GRAPH-BASED RHYTHM INTERPRETATION

Rong Jin
Indiana University
School of Informatics and Computing
rongjin@indiana.edu

Christopher Raphael
Indiana University
School of Informatics and Computing
craphael@indiana.edu

ABSTRACT

We present a system that interprets the notated rhythm obtained from optical music recognition (OMR). Our approach represents the notes and rests in a system measure as the vertices of a graph. We connect the graph by adding voice edges and coincidence edges between pairs of vertices, while the rhythmic interpretation follows simply from the connected graph. The graph identification problem is cast as an optimization where each potential edge is scored according to its plausibility. We seek the optimally scoring graph where the score is represented as a sum of edge scores. Experiments were performed on about 60 score pages showing that our system can handle difficult rhythmic situations including multiple voices, voices that merge and split, voices spanning two staves, and missing tuplets.

1. INTRODUCTION

Past decades have seen a number of efforts on the problem of Optical Music Recognition (OMR)with overviews of the history and current state of the art found at [2, 3, 8, 14]. OMR can be divided into two subproblems: identifying the music symbols on the page and interpreting these symbols, with most efforts devoted to the former problem [7,13,16]. However, the interpretation problem is also important for generating meaningful symbolic representations. In this paper, we focus on the rhythm interpretation of musical symbols, which appears to be the most challenging interpretation problem.

Many OMR systems [11] perform some sort of rhythm interpretation in order to play back and verify the recognized music symbols. When there are not enough notes or too many notes to match the meter of the measure, the OMR system often “flags” the measure to suggest that there is something wrong, alerting the user to correct the measure. In this way, rhythm interpretation is used as a checking tool for correcting recognized scores.

There are a few research efforts that correct recognition results automatically. Droettboom [6] proposed metric correction as part of an OMR system. Using the fact

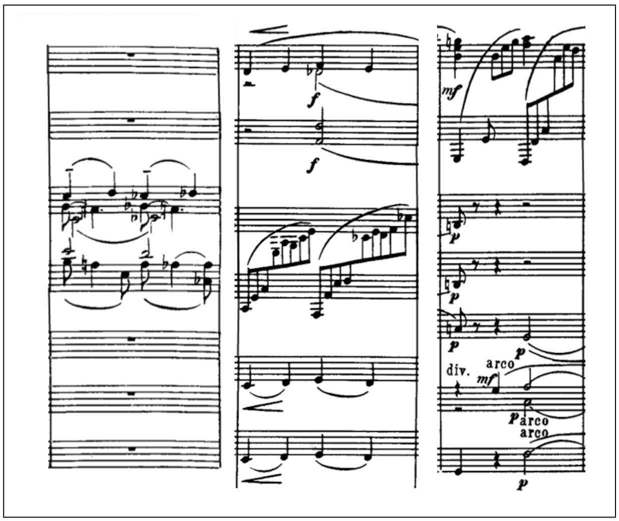

Figure 1. Three system measures from Rachmaninoff Piano Concerto No.2 showing some of the difficulties in interpreting rhythm. All three measures are in 4/4 time.

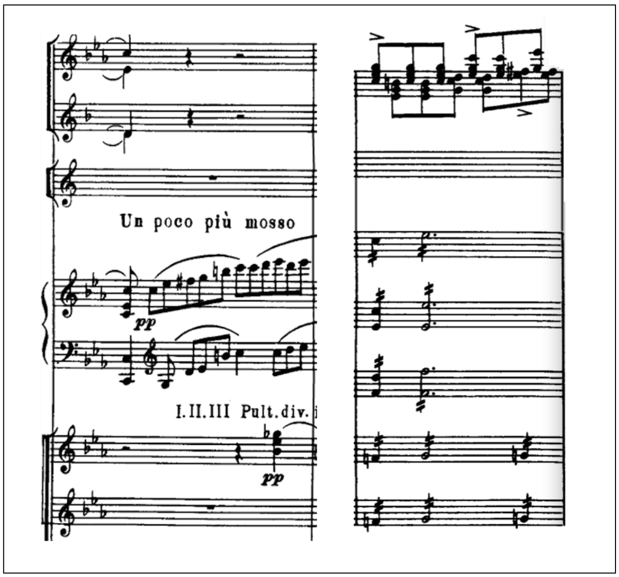

Figure 2. Two system measures from Rachmaninoff Piano Concerto No.2 showing some of the difficulties in interpreting rhythm. Both are in 4/4 time.

that rhythmically coincident notes are usually aligned vertically, this work applies different corrections on inconsistent notes. Church [5] proposed a rhythmic correction with a probabilistic model that converts the rhythm of a suspicious measure to the most similar measure in the piece. Byrd [4] proposed improving OMR with multiple recognizers and sequence alignment.

The approaches mentioned above work for simpler situations such as monophonic music or measures without complex tuplets. However, some music scores, especially those for piano, are filled with rhythmically challenging situations such as missing tuplets or voices that come and go within a measure. Simple approaches are likely to fail on a significant proportion of these measures.

Our paper differs from other work we know by addressing the most challenging examples using *complete* information (the system measure), instead of trying to correct the misrecognized symbols. Our research questions are: given perfect symbol recognition is the system able to understand rhythm as a human would? When there are multiple voices interwoven in one measure, can the system separate the voices? When there are implicit symbols such as omitted rests and missing tuplets, can the system still interpret correctly?

Figures 1 and 2 show some challenging examples that illustrate the problem we address. The left measure in Figure 1 shows an example using multiple voices. When multiple voices are present it is nearly impossible to interpret rhythm without identifying these voices as such. In an effort to avoid overlapping symbols, some notes in the measure that ideally should align vertically do not. The middle measure in Figure 1 shows an example of missing tuplets (tuplets are not labeled). What is most unusual, and would likely go unnoticed by anyone other than an OMR researcher, is that these beamed groups would normally be written with two beams rather than one, though the meaning is still clear. In addition, the 9-tuplet is not explicitly indicated with a numeral — a common notational convention.

The right measure in Figure 1 shows another example of missing triplet for the 3 beamed eighth notes in the first staff, as well as a quarter note plus an eighth note pair in the second staff. A further complication is that this measure is, in some sense, incomplete, as the voice in the second staff jumps onto the first staff on the second quarter and then jumps back on the third quarter. The left measure in Figure 2 demonstrates an example of special beaming of a sextuplet where the first eighth note is separate from five beamed eighth notes. The right measure in Figure 2 demonstrates an example where all four beamed groups are triplets while the voice jumps back and forth between the two beamed groups.

The examples all seem innocent until one considers the assumptions on rhythm notation that must underlie an interpretation engine. One quickly comes to see that typical *in vivo* notation contains a fair amount of "slang" that may be readily understood by a person familiar with the idiom, but is much harder for the machine. [9] has more demonstrations of such "slang" in music scores.

In this paper we present an algorithm that is generally capable of correctly interpreting notation such as the examples we have discussed. In our presentation, Section 2 introduces our rhythm graph and optimization on the graph score. In Section 3, we present our experiments on three scores and discuss the results.

2. METHODS

2.1 Input

We first perform optical music recognition with our *Ceres* [12] OMR system taking the score image as input. The output is stored as a collection of labeled primitive symbols such as solid note head, stem, beam, flag, and etc., along with their locations. The user deletes or adds primitive symbols using an interactive interface. Editing symbols at the primitive level allows us to keep useful information such as stem direction and beaming as well as the exact primitive locations which are important for rhythm interpretation.

After this correction phase, we assemble the primitive symbols into meaningful composite symbols (chords and beamed groups). This step is done in a simple rule-based method. Each note or rest is assigned to the staff measure it belongs to.

2.2 Rhythm Graph

We form a graph of the rhythmically relevant symbols for each system measure. The set of vertices of the graph, which we denote as V , are the notes, rests, and bar lines belonging to the system measure. All vertices are given a nominal duration represented as a rational number. For example, a dotted eighth would have nominal length 3/16, while we give the bar lines duration 0. Sometimes the actual vertex duration can differ from the nominal length, as with missing tuples. In these cases, we need to identify which symbols are tuplet symbols in order to interpret the rhythm correctly.

Vertices can be connected by either voice or coincidence edges, as shown in Figure 3. Voice edges, which are directed, are used for symbols whose position is understood in relation to a "previous" symbol, as in virtually all monophonic music. That is, the onset time of a symbol on the "receiving" end of a voice edge is the "preceding" symbol's onset time plus duration. Coincidence edges link vertices that share the same onset time, as indicated by their common horizontal location. Using these edges we can infer the onset time of any note or rest connected to a bar line. We denote by E the complete collection of all possible edges.

We formulate the rhythm interpretation problem as constrained optimization. Given the set of vertices,V, and possible edges, E, we seek the subset of E, E^*, and the labeling of V that maximizes

$$H = \sum_{e \in E^*} \phi(e) + \sum_{v \in V} \varphi(l(v)) \qquad (1)$$

where function $\phi(e)$ represents how plausible each edges is according to the music rules, l labels vertex v as tuplet or non-tuplet, and function $\varphi(l)$ penalizes labeling vertices as tuplet so as to favor simple interpretations whenever possible. The subset E^* and labeling are constrained to construct a consistent and connected graph.

2.3 Constructing edges

We construct the graph beginning with the left bar line (which has an onset time of 0), by iteratively connecting new vertices to the current graph with voice and coincidence edges until all vertices form a single connected component. More specifically, we connect the current vertex with a voice edge to a previously visited vertex. This vertex has to be either a bar line or a vertex in the same staff measure. (Piano staves are treated as one staff because voices often move between left and right hand parts.) This new voice edge defines a unique onset for the current vertex. Then we add coincidence edges between the current vertex and all past vertices so that both have nearly the same horizontal position and have the same onset time. We may also add coincidence edges between the incoming vertex and a past vertex having a *different* onset time, leading to a conflict that must be resolved, as discussed in Section 2.4. Different combinations of edges give different onset times to the vertices.

As an edge e is introduced to the graph we score it according to its plausibility $\phi(e)$. There are different kinds of musical knowledge [15] we hope to model in computing these scores, as follows.

1. The left bar line has an onset time of 0. The right bar line has an onset time of the measure's meter. No vertices can have onset times greater than the meter.

2. The onset times must be non-decreasing in the horizontal positions of the symbols in the image. That is, if vertex A lies to the left of vertex B it cannot have an onset that is after that of vertex B.

3. A vertex has a unique onset time. Thus, if multiple paths connect a vertex to the graph they must give the same onset time.

4. Vertices connected by coincidence edges should have the same approximate horizontal position in the image. Vertices with the same horizontal image positions should should have the same onset time.

5. Vertices in a beamed group note are usually connected by voice edges, while we penalize voices that exit a beamed group before it is completed.

6. Vertices connected by a voice edge usually have the same stem direction and tend to appear at similar staff height.

The first two rules above are hard constraints that *must* be followed. When they are violated our algorithm simply will not add the offending edge. The other rules can be violated for different reasons. For example, symbols having the same onset time may not align in the image because one is moved to avoid overlap with other symbols, or because the image is skewed or rotated through the scanning process. Such violations lead to penalties of the edge scores.

2.4 Conflict Resolution by Reinterpretation

If we disregard the right bar line and construct a spanning tree from the remaining vertices we are guaranteed that every vertex can be reached through a unique path starting from the left bar line, thus giving each vertex a unique onset time. While this approach has the compelling appeal of simplicity, it would fail in any case where the nominal note length is not the correct interpretation, as with missing tuplets. Instead, we identify such cases by allowing *multiple* paths to a vertex, and thus multiple rhythmic interpretations. When the result of these multiple paths gives conflicting onset positions for a vertex we consider reinterpreting some notes in terms of missing tuplets to resolve the conflict. In such a case we treat the earlier onset time as the correct one, while reinterpreting the path leading to the later onset time. This is because the nominal length of a tuplet note is usually greater than the true length. While there are exceptions to this rule, as with duplets in triple meter, we do not treat such cases in the current work.

As an example, consider the situation in Figure 3. Here the first coincidence edge considered (dotted line in the 1st graph) does not create any conflict since both paths give the onset position of 1/4. However, the coincidence edge for the quarter note on the top staff (dotted line in the 2nd graph) gives the onset time of 1/2 while the voice edge gives the onset time of 5/8, thereby generating a conflict. Thus we must reinterpret the path giving the later onset time of 5/8 to be consistent with the onset time of 1/2. In this case the desired interpretation is that the three eighth notes form an implicit triplet, and thus have note lengths of 1/12 rather than 1/8 (bottom graph). Another example of a conflict arises when a voice edge links to the right bar line and attributes an onset time for the bar line other than its true position (which is the meter viewed as a rational number). In this case we must reinterpret the path leading to the right bar line.

When reinterpreting we must consider the path that generates the onset position in conflict — but how far backward should we go? The collection of reinterpretable vertices could spill over into multiple voices and staff measures, thus generating an intractable collection of possibilities to consider. Here we make some simplifying assumptions to keep the computation from becoming prohibitively large. First of all, recall that we consider the staff measures of a system one at a time. After a staff measure is completely analyzed and reduced to a single interpretation, we do not consider future reinterpretations of the measure. Thus reinterpretation is confined to the current staff measure (or two staves in the case of the piano). Furthermore we do not allow the reinterpretation process to generate additional inconsistencies. This rules out the reinterpretation of any vertex connecting to a measure in a previously analyzed staff measure. Even with these restrictions the computation can still be significant, as we must simultaneously consider the possibility of a number of different tuplet hypotheses, thus requiring an effort that is exponential in the number of hypotheses.

One might contrast this approach with a purely top-down model-based strategy that considers every possible rhythmic labeling. Such a strategy would be our preference if computationally feasible, and, in fact, was the approach we explored in [10]. The problem is that there are, *a priori*, a large enough collection of possible labelings so that, when coupled with unknown voicing, the computation does not always prove tractable. This is why we uncover candidates for reinterpretation prompted by coincidence edges. Thus the modeling of our algorithm lies somewhere between top-down and bottom-up recognition. It is model-based, yet it relies on the data itself to prompt certain data interpretations. While not necessarily an argument in favor of our approach, this appears to be a central part of the human strategy for rhythm understanding.

We consider several cases of reinterpretation:

1. A beamed group can be reinterpreted as a beamed tuplet note of simple duration (1/2, 1/4, etc.), as in the left measure of Figure 2.

2. Three consecutive vertices that add up to 3/8 could be reinterpreted as missing triplet of total length 1/4, as in the middle measure of Figure 2. This rule can be generalized to include other kinds of triplets (quarter note or sixteenth note) and to include tuplets other than 3.

3. We can *globally* reinterpret all vertices along the voice path, as in the right measure in Figure 2, meaning that all note lengths are rescaled to create the desired collective duration.

The score function $\varphi(l(v))$ in Eqn (1) penalizes the complexity of a reinterpretation, thus favoring simple interpretations whenever possible.

2.5 Dynamic Programming for Optimization

During graph construction, each time we add a new vertex into the graph we consider adding voice edges between the new vertex and all vertices already in the graph. Thus, only considering the voice edges, the number of possible graphs with n vertices would be $n!$. Since a common system measure may have more than 50 vertices, it is computationally infeasible to search the whole graph space. This situation can be improved by dynamic programming: after any new vertex has been added to the graph, if two different graphs give identical onset times for each vertex we prune the one with lower score.

The order in which the vertices are considered is important in producing a feasible search. One way would be to visit all vertices in the system measure according to their horizontal location on the image. The problem with this approach is that the constraints imposed by the right

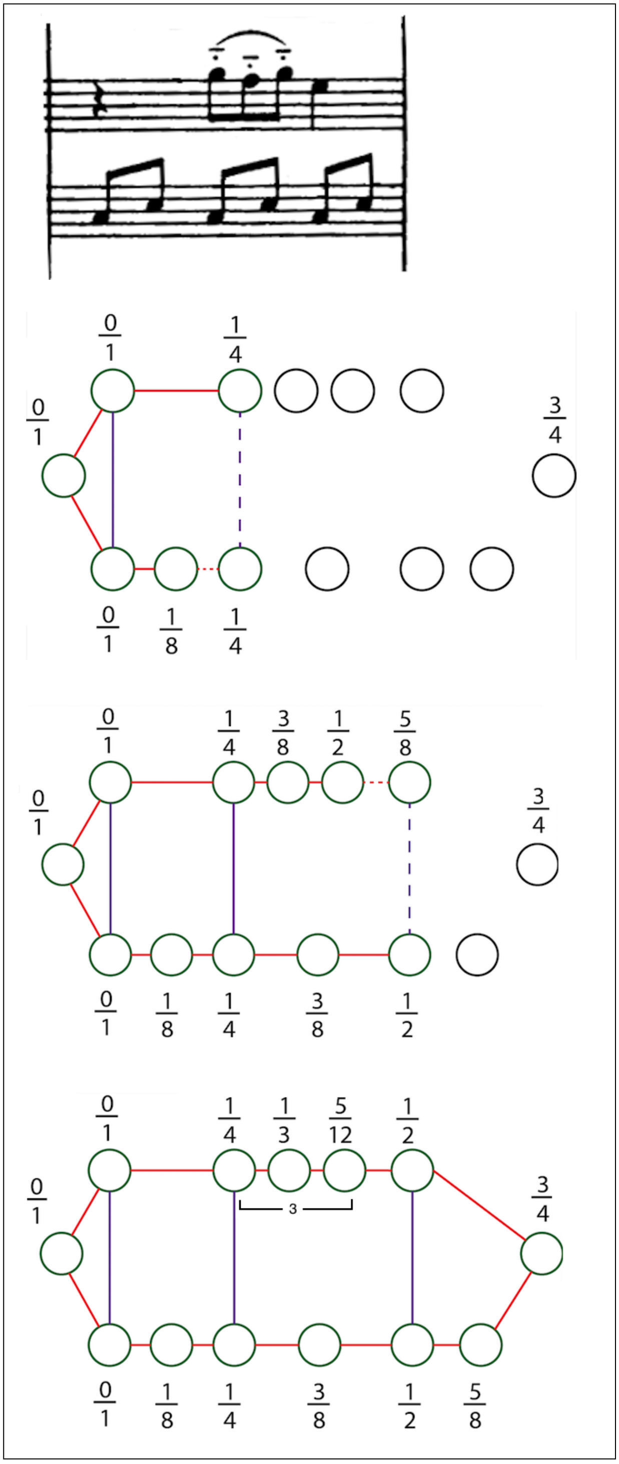

Figure 3. Constructing the rhythm graph of an example measure. Voice (red) and Coincidence (Purple) edges are automatically constructed to identify the onset time of vertices (notes, rests and bar lines).

bar line, which has a known onset position, do not come into play until the very end of the graph construction. An alternative first considers the vertices in left-right manner from a single staff measure, then continuing one-by-one with the other staff measures. Each time a staff measure is completed we continue only with the best scoring single graph. In this way, we will greatly reduce the number of partial graphs we carry through the process.

Among all measures in our experiments the maximum number of graph hypotheses we encounter during the DP computation is usually less than 100, even in the system measures with 50 to 60 vertices. The measures posing the greatest computational challenge are those having multiple voices, missing tuplets, and, at the same time, similar rhythm between the voices. The left example measure in Figure 4 shows such a case. It may seem easy for a person to recognize that there are two voices in the first staff. Here four quarter notes form one voice, and four pairs of triplets, consisting of an eighth rest and two eighth notes, form another voice. However, it's not an easy task for a computer. The second staff measure doesn't provide much information since it also has the similar missing tuplets which are hard to distinguish from nominal rhythm until one encounters the right bar line. Other measures in the same system also don't provide aligned symbols to anchor the search. The number of graph hypotheses for this system measure grows up to 2600 at the end of the measure. This measure represents the maximum number of hypotheses attained throughout our experiments. This is still easily feasible computationally.

3. EXPERIMENTS

In the experiments, we have chosen three different scores of varying degrees of rhythmic complexity for evaluation, all taken from the IMSLP [1].

3.1 Rachmaninoff Piano Concerto No.2

The orchestra score of Rachmaninoff Piano Concerto No. 2 is a highly challenging example for our rhythm interpretation algorithm. The score has 371 system measures, with each system measure containing up to 15 staff measures. The piece covers different types of rhythmic difficulties such as polyphonic voices, missing tuplets, and voices moving between staff measures. In addition some pages of the score are rotated and skewed due to the scanning process, creating difficulty detecting coincidence between notes.

We get 355 out of 371 (95.7%) system measures correctly. In the following paragraphs, we will discuss three representative examples in which our system fails to find the correct rhythm.

Failure case 1 In the left example in Figure 4 we fail to interpret all the missing triplets. The result produced by our system did not recognize the first and last triplet in the first staff, instead treating those beamed eighth notes as normal eighth notes. The system gives the left eighth note in the beam the same onset time as the eighth note rest, explaining it as coincidence with the eighth note rest since they almost align vertically. In this case we found that the correct interpretation was actually generated by our system, but survives with a lower score. This type of scenario, where the correct interpretation survives but does not win, occurs a number of times in our experiments. In this case, the reason is because we give a high penalty for tuplet reinterpretation, while a give comparatively lower penalty when allegedly coincident symbols are not perfectly aligned. Therefore, the state that has fewer tuplets but worse alignment gets a higher score.

Failure case 2 The right example in figure 4 is another example where our system does not produce the correct rhythm. The difficulty in this measure is the voice that moves between the treble and bass staves of the piano. While we successfully recognized two missing sextuplets in the treble staff, we failed to recognize that the quarter note in treble staff and eighth note in the bass staff form a triplet. In our result, they are interpreted as a normal quarter note and a normal eighth note with the eighth note aligned to the 3rd sixteenth through a coincidence edge. This happens because we impose a penalty for interpreting a missing tuplet, while the eighth note aligns reasonably well with the third 16th note, providing a plausible explanation. However, the isolated eighth note is the only note that has the wrong onset time. This case also shows that our algorithm is capable of recovering from local errors to produce mostly correct results, even though not perfect.

Failure case 3 Our third incorrect case is shown in the left of Figure 5. In the first staff of this example, the dotted half note chord and first eighth note in the first beam group both begin at the start of the measure. However, we have a maximal horizontal distance between two notes that have the same onset time, which serves the important role of pruning graphs graphs that exceed this threshold — usually this is the correct thing to do. In this particular case these two notes exceeded the threshold, thus we lose the correct interpretation. For such a case, we can always make the threshold larger, but this weakens the power of the alignment rule elsewhere. Of course, there will always be special cases where our threshold is not large enough. In the right measure in Figure 5, the eighth rest and whole note "high" c in the first staff are very far away from the half note in staff three due to the long grace note figure. Presumably the grace note figure begins *on the beat*, so the coincidence suggested by the score is correct, though this peculiarity lies outside of the basic modeling assumptions we have employed: here two notes at the same rhythmic position are not intended to sound at the same time! We have a few other examples of this general type of failure, such as when we can't compute horizontal distances accurately due to image skew. Given the reasons above, we decide to keep the threshold as strict as it is, because it provides a significant help with keeping the computation tractable.

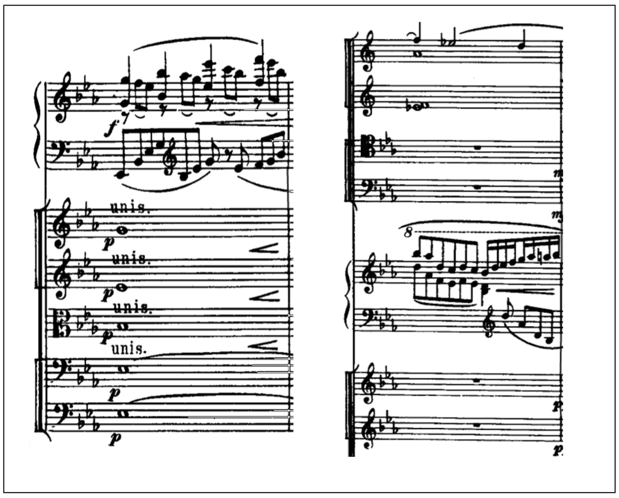

Figure 4. Examples for failure case 1 and failure case 2 from Rachmaninoff Piano Concerto No. 2. Both measures are in 4/4 time.

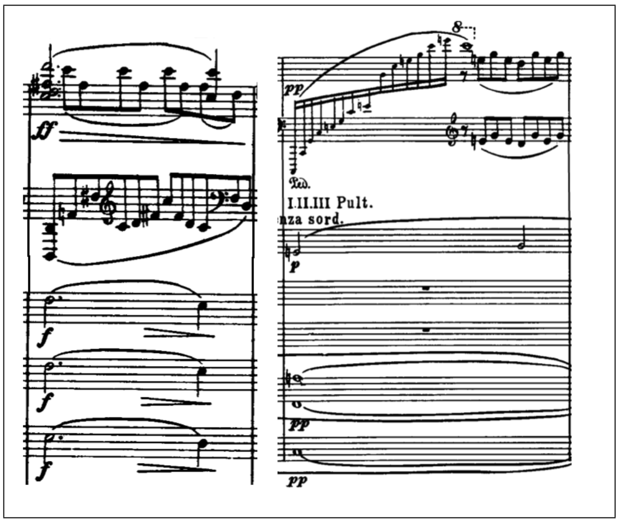

Figure 5. Examples for failure case 3 from Rachmaninoff Piano Concerto No.2. Both measures are in 4/4 time.

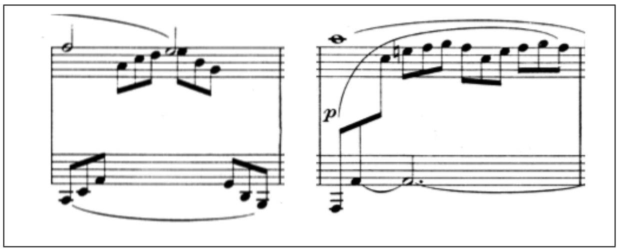

Figure 6. Two examples from Debussy's 1st Arabesque. Both measures are in 4/4 time.

3.2 Borodin String Quartet No.2, 3rd Movement

We also tested on the 3rd movement (*Notturno*) from Borodin's 2nd String Quartet. This is a "medium" difficulty score consisting of 4 staves for each system. The third movement has 180 systems measures over 6 pages. 22 out of 180 system measures contain triplets, and, while all of these are explicitly notated with numerals in the score, we deliberately didn't include these in our rhythm interpretation process. The system gets 100 percent correct rhythm on all of these measures.

3.3 Debussy 1st Arabesque

Usually the more staves in a system, the more coincidence edges between different staves, thus providing anchors for reinterpretation when needed. Thus solo piano music can be particularly challenging with only two staves. In measures that are monophonic or homophonic we can't identify inconsistencies until we reach the end of the measure as both nominal and tuplet hypotheses are consistent with spacing. In order to demonstrate that our system is also capable of handling these challenges, we experimented on the first of the two Debussy Arabesques, containing 107 measures.

This piece has a variety of rhythmic difficulties. 73/107 (68%) of the system measures have at least one, and up to six missing tuplets, while 17/107 measures contain voices moving between the two staves. This latter category is particularly difficult because the measures are monophonic as in Figure 6, and thus do not provide coincidence clues. Therefore, our algorithm only sees conflicts at the end of the measure and must reinterpret the entire measure at once. However, our results show that we are generally capable of handling such situations. There's only one measure that we don't get exactly correct as shown in the right of Figure 6. In this measure, there are four missing beamed group triplets. In our best scoring solution, we found the first and last triplets but are missing the middle two. The correct interpretation also survives into the final list but with a lower score.

4. CONCLUSION

We have presented a graph-based rhythm interpretation system. Experiments show that given the perfect symbol recognition, our system is generally capable of interpreting difficult notation involving separating multiple voices and identifying implicit symbols such as missing tuplets. It also shows that it's a difficult and interesting problem and worth further exploration. One possibility will be using trained penalty parameters for a particular score. A rare notation or rhythmic pattern could appear repeatedly in one score, thus we hope an adaptive model would improve the result. Also, since there are always exceptions in all music-related questions, human interactive methods are another interesting direction to explore.

5. REFERENCES

[1] International music score library project. *http://imslp.org/.*

[2] D. Bainbridge and T. Bell. The challenge of optical music recognition. *Computers and the Humanities*, 2001.

[3] D. Blostein, H. Dorothea, and H. Baird. A critical survey of music image analysis. *Structured Document Image Analysis. Springer Berlin Heidelberg*, 1992.

[4] D. Byrd. Prospects for improving OMR with multiple recognizers. In *Proceedings of the International Symposium on Music Information Retrieval*, 2006.

[5] M. Church and M. Cuthbert. Improving rhythmic transcriptions via probability models applied post-OMR. In *Proceedings of the International Symposium on Music Information Retrieval*, 2014.

[6] M. Droettboom, I. Fujinaga, and K. Macmilan. Optical music interpretation. In *Structural, Syntactic, and Statistical Pattern Recognition*, 2002.

[7] I. Fujinaga. Adaptive optical music recognition ph.d. thesis, mcgill university,montreal. 1997.

[8] I. Fujinaga. Optical music recognition bibliography. *http://www.music.mcgill.ca/ ich/research/omr/omrbib.html*, 2000.

[9] J. Hook. How to perform impossible rhythms. *Society for Music Theory*, 2011.

[10] R. Jin and C. Raphael. Interpreting rhythm in optical music recognition. In *Proceedings of the International Symposium on Music Information Retrieval*, 2012.

[11] G. Jones, B. Ong, I. Bruno, and K. Ng. Optical music imaging: music document digitisation, recognition, evaluation, and restoration. *Interactive Multimedia Music Technologies*, pages 50–79, 2008.

[12] C. Raphael and R. Jin. The Ceres system for optical music recognition. In *International Conference on Pattern Recognition Applications and Methods*, 2014.

[13] C. Raphael and J. Wang. New approaches to optical music recognition. In *Proceedings of the International Symposium on Music Information Retrieval*, 2011.

[14] A. Rebelo, G. Capela, and J. Cardoso. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 2012.

[15] G. Reed. Music notation:a manual of modern practice. 1979.

[16] F. Rossant and I. Bloch. Robust and adaptive OMR system including fuzzy modeling,fushion of musical rules, and possible error detection. In *EURASIP Journal on Applied Signal Processing*, 2007.