MUSIC INFORMATION PROCESSING
a new application for operations research
by Elaine Chew

The goals of the article is twofold: first, to introduce the topic of computer modeling of music information processing by humans and the potential impact of Operations Research methods in various aspects of this endeavor; and second, to introduce the Spiral Array model, a mathematical model for tonality, and to summarize some recent work on algorithms for problems in music perception and cognition based on the Spiral Array representation.

1. INTRODUCTION

The burgeoning of digital music media has allowed and necessitated the use of computational tools to analyze, process and manage music information. Every musical composition is the result of a series of decisions through a complex decision space. Each performance involves interpretation, the design of inter-related decisions constrained by requirements for economy of movement and smooth transitions. Music information is the result of human decisions within a highly structured decision space, and its analysis is the science of deciphering the operations performed in the musical system. Operations research methodology is ideal for processing this kind of information.

Any computer model for processing music information must begin with a representation that converts content information to numerical data for analysis. Music consists of streams of organized sounds with a limited number of attributes, such as pitch (frequency) and duration. The inter-pitch relations among sets of pitches result in higher level structures such as chords and keys that operate in a system called *tonality*. The symmetry and structure in tonal music is exploited in the Spiral Array representation to allow for the design of efficient algorithms to analyze, categorize and compare music. The Spiral Array model was proposed in [5] as a mathematical model for tonality. It is a geometric model that is configured so that closely related musical entities are positioned in compact clusters. Influenced by interior point approaches, the mapping from musical to 3D space transforms streams of music data to trajectories inside the spiral structure.

Two of the fundamental components to music information processing are extraction of pitch (frequency) and time (duration) structures [8]. Figure 1 shows the interrelation among the components of these structures in the context of music content extraction and analysis. Using the Spiral Array model, many problems in recognizing and determining pitch and time structures can be defined as optimization problems using Euclidean metrics. Tonal induction (or key-finding) and pitch spelling can be re-cast as nearest-neighbor searches [2,4]; determining key changes and chord recognition becomes a distance minimizing segmentation problem in the spiral space [3]. The time structures such as beat, tempo and meter govern and are influenced by and affect the pitch structures. The beat, tempo and meter can, in turn, be deduced from the pitch structure information. An overview of these methods will be given in Section 3.

The tangible products of the methodological research include engineering decision systems that can:

- transcribe and analyze music information in real-time;
- perform content-based music categorization and retrieval;
- extract patterns and structures for 3D visualization; and,
- recognize and generate meaningful musical expression.

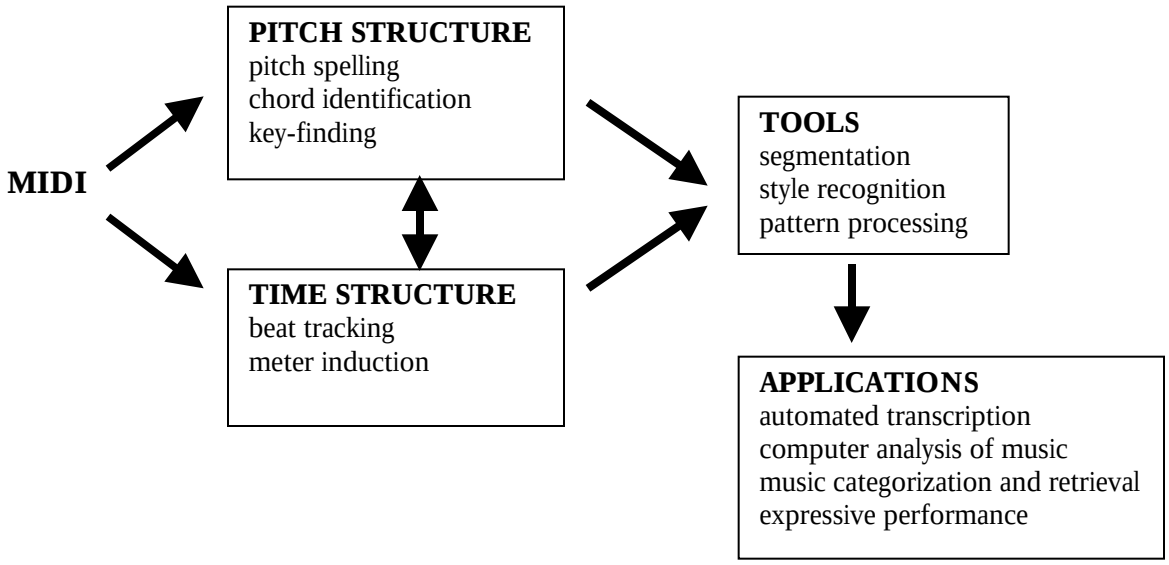


Figure 1. Music Information Content Extraction and Analysis

2. THE SPIRAL ARRAY MODEL

The mathematical approaches described in this article use a spatial representation for tonality called the Spiral Array [5]. The Spiral Array is derived from the Harmonic Network [10], an arrangement of pitch classes on a lattice that has been attributed to Leonhard Euler. The Spiral Array transforms the original 2D lattice of pitch classes into a 3D configuration that eliminates periodicity. In the model, higher level musical entities are represented in the interior of a spiral array and are successively defined as convex combinations of their lower level parts. Chords representations are created as convex combinations of the pitch positions; and, keys are generated from their defining chords in a similar manner. This interior point approach defines higher-level musical entities as coordinates in the same space as the pitch classes, mixing both discrete and continuous space.

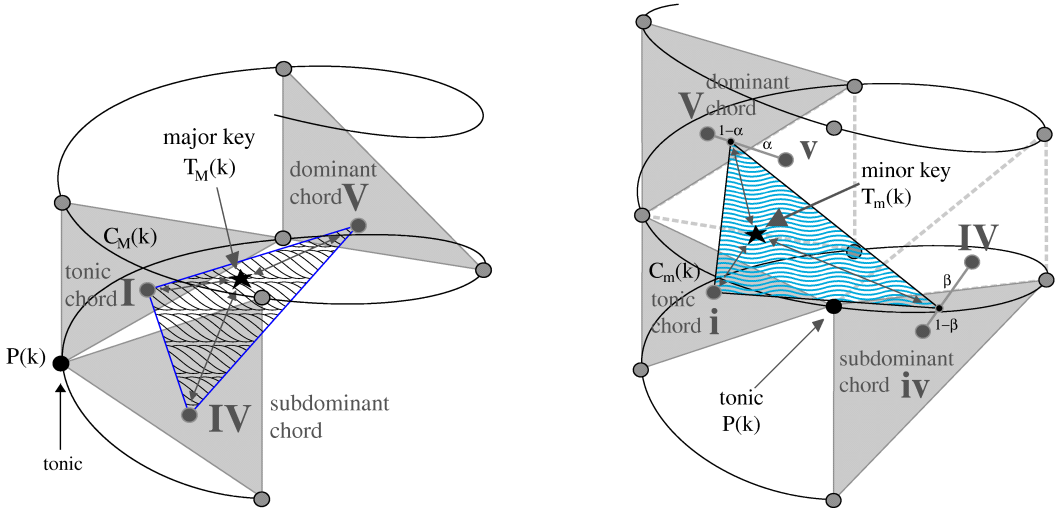
In the Spiral Array, pitch classes are represented by spatial coordinates along a spiral. Pitches are represented at each quarter turn of an ascending spiral, and neighboring pitches are five scale steps (a perfect fifth) apart, which results in vertically aligned pitches being three major scale steps (a major third) apart. Each pitch is indexed by its number of perfect fifths from an arbitrarily chosen reference pitch, C (set at position [0,1,0]):

Definition: The radius of the cylinder, r, and the height gain per quarter turn, h, uniquely define the position of a pitch representation, which can be described as: $\mathbf{P}(k) = [\ \sin(k\pi/2), \cos(k\pi/2), kh\ ]$.

Strict enharmonic equivalence (that is to say, if F# = Gb) would require that the spiral be turned into a toroid. The spiral form is assumed to preserve the symmetry in the distances among pitch entities. Major and minor chords are represented as convex combinations of their component pitch positions, and the weights on the three pitches are constrained to be monotonically non-decreasing according to the importance of that pitch.

Definition: The representation for a major triad is generated by the convex combination of its root, fifth and third pitch positions: $\mathbf{C}_M(k) = w_1\mathbf{P}(k) + w_2\mathbf{P}(k+1) + w_3\mathbf{P}(k+4)$, where $w_1 > w_2 > w_3$ and $w_1 + w_2 + w_3 = 1$. The minor triad is generated by a similar combination, $\mathbf{C}_m(k) = u_1\mathbf{P}(k) + u_2\mathbf{P}(k+1) + u_3\mathbf{P}(k-3)$, where $u_1 > u_2 > u_3$ and $u_1 + u_2 + u_3 = 1$.

Major and minor key representations are generated as convex combinations of their defining chords. Again, the weights are constrained to be monotonically non-decreasing according to the importance of the chord. Figure 2(a) shows an example of a major key representation. The minor key definition is slightly more involved because it uses both major and minor versions of the chords indexed by (k–1) and (k+1). The additional parameters α and β indicate the importance of the major vs. minor versions of the chords. Figure 2(b) shows the derivation of the minor key representation.



(a) Geometric representation of a major key (b) Geometric representation of a minor key

Figure 2: The Spiral Array Model

Definition: A major key is represented by a convex combination of its tonic, dominant and subdominant chords. The weights are restricted to be monotonically decreasing. $\mathbf{T}_M(k) = \omega_1\mathbf{C}_M(k) + \omega_2\mathbf{C}_M(k+1) + \omega_3\mathbf{C}_M(k-1)$, where $\omega_1 > \omega_2 > \omega_3$. The minor key representation is generated by a convex combination of its tonic, dominant (major or minor) and subdominant (major or minor) chords as follows: $\mathbf{T}_m(k) = \upsilon_1 C_m(k) + \upsilon_2[\alpha\mathbf{C}_M(k+1) + (1-\alpha)\mathbf{C}_m(k+1)] + \upsilon_3[\beta\mathbf{C}_m(k-1) + (1-\beta)\mathbf{C}_m(k-1)]$, where $\upsilon_1 > \upsilon_2 > \upsilon_3$, $\upsilon_1 + \upsilon_2 + \upsilon_3 = 1$ and $0 \le \alpha \le 1$, $0 \le \beta \le 1$.

Spatial distances in the model are required to parallel perceived relations among the represented entities. The selection of the weights and parameters are based on distance constraints generated by music knowledge. Feasible weights and parameters are currently found by decomposing the problem into stages that are solved by a combination of analytical and numerical means [5]. More work needs to be done to find better ways to derive feasible solutions from this nonlinear system for a variety of constraints.

In the Spiral Array model, any collection of notes (also known as pitch events) generates a *center of effect* (c.e.), a spatial point summarizing its attributes. For a sequence of music time series data, the c.e. is a point in the interior of the Spiral Array that is the convex combination of the pitch positions weighted by their respective durations. This technique maps any sequence of notes to trajectories in the interior of the Spiral Array.

Definition: If $\mathbf{p}_i$ represents the position of the i-th pitch class in the Spiral Array, and d_i represents its duration, the sequence of notes in a melody up to the t-th pitch event can be written as $(\mathbf{p}_i , d_i)$: i=1…t. The center of effect of this pitch collection is defined as: $\mathbf{c}_t = \Sigma_i d_i.\mathbf{p}_i / D_t$, where $D_t = \Sigma_i d_i$.

3. APPLICATIONS

The Spiral Array representation has been found to be particularly effective in the computational modeling of problems in music perception and cognition. A few applications are outlined below.

3.1 The Center of Effect Generator: A Key-Finding Algorithm

The key of a piece of music reveals the primary set of note material used in the piece. It serves as a point of reference for every musical event in the piece, and key-finding is a crucial component of music analysis systems. The Center of Effect Generator (CEG) algorithm for key finding maps music data to trajectories in the interior of the Spiral Array that gravitate quickly toward the spatial point representing its key context [4]. Figure 3 illustrates how the Shaker melody, *Simple Gifts*, maps to its spatial trajectory in the Spiral Array.

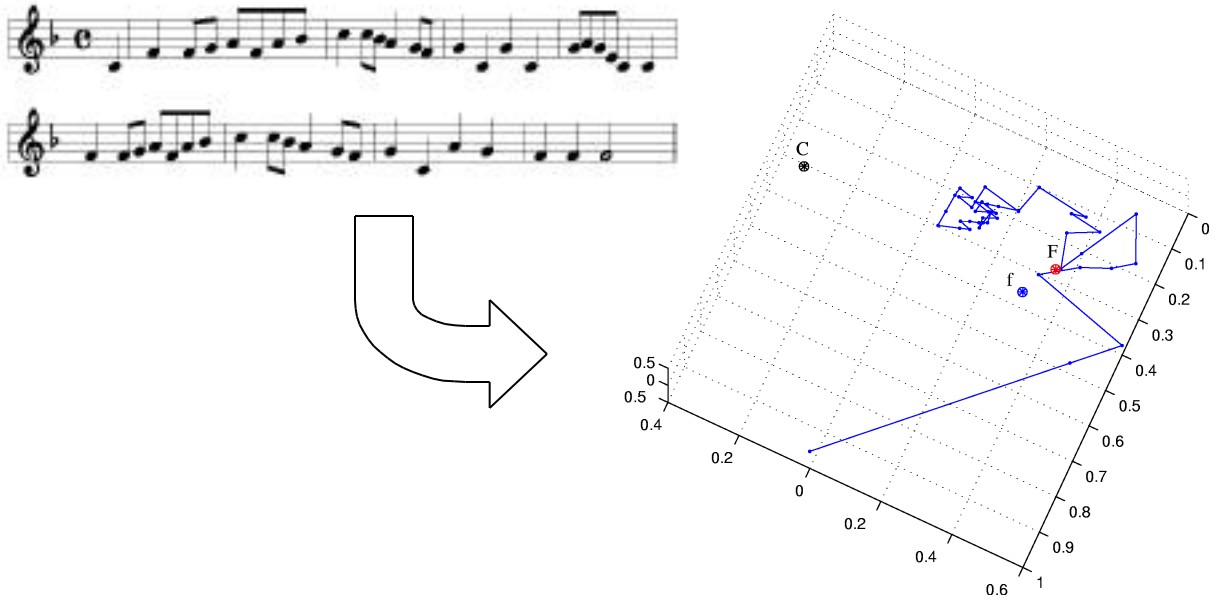


Figure 3: *Simple Gifts* and its trajectory in the Spiral Array.

Pitch names and durations were given as input to the program. At each time step, a c.e. is calculated based on the cumulative pitch and duration information, and the closest key is found through a nearest neighbor search. If **T** is a finite collection of keys represented in the Spiral Array, both major and minor, that is to say: $\mathbf{T} = \{ \mathbf{T}_m(k) \quad k \} \quad \{ \mathbf{T}_M(k) \quad k \}$, then the most likely key of the passage is given by: $\text{argmin}_{\tau \in \mathbf{T}} | \mathbf{c} - \tau |$. An indicator of the relative likelihood that the pitch collection is in any given key is given by its proximity to that key. The minimizing distance, also a likelihood indicator, is given by: $d^{min} = \min_{\tau \in \mathbf{T}} | \mathbf{c} - \tau |$.

The CEG algorithm was shown to be faster and more accurate in identifying the most likely key than existing methods. It was implemented and tested on Bach's fugue subjects in the *Well-Tempered Clavier*, and compared to Krumhansl & Schmuckler's Probe Tone Profile Method [6] and Longuet-Higgins & Steedman's Shape Matching Algorithm [7]. For the fugue subjects in Book I of the WTC, the CEG required on average 3.75 pitch events, the PTPM 5.25, and the SMA 8.71 to determine the correct key [4].

3.2 Disambiguating Pitch Spellings

The problem of assigning appropriate pitch spellings to digital music information is endemic to any music analysis or recognition system. In a MIDI file or other digital music formats, a pitch is represented by a numerical value indicating its frequency and not its letter name, and each pitch can correspond to more than one letter name. The name of the pitch determines the notation and the key context.

In the Spiral Array model, points in space labeled by pitch names from the same key form a compact cluster. In [2], we proposed a nearest neighbor approach to finding the best spelling for each pitch in the piece using the evolving c.e. as a proxy for the key context. Each plausible pitch spelling is measured against the current c.e., and the pitch that satisfies the nearest neighbor criteria is selected to be the appropriate pitch name. More concretely, each pitch read from the MIDI file corresponds to two or three most probable letter names indexed by the triplet: <index-12, index, index+12>. If $\mathbf{c}_t$ represents the center of effect of the pitch events up to time t, then for any given pitch, the index that is consistent with the key context is given by: index* = argmin { | $\mathbf{P}$(index(i)–12) – $\mathbf{c}_t$ |, | $\mathbf{P}$(index(i)) – $\mathbf{c}_t$ |, | $\mathbf{P}$(index(i)+12) – $\mathbf{c}_t$ | }. Initialization procedures are suggested for $\mathbf{c}_0$. When music is segmented into chunks for analysis, the pitch names can be assigned in batches.

A version of the algorithm has been implemented in *MuSA* – a music visualization software based on the Spiral Array Model and tested on two selected movements from Beethoven Sonatas (Op.79 and Op.109) [2]. The Spiral Array offers an effective alternative to other pitch spelling approaches such as, Temperley's preference rule approach [11] and Cambouropoulos' interval optimization method [1]. At present, the algorithm is being improved, and more benchmark data is needed to generate reliable metrics for comparison.

3.3 The Boundary Search Algorithm: Determining Key Boundaries

When two or more key areas exist in a piece, the key areas are delineated by modulation boundaries. Being able to find the modulations is an important part of music segmentation for analysis. The idea behind the Boundary Search Algorithm (BSA) is a simple one. The c.e., described in Section 2, collapses each set of pitch and duration information down to a spatial point. The better the pitch collection adheres to the pitch classes of the key in the appropriate proportions, the closer the c.e. to the key representation. Thus, the problem of finding the point(s) of modulation can be posed as a problem of determining the optimal placement of boundaries so as to minimize the distances between the respective c.e.'s and their closest keys.

Suppose that m boundaries have been chosen, given by $(B_1,\ldots,B_m)$ that divide the passage into m+1 time epochs. Let time 0, the beginning of the piece, be B_0, and the end of the passage be $B_{m+1} = L$. The note material between any two boundaries (B_i, B_{i+1}) generates a c.e. given by $\mathbf{c}_{(Bi,Bi+1)}$. The most likely key has a distance $d^{min}(B_i,B_{i+1})$ from this c.e. Our hypothesis is that the best boundary candidates will segment the piece in such a way as to minimize the sum of the respective distances $d^{min}(B_i,B_{i+1})$, i=0,…,m. The optimal boundaries, $(B_1^*,\ldots, B_m^*)$, are given by the solution to the following system: min Σ_i $d^{min}(B_i,B_{i+1})$ s.t. $d^{min}(B_i,B_{i+1}) = \min_{\mathbf{T}} | \mathbf{c}_{(Bi,Bi+1)} - T |$ and $B_i < B_{i+1}$, i=0,…,m. In addition, musical common sense constrains adjacent key areas to be distinct from each other, that is to say, $T_i \neq T_{i+1}$ (i=1…m). If the passage to be analyzed is a complete piece, the first and last key areas should be the same, that is to say, $T_1 = T_{m+1}$.

Assuming that the set of keys that are of interest **T** is finite, and that any piece of music is of finite length with n event onsets, the computational complexity of this approach is $O(n^m)$. The practical application of this method typically yields far better performance than that implied by its $O(n^m)$ computational time. The number of key changes and associated boundaries is usually small in relation to the number of event onsets ($m<<n$). In smaller-scale compositions, the pieces can often be segmented into three key areas – there is time only to modulate away from and return to the original key. BSA was demonstrated to perform well on two Minuets from Bach's *A Little Notebook for Anna Magdalena* [3]. Larger scale numerical tests need to be

performed to determine the average performance of the BSA algorithm. Future developments should either determine the number of boundaries automatically or detect the boundaries in real-time.

4. CONCLUSION

The above examples offer a sampling of computational issues in music information processing. Many other aspects of music involve optimization principles such as economy of movement and smoothness of transitions. For example, parsimonious voice leading is a composition requirement that has been noted by composer Jean-Philippe Rameau (1683-1764), and more recently, composer Daniel Schell used step-minimizing techniques to generate *beau karos*, geometric patterns that correspond to pleasing chord progressions in his composition *The Travelling Musician*, commissioned by the Association of European Operational Research Societies [9].

The growing interest in computing tools for music information processing is evidenced by the recent surge in the number of annual and biennial conferences devoted to various aspects of music information processing, such as, the International Conference on Music and Artificial Intelligence (*www.music.ed.ac.uk/icmai*), the International Symposium on Music Information Retrieval (*ismir2002.ircam.fr*) and the Performance Rendering Piano Concours (Rencon: *shouchan.ei.tuat.ac.jp/~rencon*), in addition to the more established International Computer Music Conference (*www.icmc2002.org*).

Music information processing is an ideal domain for modeling and exploring human decision processes. Operations research, the science of decision-making, has amassed a library of highly developed mathematical tools for modeling and analyzing such processes. As purveyors of these decision tools, operations researchers are poised to make significant contributions to the rapidly growing field of music information processing.

Acknowledgements

This article follows a talk given at the Turin Polytechnic in June of this year titled, "A Computational Framework for Problems in Music Perception and Cognition." I thank Marco Diana for the invitation to present the talk at the Department of Hydraulic, Transportation and Civil Infrastructures, and Federico Della Croce for the opportunity to present this work on music information processing in AIROnews.

References

[1] E. Cambouropoulos, Automatic Pitch Spelling: From Numbers to Sharps and Flats. In Proceedings of the VIII Brazilian Symposium on Computer Music (2001).
[2] E. Chew, E. & Y.-C. Chen, Mapping MIDI to the Spiral Array: Disambiguating Pitch Spellings. In a special issue on Optimization and Computation in the Network Era. Kluwer, NY (2003).
[3] E. Chew, The Spiral Array: An Algorithm for Determining Key Boundaries. In a special issue of the LNCS /LNAI on Music and Artifical Intelligence. Springer, Heidelberg, Germany (2002).
[4] E. Chew, Modeling Tonality: Applications to Music Cognition. In Proceedings of the 23rd Annual Meeting of the Cognitive Science Society (2001).
[5] E. Chew, Towards a Mathematical Model of Tonality. PhD Dissertation, MIT (2000).
[6] C. L. Krumhansl, Cognitive Foundations of Musical Pitch. Oxford University Press, NY (1990).
[7] H. C. Longuet-Higgins and M. J. Steedman, On Interpreting Bach. B. Meltzer and D. Michie (eds.) Machine Intelligence **6**. Edinburgh U. Press (1971) 221.
[8] R. Robert, Machine Musicianship. MIT Press, MA (2001).
[9] D. Schell, Daniel: Optimality in musical melodies and harmonic progressions: the travelling musician. European Journal of OR, special issue on real-world applications presented at EURO 2000.
[10] M. Steedman, The well-tempered computer. Phil. Trans. R. Soc. Lond. **349** (1994) 115—131.
[11] D. Temperley, The Cognition of Basic Musical Structures. MIT Press, MA (2002).