On Modular Approach to AI-Aided Musical Creativity

AuthorA[1] and AuthorB[2]

[1] InstituteA
[2] InstituteB
icsc2022.music@gmail.com

Abstract. Recently the AI-aided musical creativity systems are under rapid development. The vast growth of machine learning (ML) techniques has already brought a majority of powerful solutions to the field of sound synthesis, generative techniques, music information retrieval, etc. Availability of computational resources leads to advancement of raw audio solutions. The decomposition of problems may decrease the complexity of a proposed solution achieving higher quality at the each of production stages.

Keywords: AI-aided creativity, modular systems, system design

1 Introduction

The AI-aided computer methods are under rapid development in the field of music generation, jamming, copyright law, orchestration, music discovery and playlist recommendations, remixing and production, music analysis, DJing, rehearsing, teaching and learning, and live performances. Analyzing the development of computer music techniques starting from the middle of the XX century implies that the proposed solutions often become a trend, due to the corresponding mainstream technology, rather than the domain specific metrics. Nowadays ML-based techniques are in the major focus. Despite of significant success of ML based approaches in the above fields, the following known problems exist:

- the large data sets and cloud computation farms are not available for the small research groups and end users,
- uncontrollable, black-box behavior of proposed tools,
- high economical and ecological costs,
- cultural bias in the data based approaches,
- hard to maintain.

We assume, that the modular architecture may overcome listed difficulties.

2 AI music modular applications

Some approaches using modular architectures for the computer aided music creativity already existed. In [1] authors present the general modular architecture for computer systems for the expressive music performance.

2.1 Functional modules

The requirements for the modular design can be like the following:

- modules should be domain specific, not technology specific, making it possible to change the underlying solution;
- modules should have common interface of communication and configuration.

Let's consider some examples of modules and proposed technologies for their implementation. Large group of modules are to perform analysis and information retrieval tasks. Harmony and bars tracking may be implemented based on [2] and [3] respectively.

With the help of Omnizart library we may utilize music transcription tasks. We able to transcribe the pitched instruments, vocal melody, chords, drum events, and beat from polyphonic audio. The corresponding methods are described in [4], [5] and [6]

Another group of modules is considered to solve the form detection tasks: chorus and verse detection, audio preview highlighting, etc. In [7], authors proposed an algorithm for detecting the most emotional part of the track, whats is critically necessary for audio preview generation. Chorus detection module may be implemented based on the [8] research. In [9], the comparison of chorus detection algorithms can be found.

Since music is not just a generic audio signal, but it typically translates some message to the listener, the perceptional and emotional analysis module is required and can be developed after [10]. Proposed solution provides not only audio, but lyrics based analysis.

Mixing and mastering modules are the final steps within the processing chain for the majority of the production scenarios. It seems reasonable to decompose them into smaller parts. Some AI-aided approaches to music mixing and mastering can be found in [11], [12], [13] and [14].

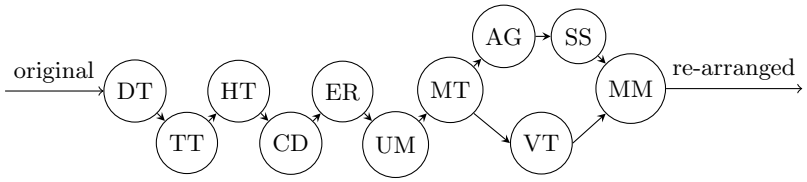
2.2 Complex music creativity systems architectures

Let's consider the complex "real-world" audio production problems are to be resolved using a combination of the proposed particular tasks solving modules. Fig. 1 shows an example of such modular systems.

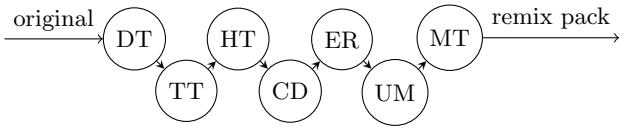
The first one is a In-Style Re-Arranger System. It produces a re-arranged version based on a raw audio track. The chain begins with the analysis modules: Downbeat Tracking (DT), Tempo Tracking (TT) and Harmony Tracking (HT). Then goes a Chorus Detection (CD) module, providing a highlighted part of the song for the further in-style composition. Emotional Ranking (ER) is used to get some human perceptual analysis. The selected part has to be unmixed, to extract the original vocals and arrangement, with the Un-Mix Module (UM) and then transformed to symbolic data (i.e. MIDI or other score system) with the Music Transcription (MT) module. Then, the obtained data controls the process of the new Arrangement Generation in a specified module (AG). Generated

arrangement is to be rendered in the Sound Synthesis (SS) module. Then, combined with the transformed vocals, all tracks go to the Mixing and Mastering (MM) module.

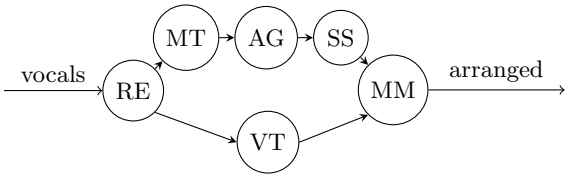
The second system is a Remix Helper System. It produces a remix sample pack, including audio tracks and MIDI, based on a source track. Its process chain begins with the same analysis modules: DT, TT and HT. Then comes chorus detection and emotion ranking modules to determine most essential parts and hooks. After that, the selected parts are processed by the UM module, to extract isolated vocals and drum loops. Also, there goes a transcription module to produce all necessary MIDI tracks.



In-Style Re-Arranger System



Remix Helper System



Arrangement Generation System

Fig. 1. Examples of complex modular systems architectures.

The last example is an Arrangement Generation System, based on a user recorded vocal part. The user is considered to record a vocal part within record module (RE) using metronome track to keep the tempo sync. One part of the system is aimed to translate the voice recording to MIDI in the MT module, then to propose a symbolic version of an arrangement in the AG module, based on a recorded melody. Then goes a sound synthesis module. In the parallel part the vocals are carefully tuned, aligned and processed in Vocal Transformation (VT) module. Finally, the track goes to the mixing and mastering modules.

The proposed systems can be easily upgraded, extended and can provide the reasonable user control on every stage.

3 Csound as Sound Synthesis Core

Csound language is naturally considered as a basis for a general purpose sound synthesis engine.

Csound has open API to provide control of all sound engine as well as particular instrument. Csound can be run inside Python, one of the most widespread AI technologies language.

Csound score events protocol of interaction is extendable for the particular synthesizer control needs. Score events may control some global synthesis parameters as well. Score events are not restricted to present equal temperament instruments notes, like MIDI note messages, so they are suitable to control untempered, continuous-pitch instruments.

Recently Csound supports the real time audio synthesis and processing, so the Csound-based sound synthesis module seems to be well suited for the live performance systems. With the development of real-time network communication technologies, like WebRTC, modular system may become distributed.

Also, Csound is well documented and continues its growth due to the community efforts.

4 Modular techno re-mixer

We provide the following example of techno style re-arranger based on the simplified system design of previously provided architecture. It was developed with Csound as a basis of sound synthesis module. We provide the results of techno style song preview rearrangements in MIDI format and final audio results[3].

Analysis modules were developed with librosa[4] to perform tempo tracking, omnizart[5] for harmony tracking and madmom[6] [15] for downbeat tracking. Chorus detection was implemented using pychorus[7] algorithm [16]. Open-unmix[8] library was used to separate vocals from the original song chorus. Pyrubberband[9] module was used to perform the original vocal time stretching. Music transcription module was implemented with specified version of omnizart provided algorithm. We developed a rule-based techno arrangement generation model. The Csound-based synthesis module consists of layered kick sounds, snare, two hat and crash instruments (all sample based), and FM-based synthesizers.

[3] https://github.com/ashekochikhin/modular_pub
[4] https://librosa.org/doc/latest/index.html
[5] https://music-and-culture-technology-lab.github.io/omnizart-doc/
[6] https://github.com/CPJKU/madmom
[7] https://github.com/vivjay30/pychorus
[8] https://github.com/sigsep/open-unmix-pytorch
[9] https://github.com/bmcfee/pyrubberband

5 Conclusions

Modular approach to the AI-aided music creativity is flexible, extensible and new technologies friendly. An independent development of different parts of the AI-aided systems brings a challenge to module realizations. We consider the following Csound development directions to bring the modular systems closer:

- With the development of interactive components of meta- and augmented reality, the generation and synthesis of sonic spaces will demand more computational resources, so the integration with DSP, CUDA or some other non-general purpose computational systems seems reasonable;
- Csound API extension, especially for the web and data science environments, seems to be in high priority.

References

1. Kirke, A., Miranda, E.R.: Performance Creativity in Computer Systems for Expressive Performance of Music. In: Miranda, E.R. (eds) Handbook of Artificial Intelligence for Music. Springer, Cham. (2021)
2. Chen, T.-P., Su, L.: Harmony transformer: incorporating chord segmentation into harmony recognition. In ISMIR. (2019)
3. Böck, S., Krebs F., Widmer G.: Joint Beat and Downbeat Tracking with Recurrent Neural Networks. Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR). (2016)
4. Hsu, J.-Y., Su, L.: VOCANO: A Note Transcription Framework For Singing Voice In Polyphonic Music. Proc. International Society of Music Information Retrieval Conference (ISMIR). (2021)
5. Wei I.-C., Wu C.-W., Su, L.: Improving Automatic Drum Transcription Using Large-Scale Audio-to-Midi Aligned Data. ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 246-250. (2021)
6. Wu, Y.-T., Chen, B., Su, L.: Multi-instrument automatic music transcription with self-attention-based instance segmentation. In TASLP. (2020)
7. Huang, Y.-S., Chou, S.-Y., Yang, Y.-H.: Pop Music Highlighter: Marking the Emotion Keypoints. Transactions of the International Society for Music Information Retrieval. (2018)
8. Wang, J.-C., Smith, J. B.L., Chen, J., Song, X., Wang, Y.: Supervised Chorus Detection for Popular Music Using Convolutional Neural Network and Multi-Task Learning . ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). (2021)
9. Nieto, O., Bello, J. P.: Systematic Exploration Of Computational Music Structure Research. Proc. of the 17th International Society for Music Information Retrieval Conference (ISMIR). New York City, NY, USA. (2016)
10. Pyrovolakis, K., Tzouveli, P., Stamou, G.: Mood detection analyzing lyrics and audio signal based on deep learning architectures. 2020 25th International Conference on Pattern Recognition (ICPR), pp. 9363-9370. (2021)
11. Martinez Ramirez, M., Reiss, J.: Deep Learning and Intelligent Audio Mixing. Proceedings of the 3rd Workshop on Intelligent Music Production, Salford, UK. (2017)

12. Moffat, D., Sandler, M.: Approaches in Intelligent Music Production. Arts. 8. 125. 10.3390/arts8040125. (2019)
13. Steinmetz, C. J., Pons, J., Pascual, S., Serra, J.: Automatic Multitrack Mixing With A Differentiable Mixing Console Of Neural Audio Effects . ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). (2021)
14. Birtchnell, T.: Listening without ears: Artificial intelligence in audio mastering. Big Data Society. (2018)
15. Böck, S., Korzeniowski, F., Schlüter, J., Krebs, F. Widmer, G.: madmom: A New Python Audio and Music Signal Processing Library. Extended abstracts for the Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conference. (2016)
16. Vivek Jayaram. Finding Choruses in Songs with Python, `https://towardsdatascience.com/finding-choruses-in-songs-with-python-a925165f94a8`