Symbolic representation and processing of musical structure: stream segments, pitch interval patterns, general chord types[1]

Emilios Cambouropoulos[1]

[1] Aristotle University of Thessaloniki, Greece
emilios@mus.auth.gr

Abstract. The difficulty of modelling musical structure in a general and cognitively plausible manner is due primarily to music's inter-dependent multi-parametric and multi-level nature that allows multiple structural interpretations to emerge. Traditional AI symbolic processing methods, however, can be used effectively for modelling particular analytic and creative aspects of musical structure. In this paper three specific problems of music structure, namely, segmentation and streaming, pattern extraction, harmonic abstraction and generation, will be addressed with a view to highlighting the importance of problem definitions, music representation and multi-parametric hierarchical cognitively-inspired processing methodologies. Existing proof-of-concept models are used as a basis for a theoretical discussion.

Keywords: Symbolic AI, segmentation, streaming, pattern matching, harmony, chord representation

1 Introduction

Understanding music means being able to make sense of musical structure. Musical structure does not simply contribute to musical meaning but is at the heart of musical meaning as basic musical concepts are in essence concepts relating to musical structure. Listeners are capable of discerning, encoding and remembering diverse aspects of musical structure when exposed to musical stimuli, such as scales, keys, tonal centers, motives, themes, metre, rhythmic patterns, harmonic progressions, cadences.

Through the centuries, music theorists, analysts, philosophers have attempted to describe and formalise, core musical concepts and processes. More recently, computational methodology (assisted by research in music cognition, linguistics, logic reasoning, neuroscience and so on), has offered new means of precision and formalisation, enabling the development of sophisticated representations and models of musical structure. Progress in this domain, however, has been much slower than expected and

[1] M. Rauterberg (Ed.): HCII 2021, LNCS 12794, pp. 1–12, 2021.
The final authenticated version is available online at https://doi.org/10.1007/978-3-030-77411-0_14

researchers are still striving for general powerful theories that can describe the complex and multifaceted nature of musical structure.

In this paper, first, we will briefly discuss why it is more difficult, than one initially believes, to model musical structure in a general and comprehensive manner. Current computational methodologies will be mentioned, and it will be maintained that traditional AI symbolic processing methods are still relevant and advantageous in certain respects, especially when drawing on general cognitive principles of human perception.

Then, three specific problems of music structural modelling, namely, segmentation and streaming, pattern extraction, harmonic representation, learning and generation, will be addressed. The aim is to present rather 'unconventional' definitions of the problems themselves, as well as the proposed representations and methodologies. Emphasis is given to common underlying fundamental mechanisms and interconnections that apply to seemingly disparate domains. All these suggestions draw on previous proof-of-concept models by the author that require further development and investigation.

2 Symbolic processing and general cognitive principles

Perhaps, the main reason musical structure is difficult to model is its inter-dependent multi-parametric and multi-level nature that allows multiple structural features to emerge. Moreover, there exist diverse musical styles and idioms each with their own representational and processing schemes. For instance, in a musical surface for western music (in this paper we assume a piano-roll-like encoding), pitch, rhythm and harmony are basic structural features that shape the music and interact with each other on multiple levels of abstraction. Studying, for instance, patterns solely in the pitch domain, one soon discovers that absolute pitch is probably too elementary (not transposition invariant), but, then, to deal with relative pitch, a tonal center is required and emergent tonal hierarchies play a significant role depending on the structure of keys, which emerge within specific metrical and grouping structures influenced by hierarchic structures of harmony. And, soon one realizes that a fully-fleshed theory of musical structure is required to deal with 'mere' pitch patterns.

Let us briefly examine another example, namely the extraction of the musical surface itself from audio (transcription) which in the minds of many is essentially a bottom-up process. It is now accepted that higher-level aspects of musical hierarchical organisation are necessary to transcribe music in some form of symbolic notation ([3], [29]); a purely bottom-up approach to score extraction has been shown to be untenable. Apart from multipitch analysis and instrument recognition, broader information is necessary such as beat tracking, rhythmic organisation, chord recognition and harmonic analysis, along with notational conventions, and, even, high-level musical structure analysis and knowledge of expressive performance. For instance, a human transcriber can fill in gaps in the audio signal (due to recording defects or noise) based on knowledge of structure (e.g., even though one or more tones may be absent from the audio input, they can be recovered due to knowledge of key or motivic structure or harmonic expectation, and so on). In essence, a comprehensive computational theory of musical structure is required for full blown score transcription. Abstraction, categorisation, hierarchic organisation and prior knowledge are all at full play in the task of music transcription. A seemingly 'simple' task is complex and significant progress in this domain will occur when high-level music analysis models mature.

The complex multi-parametric and multi-level nature of the structure of diverse musical styles renders efforts to built hand-crafted rule-based models impractical and often unworkable. For this reason, deep learning techniques have drawn the attention of researchers in recent years (e.g., [5], [13]). Often such methods are considered as the obvious, if not the only, way to deal effectively with modelling musical tasks. Deep neural networks present the important ability to abstract knowledge on higher-levels of representation, based on sample data; they are flexible, adaptive, easy to build as they do not require fully fleshed-out models, and they are resilient to noise or incomplete information. So, why bother follow a symbolic rule-based approach that requires manual coding, does not allow dynamic change and it cannot capture the complexity of the real world?

Traditional symbolic AI modelling enables the development of music models that may have both theoretical and practical advantages. In terms of theory, our understanding of music *per se* is enriched, traditional assumptions are tested, empirically-derived cognitive principles evaluated and new musical knowledge is acquired. As knowledge is explicit in such AI models, sophisticated practical systems can be created that allow intelligent interaction with musicians / users though the manipulation of meaningful symbolic representations (e.g., educational systems, compositional assistants, interactive performers, content-based music search engines, and so on). Such systems make use of prior knowledge acquired through years (or even centuries) of experience and introspection, and, also, capitalize on findings resulting from empirical work in music cognition. This way sophisticated models can be built relatively quickly combining diverse components on different hierarchical levels of organisation. Additionally, symbolic systems reinforced with simple statistical learning capacities, can adapt to different contexts based on relatively small training datasets allowing this way a certain degree of flexibility. Furthermore, such models can bridge different conceptual spaces enabling the invention of novel concepts not present in the initial input spaces.

A debate on the pros and cons of traditional symbolic AI methods vs deep neural network learning techniques can be found in studies such as [21], [23], (see also [6] for a defense of the symbolic AI approach in music modelling). Recently, attempts are made to combine the strengths of both approaches reconciling symbolic systems, that are strong in abstraction and inference, with deep learning techniques that excel in perceptual classification [10].

Our mind continuously groups sounds together based on their similarity and by trying to find simple ecological patterns that can describe them ([2], [4]). The fundamental principles of perception, first studied by the Gestalt psychologists [19], give an account of the basic rules that account for such grouping. A common notion underlying many of these principles is similarity (which is directly linked to change). For instance, in music, many models that attempt to break the musical continuum into smaller constituent parts (e.g., segments, voices) have relied on principles such as pitch similarity, temporal proximity, parallel motion, i.e., similarity in the pitch, time and pitch interval domains respectively. Similarity is also at work on higher levels of cognition whereby learnt patterns (e.g. a fugue theme) can be recognised in a rather complex musical continuum. More generally, learning techniques are based on finding regular patterns in data, and commonly a distance function (similarity measure) is used somewhere in the learning process (training or classification or clustering stage).

Symbolic models in computational musicology that strive for generality (i.e., applicability in a broad spectrum of musics) often rely on general cognitive principles such as similarity / change. Additionally, specific acoustic or auditory principles are employed as constraints that help narrow down the usually large search spaces. Such auditory-specific constraints rely on aspects of sound perception that have to do with properties of sound sources, the auditory system and typical sound environments listeners are exposed to (e.g., harmonicity, octave equivalence, dissonance, masking, onset simultaneity thresholds). The use of such perceptual principles, accompanied by probabilities of features and patterns (learned from data) that reflect regularities and tendencies of specific musical environments, can give rise to rather sophisticated musical systems.

In the next three sections, three different musical problems will be presented examined from relatively unusual angles, redefining the problems themselves or suggesting novel solutions so as to be more general and idiom independent. The presentation below is mostly theoretical; it is, however, grounded on earlier proof-of-concept implementations by the author. It is suggested that further research in the proposed line of inquiry may produce new more flexible and adaptive models of musical structure.

3 Stream Segments

Voice or stream separation algorithms attempt to model computationally the segregation of polyphonic music into separate voices [12]; music segmentation algorithms on the other hand, segment music voices/streams into smaller coherent groups [27]. Both segmentation and streaming rely on fundamental Gestalt principles such as temporal and pitch proximity. In principle, firstly, voices / streams are determined and then voices / streams are segmented into smaller groups. Is it possible to develop a model that separates notes vertically into streams and, at the same time, locates segment boundaries? In other words, is it possible to parse a general two dimensional pitch-time space into *stream segments*? The main advantage of adopting the concept of stream segments is that they are meaningful in any type of music, not only when music has a rather ‘fixed’ number of independent voices (e.g., fugues) - see stream segment illustration in **Fig. 1**.

An algorithm that makes use of a single set of auditory principles for the concurrent horizontal and vertical segregation of a musical texture into stream segments has been proposed by [26]. This algorithm groups together in the same stream segment notes that have proximal onsets (synchronous onsets for quantised data), similar durations (same for symbolic data) and similar pitch interval direction (parallel/similar motion), and, additionally, successive (non-overlapping) notes that are temporally proximal and similar in pitch; non-synchronous overlapping notes belong to different stream segments along with successive (non-overlapping notes) that are temporally distant and dissimilar in pitch. This prototype algorithm was tested against a small manually-annotated dataset of musical excerpts, and preliminary results were encouraging. An example of stream segmentation parsing (coherent groups of notes such as melodic segments, harmonic accompanimental fragments, homophonic passages) is presented in **Fig. 1** – problems and shortcomings of the algorithm are discussed in [26].

Models that can detect stream segments can be very useful as they enable the organisation of the low-level musical surface into coherent groups that are musically meaningful; such organisation facilitates more efficient and higher-level analytic processing. For instance, in searching for instances of the pitch pattern descending-perfect-fifth - followed-by-unison [-P5, unison] in the example of **Fig. 1**, a general polyphonic pattern identification algorithm would correctly detect the melodic instances in mm. 10, 11, & 18 but would also (incorrectly in perceptual terms) detect this pattern in (at least) the homophonic textures of mm. 12 and 14; being able to separate melodic from homophonic / accompanimental textures may contribute to more accurate and efficient search. This line of research into stream segments does not seem to have been taken on by the MIR research community (a possible problem is the lack of annotated ground truth data against which to test algorithms). However, it is herein maintained that it is a worthwhile research project in the direction of building a more general model for breaking down the musical surface into perceptually meaningful subgroups.

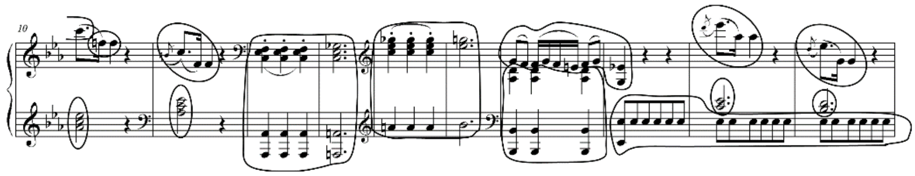

Fig. 1. Stream segments detected by algorithm in the opening of Beethoven's Sonata Op.31, No.3 (Fig.4, [26])

4 Pitch interval patterns

The capacity of listeners to 'match' varied musical materials is essential to the process of identifying meaningful musical entities such as interesting motifs, themes, melodic and rhythmic patterns, characteristic harmonic progressions, and other memorable musical entities. In recent years, a number of computational systems have been developed that describe symbolic melodic similarity (see overviews in [7], [30]). Such algorithms address different perspectives of this multi-faceted similarity task, such as representation, scope, similarity function, polyphony and so on. For instance, most algorithms are applied on monophonic strings of symbols, whereas few employ geometric models on two-dimensional point-set representations. The latter are more powerful in the sense that they can identify melodic patterns directly in unprocessed polyphonic music, at the expense, however, of retrieving higher numbers of false positives. It is known that listeners cannot identify patterns across auditory streams [4]; in this sense, it is more practical to segregate a musical surface into distinct voices / streams, and then to apply string matching algorithms (that are computationally simpler and more efficient) on (melodic) strings of symbols identifying patterns that are more likely to be musically interesting and cognitively plausible.

Dynamic programming techniques, often based on various types of edit distance, are commonly used to find approximate matches in melodic strings. Edit distance is a very useful technique commonly applied to strings of pitches [22]. Techniques, however, using standard edit distance operations (replacement, insertion, deletion, along with consolidation and fragmentation) applied on strings of notes have limitations and inherent shortcomings such as defining a similarity threshold (any sequence can match with any sequence if enough edit operations are applied) and lack of transposition invariance. If edit distance is applied to strings of pitch intervals, problems occur, such as the fact that the insertion or deletion or replacement of a single interval changes drastically the rest of the pitch sequence – see, however, proposal by [20].

In [1] the problem of matching is redefined in a way that is appropriate for strings of melodic intervals (not notes). Matching can be applied directly to strings of intervals (in semitones) without any preprocessing (as is required in [20]). To this aim, the replacement, insertion and deletion operations are abolished, and only consolidation and fragmentation operations are retained, adapted to the interval domain. Two or more intervals of one string may be matched to an interval from a second string through consolidation (i.e., the sum of one or more intervals of the first string should be equal to an interval of the second string) – this is the many-to-one matching problem; in a similar fashion, fragmentation is defined, i.e., one-to-many interval matching. The general case is many-to-many interval matching (the sum of two or more consecutive intervals from the first string is equal to the sum of two or more intervals of the second string) – see example in **Fig. 2**. Working with intervals means melodic matching is transposition-invariant. Additionally, matching is confined by equality in the consolidation and fragmentation operations (the only threshold necessary in the pitch domain is the maximum number of intervals allowed in consolidation / fragmentation).

melody 1	5	-1	1 -3	2 -4
melody 2	2 2 1	2 -3	-2	-2
reduction	5	-1	-2	-2

Fig. 2. The two melodic segments can be matched via the proposed pitch interval consolidation-fragmentation operations as seen in the first two rows of the table where the sum of the intervals in corresponding cells is equal. The melodic reduction pattern can be matched to each of the melodies via fragmentation (each cell in the last row of the table is equal to the sum of two or more intervals in each corresponding cell of the two melodies).

The implementation of interval matching via fragmentation / consolidation presented in [1], allows only one-to-many and many-to-one matches; the algorithm was tested only on one piece by W. A. Mozart (Sonata in A major KV331) searching for reduced versions of the theme (one-to-many problem); preliminary results were very encouraging.

Listeners are capable of discerning common elements between varied musical material primarily through hierarchic reduction, i.e., identifying ‘essential’ common characteristics, such as interval patterns between disjunct salient notes. The one-to-many implementation of the proposed algorithm allows such patterns to be retrieved. An example (handmade) is presented in **Fig. 3** to illustrate the kinds of themes that may be retrieved by such an algorithm from a theme database (such as the database in themefinder.org) given a reduced pitch sequence query. A variant of this algorithm that additionally takes into account rhythmic durations is presented in [11, 10], tested on four classical theme-and-variation pieces.

Fig. 3. Example of incipits of themes that may be matched to a given pitch interval query (top) employing the proposed melodic matching via interval fragmentation methodology. Each interval of the query can be fragmented up to (this this case) four intervals; the intervals matched to the query are designated by horizontal brackets.

It is suggested that this new definition of the problem of melodic matching, requires no preprocessing and is reliable in capturing hierarchically related pitch patterns (i.e., underlying salient pitches) that are transpositionally invariant. Rethinking representation and hierarchic structure issues is sometimes useful to overcome problems and shortcomings of more standard approaches.

5 Chords and Harmony

Representing chords and harmony is the last area to be discussed in this paper. Multiple notes occurring simultaneously are grouped by listeners into a ‘chord’, which is an entity in its own right, carrying functions, expectations, meaning, even emotions. Note simultaneities are often perceived as ‘wholes’ prior to establishing finer elaborations such as individual constituent pitches, octave information, note doubling, note omission, chord inversion, roots, and so on (for instance, chords in different positions are essentially equivalent as shown by [14]). The notion of the root of a chord is attributed to a note (often missing from the simultaneity that constitutes the chord) depending on psychoacoustic phenomena and tonality hierarchies (see perceptual root calculation model by [25]).

How can the infinite variety of possible simultaneities (in terms of octave position, note doubling, note omissions, note extensions, inversions) be reduced to a cognitively manageable number of abstract chord types/families? How can note verticalities be represented? Additionally, can we represent chords in different ways (using a common cognitively-inspired mechanism) depending on different qualities of diverse harmonic systems? The standard encoding of chords for tonal music is appropriate for tonal music, but not for other non-tonal idioms; pc-set encodings, on the other hand, are useful for atonal music but have weaker explicatory power for tonal music. Is an adaptive representation possible?

The General Chord Type (GCT) representation ([8], [9]), allows the re-arrangement of the notes of a harmonic verticality such that abstract idiom-specific types of chords may be derived. Given a consonance-dissonance classification of intervals (that reflects sensory and/or culturally-dependent notions of consonance / dissonance), the GCT algorithm finds the maximal subset of notes of a given note simultaneity that contains only consonant intervals; this maximal subset forms the base upon which the chord type is built and the lowest note of the base is the root of the chord. This encoding is inspired by the standard roman numeral chord type labeling, but is more general and flexible (it can encode, for instance atonal normal order pc-set types). Currently the GCT is revised to account for a multi-valued ranking of dissonance that enables the disambiguation of certain ambiguities that appear in the original version that is based on a binary dissonance vector.

In the example of **Fig. 4**, a standard roman numeral analysis is presented along with the GCT encoding for a tonal context. The GCT analysis is given for every vertical slice of the excerpt. For instance, [0, [0,4,7]] represents a tonic major chord in the C major key, whereas [7, [0,4,7]] a dominant chord. The second beat of the second measure comprises of two vertical slices, both of which have the same chord type base [2, [0, 4, 7]]; the two chords can be merged into a single more abstract chord type (in this case a secondary dominant to the dominant). The last beat of the second measure and the first beat of the third measure correspond to two vertical slices each; choosing one of the two (following the underlying harmonic rhythm) is possible if prior knowledge regarding the Bach chorale idiom is employed (e.g., acquired via corpus-based learning), such as chord typicality (e.g., a minor dominant chord [7,[0,3,7]] is very rare) or chord progression typicality (e.g., IV→vii° more common than vi^7→vii°). The GCT representation can be used, not only to encode any note simultaneity (in tonal or atonal or other contexts) but additionally to determine broader more abstract families of chords based on similarity and/or functionality (see [17]). It is suggested that such chord relations can be employed in the context of automated harmonic analysis, enabling not only the encoding of chords but the reduction of musical surfaces to underlying harmonic progressions; this can be done in both tonal and non-tonal musics, as the GCT can be adapted to different harmonic idioms.

Fig. 4. Roman numeral analysis and GCT encoding of the opening of J. S. Bach's, Chorale 40 (Ach Gott und Herr) BWV255.

Representing and processing harmonic structure involves developing sophisticated hierarchical representations (e.g., [28]). A simple approach for composing melodic harmonisations in relation to the GCT scheme was presented by [18], where chords are labelled employing the GCT representation, and corpus-based learning (from annotated harmonic reductions) involves learning chord transitions at the lowest chord-to-chord level and at the level of phrase boundaries (cadences). In the context of a generation (harmonisation) framework, constraints are inserted at phrase boundaries ensuring appropriate cadential schemata at structurally important positions, and, then, intermediate chord progressions are filled in according to the learned chord transition matrices. This method is incorporated in the *Chameleon* melodic harmonisation assistant ([16], [17]) that is adaptive (learns from data), general (can cope with any tonal or non-tonal harmonic idiom) and modular (learns and encodes explicitly different components of harmonic structure: chord types, chord transitions, cadences, bass line voice-leading).

The harmonic knowledge acquired by this system, can be used creatively in a cognitively-inspired conceptual blending model that allows the creation of combinational components between disjoint spaces, with very little (if any) training and with transparent access to what concepts are combined. The *Chameleon* melodic harmonisation assistant is essentially a proof-of-concept creative model that demonstrates that new harmonic concepts can be invented that transcend the initial harmonic input spaces. It is argued that such original creativity is more naturally accommodated in the world of symbolic reasoning that allows links and inferences between diverse concepts at high abstract levels [6, 15]. Moreover, symbolic representation and processing facilitates interpretability and explanation that are key components of musical knowledge advancement. Overall, a symbolic hierarchical modular representation coupled with basic statistical learning of harmony, not only, gives rise to a rather sophisticated description of harmonic structure but, additionally, allows generation of new harmonisations in certain styles and, even, production of more adventurous creative cross-idiom harmonisations.

6 Conclusions

In this paper, three areas of music modelling, namely, segmentation and streaming, pattern extraction, harmonic abstraction, learning and generation have been examined in terms of fundamental principles of perceptual hierarchic organisation that can form the basis for general computational systems of musical structure. Emphasis has been given to approaching these problems from somewhat ‘unconventional’ viewpoints that give rise to relatively new definitions, representations and methods. Common underlying fundamental mechanisms and interdependencies that apply to seemingly irreconcilable areas have been highlighted. It is maintained that cognitively-inspired computational models of musical structure should take into account psychoacoustic / perceptual constraints, fundamental cognitive principles, logical principles, and should strive for generality and parsimony. Traditional AI symbolic representations and methodologies (despite a number of drawbacks discussed above) allow building sophisticated models relatively quickly, combining diverse components on different hierarchical levels of organisation. As knowledge is explicit in such AI models, sophisticated practical systems can be created that allow intelligent interaction with musicians / users though the manipulation of meaningful symbolic representations. At the same time, such systems can be used for testing various hypotheses and acquiring new insights into our understanding of music.

References

1. Barton, C., Cambouropoulos, E., Iliopoulos, C.S., and Lipták, Z.: Melodic String Matching Via Interval Consolidation and Fragmentation. *IFIP Advances in Information and Communication Technology,* Vol. 382, pp. 460-469. Springer Publications. (2012).

2. Bendixen, A., Bőhm, T. M., Szalárdy, O., Mill, R., Denham, S. L., & Winkler, I.: Different roles of similarity and predictability in auditory stream segregation. *Learning & Perception*, *5*(Supplement-2), 37-54. (2013).
3. Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., & Klapuri, A.: Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, *41*(3), 407-434 (2013).
4. Bregman, A. S.: *Auditory scene analysis: The perceptual organization of sound*. MIT press. (1994).
5. Briot, J. P., Hadjeres, G., & Pachet, F.: *Deep learning techniques for music generation* (pp. 1-249). Springer (2020).
6. Cambouropoulos, E. and Kaliakatsos-Papakostas, M.: Cognitive Musicology and Artificial Intelligence: Harmonic Analysis, Learning and Generation. In *Handbook for Artificial Intelligence in Music*. E.R. Miranda (Ed). Springer. (forthcoming).
7. Cambouropoulos, E. and Kaliakatsos-Papakostas, M.: Symbolic Approaches and Methods for Analyzing Similarity, In Daniel Shanahan, John Ashley Burgoyne, and Ian Quinn (Eds.) *The Oxford Handbook of Music and Corpus Studies,* Oxford University Press. (forthcoming).
8. Cambouropoulos E.: The Harmonic Musical Surface and Two Novel Chord Representation Schemes. *Computational Music Analysis,* David Meredith (ed.), ch.2, pp.31-56, Springer, (2015).
9. Cambouropoulos E., Kaliakatsos-Papakostas M. and Tsougras C.: An Idiom-independent Representation of Chords for Computational Music Analysis and Generation. In *Proceedings of the Joint 11th Sound and Music Computing Conference (SMC) and 40th International Computer Music Conference (ICMC),* Athens, Greece. (2014).
10. Garnelo, M., & Shanahan, M.: Reconciling deep learning with symbolic artificial intelligence: representing objects and relations. *Current Opinion in Behavioral Sciences*, *29*, 17-23. (2019).
11. Giraud, M. Déguernel, K., Cambouropoulos, E.: Fragmentations with Pitch, Rhythm and Parallelism Constraints for Variation Matching. *International Symposium on Computer Music Multidisciplinary Research (CMMR 2013), LNCS 8905*, pp.298–312 (2014).
12. Gray, P., & Bunescu, R.: From Note-Level to Chord-Level Neural Network Models for Voice Separation in Symbolic Music. *arXiv preprint arXiv:2011.03028*. (2020).
13. Herremans, D., & Chuan, C. H.: The emergence of deep learning: new opportunities for music and audio technologies. *Neural Computing and Applications,* vol. 32, pp. 913–914 (2020).
14. Hubbard, T. L. and Datteri, D. L.: Recognizing the Component Tones of a Major Chord. *American Journal of Psychology*, 114(4): 569-589 (2001).
15. Kaliakatsos-Papakostas, M. and Cambouropoulos, E.: Conceptual Blending of High-Level Features and Data-Driven Salience Computation in Melodic Generation. *Cognitive Systems Research*. Volume 58, pp. 55-70 (2019).
16. Kaliakatsos-Papakostas, M., Queiroz, M., Tsougras, C., & Cambouropoulos, E.: Conceptual blending of harmonic spaces for creating melodic harmonisation, *Journal of New Music Research*, 46(4), 305-328 (2017).
17. Kaliakatsos-Papakostas M., Zacharakis A., Tsougras C. and Cambouropoulos E.: Evaluating the General Chord Type Representation in Tonal Music and Organising GCT Chord Labels in Functional Chord Categories, *Proceedings of the 16th International Society for Music Information Retrieval (ISMIR) Conference*, Malaga, Spain (2015).
18. Kaliakatsos-Papakostas M. and Cambouropoulos E.: Probabilistic harmonisation with fixed intermediate chord constraints. In *Proceeding of the Joint 11th Sound and Music Computing*

Conference (SMC) and 40th International Computer Music Conference (ICMC), Athens, Greece (2014).
19. Köhler, W.: *Gestalt Psychology*. Liveright, New York (1947)
20. Lemström, K., and Ukkonen, E.: Including interval encoding into edit distance based music comparison and retrieval. *Proceedings of the AISB'00 Convention* (Artificial Intelligence and Simulation of Behaviour), (2000).
21. Marcus, G.: Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631* (2018).
22. Mongeau, M., & Sankoff, D.: Comparison of musical sequences. *Computers and the Humanities*, *24*(3), 161-175 (1990).
23. Mill, R. W., Bőhm, T. M., Bendixen, A., Winkler, I., & Denham, S. L.: Modelling the emergence and dynamics of perceptual organisation in auditory streaming. *PLoS Comput Biol*, *9*(3), e1002925. (2013).
24. Nilsson, Nils J.: The Physical Symbol System Hypothesis: Status and Prospects. *SpringerLink*, 2007, 9–17 (2017).
25. Parncutt, R.: A model of the perceptual root(s) of a chord accounting for voicing and prevailing tonality. In Marc Leman (ed.) *Music, Gestalt, and Computing: Studies in Cognitive and Systematic Musicology.* Springer, Berlin (1997).
26. Rafailidis, D., Nanopoulos, A., Cambouropoulos, E. & Manolopoulos, Y.: Detection of Stream Segments in Symbolic Musical Data. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR08),* Philadelfia, Pennsylvania (2008).
27. Rodríguez-López, Marcelo, & Volk, A.: Symbolic segmentation: A corpus-based analysis of melodic phrases. In *International Symposium on Computer Music Multidisciplinary Research*, pp. 548-557. Springer (2013).
28. Rohrmeier, M.: Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, *5*(1), 35-53 (2011).
29. Ryynänen M. P., & Klapuri A. P.: Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3), 72-86 (2008).
30. Velardo, V., Vallati, M., & Jan, S.: Symbolic melodic similarity: State of the art and future challenges. *Computer Music Journal*, *40*(2), 70-83 (2016).