-35, March, 2015.

Pachet, F. *Playing with virtual musicians: the Continuator in practice*. IEEE Trans. Multimedia, vol. 9, no. 3, pp. 77 82, 2002.

Putman, A.M. and Keller, R.M. *A Transformational Grammar Framework for Improvisation*. Proc. First International Conference on New Music Concepts, Treviso, Italy, pp. 39-45, March, 2015.

SuperCollider. http://supercollider.github.io/tutorials/ (last consulted 10 May 2016).

Young, M. and Bown, O. *Clap-along: A Negotiation Strategy for Creative Musical Interaction with Computational Systems*. Proc. First International Conference on Computational Creativity, Lisbon, pp. 215-222, 2010.

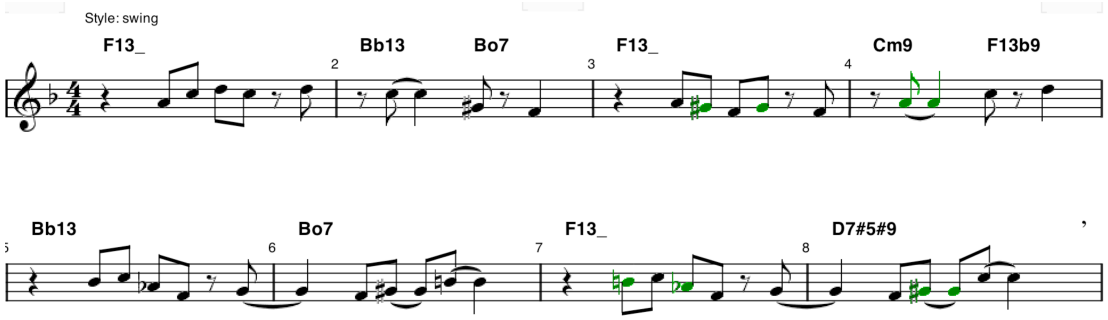

Figure 1. Screen capture of an active trading session with Impro-Visor over a 12-bar blues recorded in real-time (160 beats per minute). The first two bars capture what the user played, while the third and fourth bars are Impro-Visor's response, using Modify and Rectify mode. In the first line, the modification is a simple inversion. In the second line, which follows the same trading pattern, there is only a rectification. This happens, by random selection, to illustrate how rectification adjusts the melody notes to be either chord or color tones (shown with green note heads).

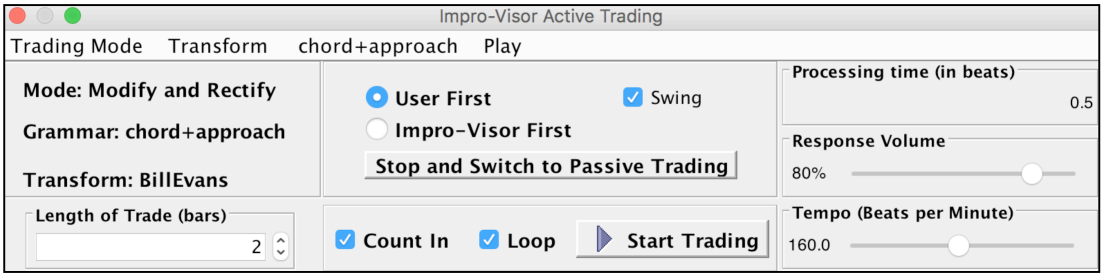

Figure 2. Active trading control panel, as set for the above trading

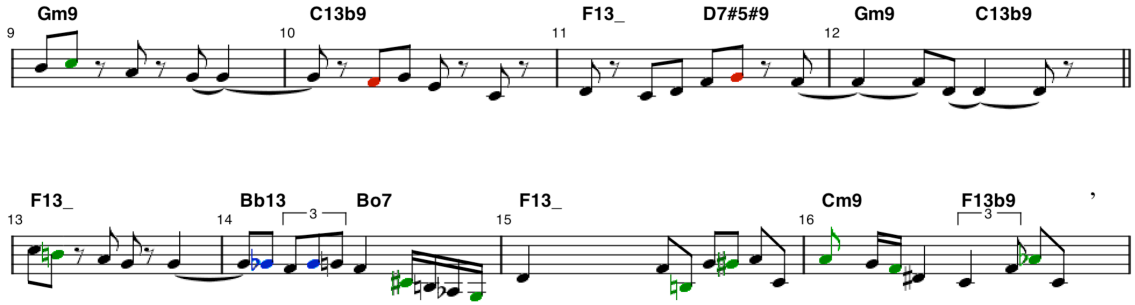

Figure 3. Trading four bars, beginning with the last four of a 12-bar blues as played by the user, followed by the first four of the blues as Impro-Visor's response. Here Transform Mode is used, with the Bill Evans transformational grammar. As can be seen, the response melody is considerably embellished compared with the user's input.

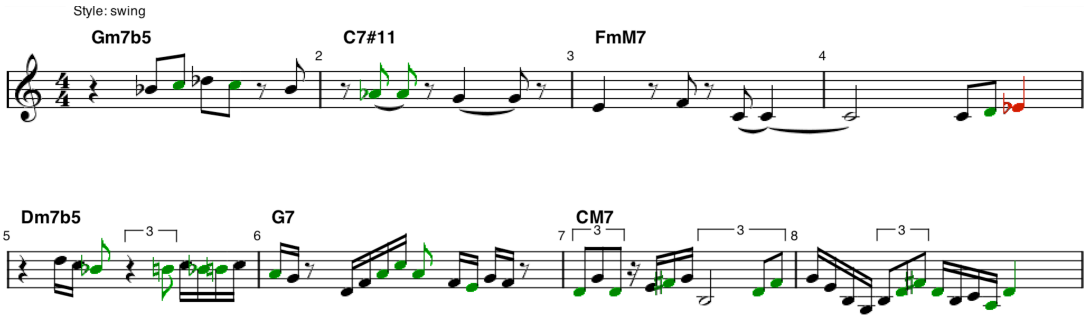

Figure 4. Trading four bars, over the tune Hot House by Tadd Dameron using Transform Mode with the West Montgomery transformational grammar, with the user starting. Note that this grammar provides considerably more complexity than the Bill Evans grammar in the preceding example, as this grammar was learned from a more complex solo.