Listening to "Naima": An Automated Structural Analysis of Music from Recorded Audio

Roger B. Dannenberg

School of Computer Science, Carnegie Mellon University
email: dannenberg@cs.cmu.edu

1.1 Abstract

A model of music listening has been automated. A program takes digital audio as input, for example from a compact disc, and outputs an explanation of the music in terms of repeated sections and the implied structure. For example, when the program constructs an analysis of John Coltrane's "Naima," it generates a description that relates to the AABA form and notices that the initial AA is omitted the second time. The algorithms are presented and results with two other input songs are also described. This work suggests that music listening is based on the detection of relationships and that relatively simple analyses can successfully recover interesting musical structure.

2 Introduction

When we listen to a piece of music, we pay attention to repetition, and we use repetition or the lack of it to understand the structure of the music. This in turn helps us to anticipate what will come next, remember what we have heard, relate the music to other music, and explain or develop simple models of what we are hearing. Any structural relationship that we perceive, not just repetition, functions in the same way. Although not used in the present study, pitch transposition is an especially salient relationship.

In my opinion, this is the essence of listening to music. We hear relationships of all kinds among different sounds. We develop theories or models that predict what relationships are important and recurrent. Sometimes we are right, and sometimes that is interesting. Sometimes we are wrong, and that can be even more interesting. These ideas are not at all new (Simon & Sumner, 1968), but it is good to repeat them here in a way that emphasizes how simple the whole music listening and music understanding process might be, at least at some level.

Starting with this completely simple-minded view of what music and listening are about, my question is, can this conception of music understanding be modeled and automated? In particular, I am interested in recovering structure and information from actual audio – not symbolic notation, not synthesized examples, but recorded audio as found on a CD.

This project was motivated by music information retrieval problems. Music information retrieval based on databases of audio requires a significant amount of meta-data about the content. Some earlier work on stylistic classification indicated that simple, low-level acoustic features are useful, but not sufficient to determine music style, tonality, rhythm, structure, etc. It seems worthwhile to reexamine music analysis as a listening process and see what can be automated. A good description of music can be used to identify the chorus (useful for music browsing), to locate modulations, to suggest boundaries where solos might begin and end, and for many other retrieval tasks. In addition to music information retrieval, music listening is a key component in the construction of interactive music systems and compositions. (Rowe, 1993) The techniques described here show promise for all of these tasks.

While thinking about the problems of automated listening, given the well-known problems of polyphonic transcription, I happened to hear a recording of a jazz ballad played in the distance. After recognizing the tune, it occurred to me that the signal-to-noise ratio of this setting was so bad that I could hardly hear anything but the saxophone, yet the structure of the music was strikingly clear. I wondered, "Could a computer derive the same structure from this same signal?" and "If so, could this serve as a model for music understanding?"

3 Related Work

Many other researchers have considered the importance of patterns and repetition in music. David Cope's work explores pattern processing to analyze music, generally with the goal of finding commonalities among different compositions. (Cope, 1996) This work is based on symbolic music representations and is aimed at the composition rather than the listening process. Eugene Narmour has published a large body of work on cognitive models for music listening. In one recent publication, Narmour (2000) explores structural relationships and analogies that give rise to listeners' expectations. Narmour quotes Schenker as saying that "repetition … is the basis of music as an art." The more elaborate rules developed by Narmour are complex examples of structural relationships described here.

Published as: Roger B. Dannenberg (2002). "Listening to 'Naima': An Automated Structural Analysis of Music from Recorded Audio." In *Proceedings of the International Computer Music Conference.* San Francisco: International Computer Music Association.

Simon and Sumner (Simon & Sumner, 1968) developed a model of music listening and music memory in which music is coded as simply as possible using operators such as repeat and transpose. Compact encodings convey structural relationships within a composition, so my work is consistent with theirs, and is certainly inspired by it. Other researchers have noticed that data compression relies upon the discovery and encoding of structure, and so data compression techniques have been applied to music as a form of analysis. An application to music generation is seen in work by Lartillot, Dubnov, Assayag, and Bejerano (2001).

Mont-Reynaud and Goldstein (1985) investigated the discovery of rhythmic patterns to locate possible transcription errors. Colin Meek created a program to search for common musical sequences, and his program has been used to identify musical themes. (Meek & Birmingham, 2001) Conklin and Anagnostopoulou (2001) describe a technique for finding recurrent patterns in music, using an expectation estimation to determine which recurring patterns are significant. This analysis relies on exact matches. Another approach to pattern extraction is found in Rolland and Ganascia (2000). Stammen and Pennycook used melodic similarity measures to identify melodic fragments in jazz improvisations. (Stammen & Pennycook, 1993)

The nature of music listening and music analysis has been a topic of study for many years. A full review is beyond the scope of this paper, but this list may highlight the variety of efforts in this area.

4 Overview

The recording initially examined in this work is "Naima," composed by John Coltrane (1960) and recorded by his quartet. As an aside, a danger of this work is that after repeated exposure, the researcher is bound to have any recording firmly "stuck" in his or her head, so the choice of material should be made carefully! "Naima" is basically an AABA form, where the A section is only 4 measures, and B is 8 measures. There are interesting and clear rhythmic motives, transpositional relationships, and harmonic structures as well, making this an ideal test case for analysis.

The analysis takes place in several stages. First, the melody is extracted. This is complicated by the fact that the piece is performed by a jazz quartet, but the task is simplified by the clear, sustained, close-miked, and generally high-amplitude saxophone lines. Second, the pitch estimates are transcribed into discrete pitches and durations, using pitch confidence level and amplitude cues to aid in segmentation. Third, the transcribed melodic sequence is analyzed for embedded similarities using a matrix representation to be described. A simple, recursive melodic similarity algorithm was developed to be tolerant of transcription errors. Fourth, the similarity matrix is reduced by removing redundant information, leaving the most significant similarities. Fifth, a clustering algorithm is used to find groups of similar melodic material. For example, we would hope to find a cluster representing the three A's in the AABA structure. Sixth, while interesting, the clusters reflect many more relationships than a human would typically describe. A final pass works left-to-right (in time order) to find an "explanation" for the piece as a whole.

The following sections describe this analysis in detail. The results of each stage are described, leading to a final analysis.

5 Melody Extraction

After making a digital copy of the entire recording from CD, it was observed that the left channel contains a much stronger saxophone signal. This channel was down-sampled to 22.05kHz and saved as a mono sound file for analysis. The manual steps here could easily be automated by looking for the channel with the strongest signal or by analyzing both channels separately and taking the one with the best results.

Pitch is estimated using autocorrelation to find candidate pitches, and a peak-picking algorithm to decide the best estimate: Evaluate windows of 256 samples every 0.02s. Perform an autocorrelation on the window. Searching from left to right (highest frequency to lowest), first look for a significant dip in the autocorrelation to avoid false peaks that occur very close to zero. Then search for the first peak that is within 90% of the highest peak. Sometimes there is a candidate at double this frequency that looks almost as good, so additional rules give preference to strong peaks at higher frequencies. Details are available as code from the author; however, the enhanced autocorrelation method (Tolonen & Karjalainen, 2000), unknown to us when this work was started, would probably give better results. Furthermore, there are *much* more sophisticated approaches for dealing with pitch extraction of melody from polyphonic sources. (Goto, 2001)

Figure 1 illustrates the waveform and an associated pitch contour.

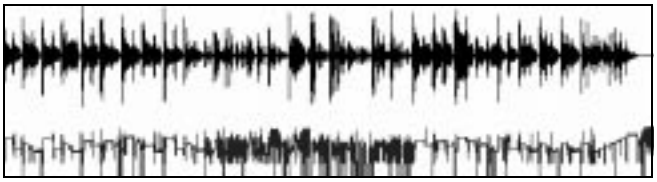

Figure 1. "Naima" left channel from CD recording amplitude (top) and pitch contour (bottom). The lines that descend to the bottom represent locations where no pitch was detected, reported and plotted as zero values. The middle of the piece is a piano solo where very little pitch information was recovered. An ascending scale is clearly visible at the end of the piece.

6 Transcription and Segmentation

The next step is to create a list of discrete notes. RMS amplitude information is derived from the original signal by removing frequencies below 200 Hz, computing the RMS over non-overlapping, square windows of duration 0.01s. The transcription works by looking for consecutive, consistent pitch estimates. We step one pitch estimate at a time, but look at overlapping groups of 15 estimates to help deal with noise and error. At each step, a group of 15 pitches is retrieved (corresponding to a time interval of 0.3s). Pitch estimates where the RMS value is below a threshold are considered unreliable, so they are forced to an erroneous value of zero. The pitches are then sorted. If 2/3 of the data falls in a range of 25 cents, then the pitch is deemed to be stable, marking the beginning of a note. Consecutive samples are processed similarly to find the extent of the note: if 5 of 15 estimates differ by less than 25 cents, and the median of these is within 70 cents of the start of the note, then we extend the note with the median of the new group of estimates. When the end of the note is reached, we report the pitch as the median of all the pitch estimates up to the first 1s of duration. This helps to ignore pitch deviations sometimes encountered near the beginnings of notes.

To avoid problems with absolute tuning differences, all pitches are kept as floating point numbers giving fractional semitones. To transcribe the data, a histogram is contructed from the fractional parts of all note pitches, quantized to 10 cents. The peak in this histogram indicates the difference between the tuning reference used in the recording and the A440 reference used in our analysis. This will also compensate for any systematic error or rounding in our admittedly low-resolution pitch estimation procedure.

Figure 2 illustrates the note transcription as a plot of pitch vs. time. The transcription does not include any metrical information.

7 Finding Similarities

The next step begins the interesting task of looking for structure in the data obtained so far. A melodic similarity matrix, $M_{i,j}$ is defined as the duration of similar melodic sequences starting at all pairs of notes indexed by i and j. We will assume a sequence of n notes is described by pitch p_i and duration d_i, $0 \le i < n$. If pitches do not match, then M is zero: $p_i \neq p_j \rightarrow M_{i,j} = 0$, so much of M is zero. Non-zero entries indicate similarity. For example, the second 4 measures repeat the first 4, starting at the 7th note. $M_{0,6}$ = 14.62 (seconds), the duration of the matching 4-measure repetition.

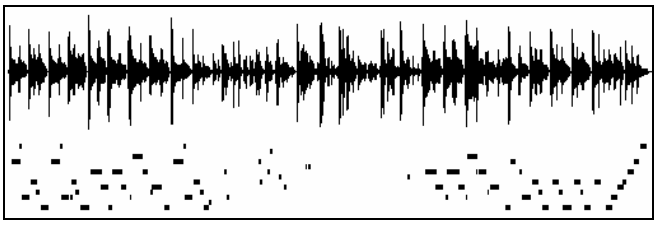

Figure 2. Transcription of "Naima." The saxophone at roughly the first and last thirds of the piece is transcribed fairly well, with only some short notes missing. The piano solo in the middle third is almost completely missed.

7.1 Melodic Similarity

A simple algorithm is used for determining the duration of matching melodic sequences, inspired by Mongeau and Sankoff (Mongeau & Sankoff, 1990). The two sequences to be compared are processed iteratively: if some initial part of sequence 1 matches some initial part of sequence 2, the initial parts are discarded and the remainders are compared in the next iteration. Matches occur when:

- the pitches of two notes match and either their durations or inter-onset-intervals (IOI) agree within 20% or 0.1s.
- a match occurs as just described after skipping one note of either melody, or one short note (< 0.6s) in each melody.
- two notes of either or both melodies have matching pitches, and when merged together, lead to matching durations (within 20%).

This algorithm is applied to every pair of non-equal note positions. Since the matrix is symmetric, we store the length of the matching sequence starting at i as $M_{i,j}$ and the length of the matching sequence starting at j as $M_{j,i}$.

7.2 Simplification

If location i,j represents similar sequences, then i+1,j+1 will probably represent similar, but shorter, sequences, the same sequences starting at i,j, excepting the first notes. Since this is uninteresting, or at least less significant than i,j, we want to remove these entries from M. The algorithm is simple: determine the submatrix $M_{i:u,j:v}$ that corresponds the matching sequences at i and j, i.e., the sequence at i runs to note u, and the sequence at j runs to note v. Set every entry in the submatrix to zero except for $M_{i,j}$. This simplification is performed on half the matrix, and the other half is zeroed symmetrically about the diagonal.

In addition, we are not interested in matching sequences that contain only one note, so these are also zeroed.

8 Clustering

After simplifying the matrix, we have all pairs of similar melodic sequences. What if a sequence is repeated more than once? If we scan any row or column, all non-zero entries represent the beginnings of similar sequences. Why?

Because each entry denotes a similarity to the sequence starting at the given row or column. We can use this fact to construct clusters of similar sequences. A cluster will be a group of melodic sequences that are all similar to one another.

The algorithm scans each row of *M*. At the first non-zero element, we note the duration, construct an empty cluster, and insert the corresponding pair of sequences into the cluster. Continuing to scan the row, as each non-zero element is found, and if the duration roughly matches the first one (within 40%), we insert the sequence (corresponding to the current column) into the cluster. If the duration does not match, the element is ignored. The cluster is complete when the end of the row is reached. To keep track of what sequences have been inserted into clusters, we zero all combinations; for example, if the cluster has sequences starting at *i, j, k,* then we zero locations *i,j, j,i, i,k, k,i, j,k,* and *k,j.* Scanning continues on the next row until the entire matrix has been scanned.

Figure 3 illustrates the clusters that were found. A horizontal line denotes a cluster, and the (always non-overlapping) sequences contained in the cluster are indicated by thick bars at the appropriate times. The vertical position of a cluster line has no meaning; it is chosen to avoid overlap with other clusters. For example, the bottom line has 4 thick bars. These correspond to the "A" sections in the opening AABA form. The fourth element of the cluster corresponds to the "A" section when the saxophone enters with BA after the piano solo. Already, the clusters express a rich fabric of relationships, many of which would be described or at least noticed by a human listener. Included within these relationships is the information that the structure is AABA and that the melody returns after the solo with BA rather than AABA and that the last 2 measures are repeated three times near the end. However, the information is not very clear, and there is a lot of detail that is confusing. In the next section, I show how this can be simplified considerably.

9 A Simplified Representation

The goal of this final step is to produce an "explanation" of the entire piece in terms of structural relationships. This is a non-hierarchical explanation and it only presents one *possible* explanation of the material, thereby achieving a great simplification over the clusters, which provide for essentially every possible explanation. Rather than an explanation, you can think of this as a parsing algorithm. The output will be a string of symbols, e.g. AABA, representing musical structure, but unlike a typical parsing, the grammar is unknown, and the symbols are generated by the algorithm rather than being defined in a grammar.

The procedure uses an incremental "greedy" algorithm: proceeding from left-to-right, explain each unexplained note. An "explanation" is a relationship, i.e. "this note is part of a phrase that is similar to this other phrase." If the explanation also explains other notes, they are marked as such and not reexamined (this is the greedy aspect).

More specifically, we start with the first note, and all notes are initially marked as "unexplained." Search for a cluster that contains the first note in one of its sequences. Create a new symbol, e.g. "A," as the label for all notes in the cluster and mark them. Once a note receives a label, the label is not revised. Now, find the next unmarked note and repeat this process until all notes are marked or "explained". Any notes that are not covered by any cluster are ignored.

Figure 4 illustrates the results of this step. Rather than using letters, different shadings are used to show the labels graphically. We have mentioned the AABA structure many

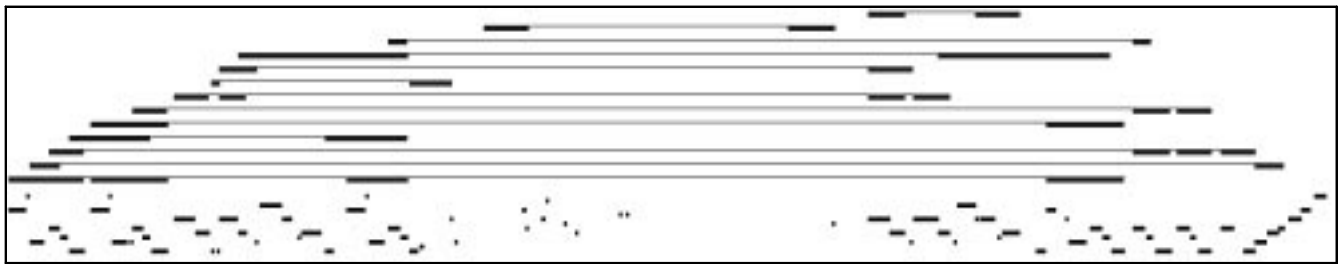

Figure 3. Each horizontal line represents one cluster. The elements of the cluster are indicated by heavy lines, showing the locations of similar melodic sequences. The melodic transcription shown at the bottom.

Figure 4. Simplified structural representation of "Naima," shown below the transcription. Similar sections are shaded similarly. The letter labels were added by hand for illustration purposes. Some of the sections in the middle reflect a spurious similarities between parts of the piano solo.

times in this paper. Looking at Figure 4, the initial AA is indicated by the leftmost two rectangles. The B section is actually composed of the next 3 rectangles, showing substructure of the bridge (which in fact does have the same $b_1b_1b_2$ structure shown here). Next comes the final A section, indicated by a shading that matches the first and second rectangles. The rectangles representing BA are repeated when the saxophone returns after the solo. Thus, the program derives almost exactly the same high-level description a jazz musician would use to describe the structure, without any prior knowledge or grammar of an acceptable description! It would be trivial to use a tree-matching or parsing scheme to map the actual data onto a standard form (including AABA) and then produce a hierarchical description. ("B is structured as $b_1b_1b_2$.")

Further analysis could be applied to the durations of these patterns or motives. It is clear by inspection that the ratios of durations of the $AAb_1b_1b_2A$ form is 221122. There is no way to tell that the unit here is 2 measures, but this would at least give candidates for beat durations that might help a beat-tracking algorithm. Also, the fact that these add up to 10 rather than 8 or 16 is interesting, an observation that a program could easily make if it knew a few conventions about song form.

10 Evaluation With New Input

This analysis method works well for "Naima," which is to be expected. After all, the system was built specifically for this piece, and was modified to overcome problems as they were encountered. What about other input? I tested the analysis system with two other songs: "Freddie the Freeloader," a jazz standard by Miles Davis, and "We Three Kings," a popular Christmas Carol by John H. Hopkins. These were played on trumpet and violin, respectively. Because the interesting part of the work is in the analysis rather than the polyphonic signal processing, these two performances are monophonic. To be fair, these are the first test cases after "Naima" (there was no search for good examples), and the software was not altered or tuned at all to prepare or tune the system for new input.

"Freddie the Freeloader" is a standard 12-bar blues with a simple repeating figure. It was performed by the author, an experienced jazz trumpet player, with a moderate amount of expression including expressive pitch deviations and articulation. The transcription and analysis are shown in Figure 5. At first, this result was disappointing. It only seems to show the riff in the first two measures repeating in measures 3-4 and measures 7-8. Upon closer inspection, more structure is revealed. The 12-bar form was played twice, with a change in the last 2 measures the second time. This created a cluster representing 12 bars repeated twice (the small variation was ignored). When the simplification algorithm looked for an explanation of measure 5, it found this overarching cluster. Thus the explanation of measure 5 is that it is part of a 12-measure sequence that repeats. This ends the explanation because all 24 measures are covered by it. Since measures 1 through 4 of the 12-measure sequence were already explained in terms of a different cluster, it was surprising to see that the program chose the 12-measure sequence to explain measure 5. In "Naima," the clusters do not overlap. Nevertheless, the result makes sense and has a hierarchical interpretation: the piece consists of 12 measures repeated twice. Within each 12 measures, there is additional structure: the first 2 measures are repeated at measures 3-4 and 7-8. (Although transposition relationships are not studied here, it is interesting to note that measures 5-6 are a simple transposition of measures 1-2.)

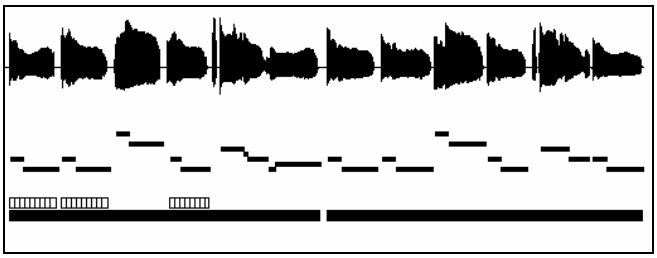

Figure 5. Analysis of "Freddie the Freeloader," a repeated 12-bar blues form. Audio is at top, transcription is in the middle, and the structural "explanation" is at the bottom. The structure shows a repeated riff (3 times) and the repetition of the entire 12-bar blues form.

The Freddie the Freeloader example is successful in that it reveals all the structure that can be obtained by considering repetition, including hierarchal relationships that the software was not intended to find. This example illustrates the importance of hierarchy, and future work should explicitly allow for hierarchical structure discovery. This example also illustrates some of the danger of "greedy" algorithms. In this case, the simplification algorithm destroyed some potentially interesting structure, namely the recurrence of the first two measures at measures 13-14, 15-16, and 19-20. Fortunately, this is redundant information in this case. More work is needed to rank relationships according to their importance though.

"We Three Kings" is a 32-measure form. An amateur student performed it on solo violin. If we were to consider only 4-measure groups, the form would be AABCDDED. The analysis, shown in Figure 6, comes close to revealing this structure. The AA repetition is found, as is the first DD repetition. Interestingly, the program found a similarity between B and E. Any listener would probably agree these sections are similar, sharing some pitches and having similar arch shapes. The program also found a similarity between part of C and part of the final D, thus it did not label the final 4 measures correctly.

It should be emphasized again that the input is audio. No parameters were adjusted in the pitch analysis software, so there are transcription errors. No beat detection is performed, so the program does not have the knowledge of beats or bar lines. Nevertheless, the overall analysis is quite

good, identifying the more important motives A and D, and organizing them within the 32-measure form.

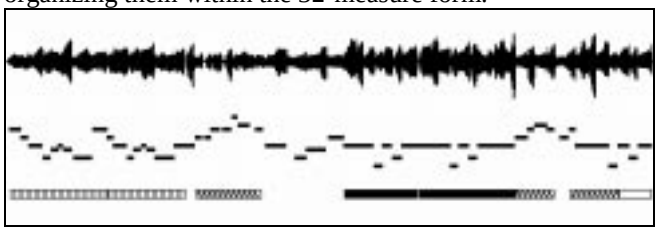

Figure 6. Analysis of "We Three Kings." Audio is at top, transcription is in the middle, and the structural "explanation" is at the bottom. The structure shows a repeated passage (vertical bars) at the beginning and a different repetition (black) in the middle. The contrasting arch-shaped phrases are not literal repetitions, but were found to be similar (diagonal \\\).

Overall, the performance of this analysis software is quite good. The works chosen for analysis are not difficult cases, but on the other hand, the program was not modified or adapted to cope with new problems that arose. To make this a meaningful test, "Freddie" and "We Three Kings" are the first test cases after "Naima." Thus, the algorithm could be expected to give similar performance on comparable pieces.

11 Future Work

Further work is required to consider other relationships. For example, in "Naima," there is a rhythmic motive that occurs frequently, making connections between the A and B parts, and there is a descending pattern in the second half of the B part where essentially the same figure is repeated at different transpositions. It should not be too difficult to detect these relationships, if the notes are detected. (In the present example, some of the shorter notes of the figures are not always transcribed.) The difficult problem seems to be deciding what relationships are important and which take priority. Conklin and Anagnostopoulou (2001) looked at a statistical measure for the repetition of a pattern by chance as a way to decide if a relationship is significant or not, and perhaps similar techniques could be applied here.

This work could benefit from better transcription tools. As mentioned earlier, there is work that already demonstrates impressive performance on much more difficult transcription tasks. Another possibility is to apply polyphonic transcription and look for harmonic relationships within a polyphonic preformance. We are pursuing this idea now, using a transcription system created by Matija Marolt (2001). We plan to perform an analysis very much like the one described here but using harmonies rather than pitches. This will require a similarity measure for harmony and ways to combine outputs from the transcriber into harmonic regions. It will be interesting to see what this approach does with the piano solo in "Naima." (Our simple pitch analysis detected very little of the piano solo, so the music analysis is mostly vacant during the solo section, but Marolt's system captures and transcribes much of the polyphonic piano solo.)

It is important to try this work on a wider range of pieces and to work on techniques that work robustly with all kinds of music. It may be unreasonable to expect a machine to "understand" music as well as humans, but we want the system to be as general as possible. This work might be extended to handle a broader range of pieces.

It is not at all clear that the algorithms presented here are the best for the task. In fact this work was originally intended as a proof-of-concept demonstration, and it was surprising to see how well it works. An improved version should use a more reliable measure for melodic similarity (Mazzoni & Dannenberg, 2001) and should be less eager to throw out entries in the similarity matrix. Perhaps a hierarchical or lattice-based representation of similarities would be better. Finally there is much more that can be done in terms of harmonic analysis, melodic tension and resolution, and rhythmic structure.

12 Conclusions

Listening to music is a rich human experience that no computer model can fully replicate. However, some of the principle activities and by-products of music listening may be subject to modeling with simple mechanisms. Specifically, music listening involves the recognition of patterns and relationships. The most important relationship is repetition. This work demonstrates how a model of musical listening can be constructed upon the idea that musical repetition gives rise to structural relationships. Listening is largely a matter of finding and organizing these relationships in order to construct an "explanation" of the music in terms of how each part relates to some other part. This model is realized by a fully automated music analysis system that accepts audio as input and produces a structural description as its output. Motives are identified, and the structural description tells how a small set of motives can be ordered and repeated to form the music as a whole. This information reflects common notions of musical description, including abstract form (e.g. AABA), identification of common themes or motives, and the temporal organization of phrases into 4-, 8-, 12-, and 16-measure groups.

The analysis system has been demonstrated on three examples that include jazz and popular melodies, and in all cases the analysis is quite close to a standard interpretation. Given the difficulties of acoustic analysis, it is quite remarkable how well the system produces explanations of structure within these examples. Currently, the system is somewhat limited by the quality of transcription. With improvements to transcription, future enhancements should allow the identification of transposition as a relationship, thus providing an even more detailed and complete analysis.

13 Acknowledgements

The author wishes to thank his collaborators at the University of Michigan, especially Bill Birmingham. This work is supported in part by NSF Award #0085945.

References

Coltrane, J. (1960). Naima, *Giant Steps*: Atlantic Records.

Conklin, D., & Anagnostopoulou, C. (2001). "Representation and Discovery of Multiple Viewpoint Patterns." *Proceedings of the 2001 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 479-485.

Cope, D. (1996). *Experiments in Musical Intelligence* (Vol. 12). Madison, Wisconsin: A-R Editions, Inc.

Goto, M. (2001, May). "A Predominant-F0 Estimation Method for CD Recordings: MAP Estimation using EM Algorithm for Adaptive Tone Models." *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, pp. V-3365-3368.

Lartillot, O., Dubnov, S., Assayag, G., & Bejerano, G. (2001). "Automatic Modeling of Musical Style." *Proceedings of the 2001 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 447-454.

Marolt, M. (2001). "SONIC: Transcription of Polyphonic Piano Music With Neural Networks." *Workshop on Current Research Directions in Computer Music*. Barcelona, Spain: Audiovisual Institute, Pompeu Fabra University, pp. 217-224.

Mazzoni, D., & Dannenberg, R. B. (2001). "Melody Matching Directly From Audio." *2nd Annual International Symposium on Music Information Retrieval*. Bloomington: Indiana University, pp. 17-18.

Meek, C., & Birmingham, W. P. (2001). "Thematic Extractor." *2nd Annual International Symposium on Music Information Retrieval*. Bloomington: Indiana University, pp. 119-128.

Mongeau, M., & Sankoff, D. (1990). Comparison of Musical Sequences. In W. Hewlett & E. Selfridge-Field (Eds.), *Melodic Similarity Concepts, Procedures, and Applications* (Vol. 11). Cambridge: MIT Press.

Mont-Reynaud, B., & Goldstein, M. (1985). "On Finding Rhythmic Patterns in Musical Lines." *Proceedings of the International Computer Music Conference 1985*. San Francisco: International Computer Music Association, pp. 391-397.

Narmour, E. (2000). "Music Expectation by Cognitive Rule-Mapping." *Music Perception, 17*(3), 329-398.

Rolland, P.-Y., & Ganascia, J.-G. (2000). Musical pattern extraction and similarity assessment. In E. Miranda (Ed.), *Readings in Music and Artificial Intelligence* (pp. 115-144): Harwood Academic Publishers.

Rowe, R. (1993). *Interactive Music Systems: Machine Listening and Composing*: MIT Press.

Simon, H. A., & Sumner, R. K. (1968). Pattern in Music. In B. Kleinmuntz (Ed.), *Formal Representation of Human Judgment*. New York: Wiley. Reprinted in S. Schwanauer and D. Levitt, eds., *Machine Models of Music*, MIT Press, pp. 83-112.

Stammen, D., & Pennycook, B. (1993). "Real-Time Recognition of Melodic Fragments Using the Dynamic Timewarp Algorithm." *Proceedings of the 1993 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 232-235.

Tolonen, T., & Karjalainen, M. (2000). "A computationally efficient multi-pitch analysis model." *IEEE Transactions on Speech and Audio Processing, 8*(6).