Tree-structured representation of musical information

David Rizo, José Manuel Iñesta and Francisco Moreno-Seco

Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante,
Ap. 99, E-03080 Alicante, Spain
{drizo,inesta,paco}@dlsi.ua.es

Abstract. The success of the Internet has filled the net with lots of symbolic representations of music works. Two kinds of problems arise to the user: content-based search of music and the identification of similar works. Both belong to the pattern recognition domain. In contrast to most of the existing approaches, we pose a non-linear representation of a melody, based on trees that express the metric and rhythm of music in a natural way. This representation provide a number of advantages: more musical significance, more compact representation and others. Here we have worked on the comparison of melodies for identification.
Keywords: Multimedia applications, computer music, structural recognition.

1 Introduction

There are lots of symbolic representations of music works in the Internet (for example, in standard MIDI file format). Two kinds of problems arise to the user: content-based search of music and the identification of similar works. Both belong to the pattern recognition domain. The applications range from the study and analysis tasks in musicology to the detection of plagiarism, useful to protect copyrights in the music record industry.

Traditionally music has been represented by means of a set of tuple strings, where each tuple, in diverse ways, usually contains information on pitch, duration and onset time. Both the retrieval and the comparison have been tackled with structural pattern matching techniques in strings [4]. There are some other approaches, seldom applied, like the geometric one, which transforms the melody into a plot obtained tracing a line between the successive notes in the staves. This way, the melody comparison problem is converted into a geometric one [2].

In this paper, we use a nonlinear representation of melody: by means of trees that express the metric and rhythm of music in a natural way. The approach to tree construction is based on the fact that the different music notation figures are designed on a logarithmic scale: a whole note lasts twice a half note, whose length is the double of a quarter note, etc. This representation provides us with a richness of possibilities that the strings never will: implicit description of rhythm and more musical meaning and automatic emphasising of relevant notes, for example. Moreover, the way in which a string representation is coded strongly conditions the outcome of the string processing algorithms [6].

In this work, we have dealt with the comparison of melodic lines and compared the performance to that with string representations. Although tree comparison algorithms have higher complexity than the existing methods for strings, the results improve the ones in the same way using strings. This preliminary results open a promising new field for experimentation in a number of applications on the symbolic representation of music.

Firstly, the method for tree construction is presented and how it deals with the notation problems that may appear. Secondly, a procedure for tree pruning and labelling is described in order to deal with the complexity above described. Then, the method for comparison and the results are presented, and finally conclusions are stated.

2 Tree construction method

As described above, the tree construction method is based on the logarithmic relation among duration of the different figures. A sub-tree is assigned to each measure, so the root of this sub-tree represents the length in time of the whole measure. If just a whole note is found in the measure, the tree will consist of just the root, but if there were two half notes, this node would split into two children nodes. Thus, recursively, each node of the tree will split into two until representing the notes actually found in a measure (see Fig. 1).

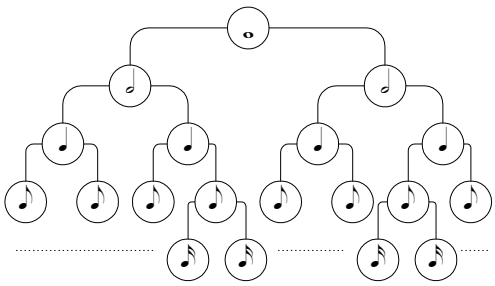

Fig. 1. Duration hierarchy

For the representation of a melody, each leaf node represents a note or silence. Different kind of labels can be used to represent a note, but we have used five of them: 1) the absolute pitch (the name and octave of each note); 2) the pitch name (same as before but without octave); 3) the contour (three possible labels: $+1$ if the pitch of the note is higher than that of the one before, -1 if is lower and 0 if is the same); 4) the high-definition contour (same as before but also including $+2$ and -2 if the pitch differences exceed ± 4 semitones) [10]; and 5) intervals: the difference in semitones between a note and the one before. Silences are represented with a special label. Each node has an implicit duration according to the level of the tree in which it appears. In addition to the duration of the notes, the left to right ordering of the leaves also establish the time in the

measure in which they begin to play. Initially, only the leaf nodes will contain a label value, but then a bottom-up propagation of these labels is performed to fully label the tree nodes. The rules for this propagation will be described later.

An example of this scheme is presented in Fig. 2 with pitch labels. The left child of the root has been splitted into two subtrees to represent the quarter note C. This one lasts the time represented by the leaf node in which it is: one beat in the measure. In order to represent the durations of the two eighth notes it will be necessary to unfold one more level. The half note F onsets at the third beat and, as it lasts two beats, its position is in the second level of the tree.

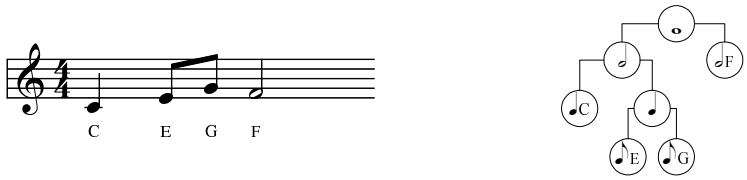

Fig. 2. Simple example of tree construction.

In some occasions the situation can be more complicated. For example, if the duration of a note is greater than that of the half corresponding subdivision, like happens for dotted or tied notes (see Fig. 3). In this situation, a note can not be represented only by the complete subtree in which it onsets. It is well known that the ear does not perceive in a very different way a whole C note from two half C notes played one after the other, even more if the interpreter play them *legato* [9]. Thus, when a note exceeds the proper duration, we will subdivide it in order to complete the time of the note by means of nodes in sub-trees enough to complete the duration of the note with smaller Also, tied notes, are represented in the same way, breaking the tie in the tree representation. In Fig. 3 an example of these situations is presented and how they are represented in this scheme.

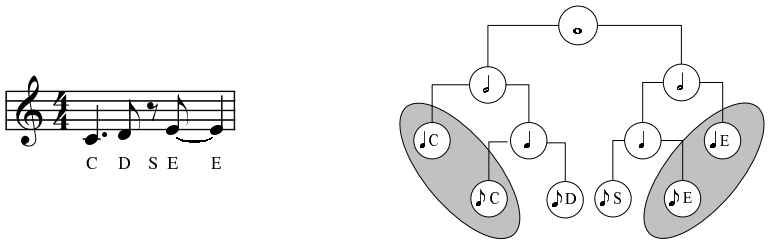

Fig. 3. Tree representations of notes exceeding their notation length: dotted and tied notes. Rounded leaves correspond to those notes. 'S' stands for "silence".

Other music notation events, like other rhythm meters, non-binary structures, compound meters, adornment notes, trills, etc., can appear, but the de-

scribed method can be extended without difficulty to cope with all these situations [8].

Once each measure has been represented by a single sub-tree, joining all of them is needed to build the tree for the complete melody. For this, a method for grouping the sub-trees is required. Initially we could group them by adjacent pairs, hierarchically, repeating this operation bottom-up with the new nodes until a single tree is obtained. Nevertheless, trees would grow in height very quickly this way and this would make the tree edit distance computation algorithms very time consuming. We have chosen to build a tree with a root for the whole melody and each measure is a child of the root (Fig. 4). Thus, the level of the tree for the whole melody only grows in one with respect to the measure's sub-tree. This is like having a forest but linked to a common root node that represents the whole melody.

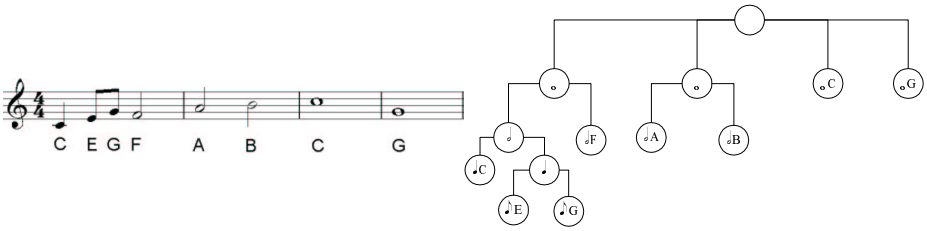

Fig. 4. All the measures of a melody are represented by a single tree.

3 Bottom-up propagation of labels and pruning

The tree edit distance algorithms need all the nodes to have a label [1]. We will use a set of rules for the propagation of labels from the leaves to the root according to musicology criteria (see below). The propagation of a label upwards implies that the note in that node is more important than that of the sibling node. The propagation criteria proposed here are based on the fact that, in a melody, there are notes that contribute more than others to its identity.

In addition, the resulting trees can be very complex if the rhythmical structure does not agree exactly with the successive subdivisions of the binary tree, for example in real-time sequenced MIDI files. This implies a greater time and space overhead in the algorithms [1] and makes it more difficult to match equivalent notes between two different interpretations of the same score. Our goal is to represent melodies in a reduced format able to keep the main features of the melody. For this, the trees need to be pruned.

If a maximum tree depth level is established, when a label is upgraded from children that are below that level, then those children nodes are pruned in addition to the label propagation.

These are the propagation (and pruning when applicable) rules:

R1 Given a node with two children, if one of those children contains the same label as the brother of the father node, the other child is promoted. Thus, more melodic richness is represented with less tree depth.

R2 In case that all the children of a node have the same label, they are deleted and its label is placed in the father node. Thus, two equal notes are equivalent to just one with double duration (see [3] for justification).

R3 If one of the brothers is the result of applying R3 three or more times (it had originally at least one eighth of the duration of the other brothers, then the brother of greater original duration is chosen. Thus we avoid very short notes (adornment notes) having more importance than longer notes[1].

R4 When various nodes are equivalent in original duration or when promoting a note implies losing the other, the label of the left node is upgraded.

R5 Silences never have greater precedence than notes.

R6 In case that there is only one child (either because of the tree construction or by propagation) it is automatically upgraded.

We will illustrate how these rules perform in an example of a melody. In Fig. 5-left one measure with some notes with different durations is presented, and Fig. 5-right, shows the tree originally built for its representation.

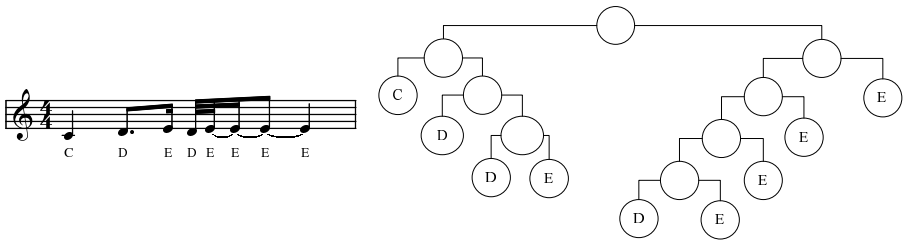

Fig. 5. One measure-melody and its tree representation with pitch labels (only in the leaves now) before pruning and label propagation.

In Fig. 6-left it can be observed how the propagation rules apply and prune the tree. In the first half of the melody, the labels E and A ascend by the rule R1. The second part shows how an adornment note is deleted by the application of the rules. The resulting tree corresponds to the score displayed in Fig. 6-right, that retains the perceptually important features of the melody. Once the tree has been pruned, the labels are propagated upwards, applying the same rules, without deleting nodes, until the roof in order to achieve a fully labelled tree.

4 Tree edit distance

We can define the edit distance between two trees like the minimum cost of the sequence of operations that transforms a tree into the other [1]. The edit operations are the same as those used in the string edition: deletion of a node, insertion, and substitution of the label of a node. In the insertion, a new node is

[1] the difference of one eighth of the duration has been established in an empirical way

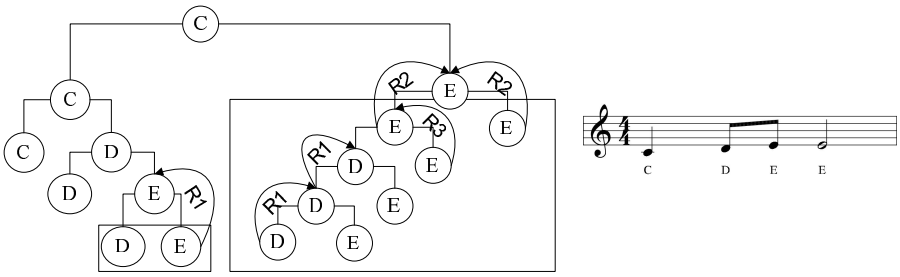

Fig. 6. Propagation of the leaf labels using the rules. The nodes into the rectangles disappear after pruning. The resulting melody is displayed in a score on the right.

added to the tree in a given point. The children of the node where the new node is inserted will become children of the new node. In the deletion, the children of the deleted node will become children of their previous grandfather node. The more similar the structure of the trees are, the less operations of deletion and insertion have to be done, and the smaller distance between them is achieved.

The deletion and insertion of nodes in a tree are not trivial matters, and it is necessary to understand the musical meaning of those actions. The important point is to note that the tree structure is closely related to the rhythmical structure of the melody.

5 Experiments and results

In our experiments the influence of different pitch representations on classification rates has been explored. Also, the application of prune rules and label propagation has been studied in relation with performance and error rates.

Three corpora made up with monophonic melodies have been used in our tests (in all cases only 8 measures have been taken from the melody start):

Real: built from 110 MIDI files fetched from Internet, it has 12 different classes (musical themes) from classic, jazz and pop/rock. The track containing the melody and the initial measure have been manually selected.

Latin: a synthetic database built from latin jazz melodies previously normalised which have been distorted with simulated human-made mistakes to obtain 3 more melodies of each. These melody distortions are based on small changes in both the note onset time and small errors in the pitch (e.g. errors like pressing the adjacent key instead of the right one in a keyboard). The original set had 40 melodies, and with the distorted melodies added we have obtained 160 melodies.

Classical: another synthetical database built using the same technique as above. The original set had 99 melodies, and the hole set has 393 melodies.

The weights used for the edit distance have (in all experiments) been set to 1 for insertion and deletion. For substitution, the weight is 0 if the interval/note is the same and 1 otherwise. Other tested weights did not improve the results.

The experiments with the three corpora have been made using the nearest neighbour rule and a leave-one-out scheme. Figure 7 shows the average error rates and time for the three corpora (tested separately). Experiments were run on a 750 MHz PC under Linux.

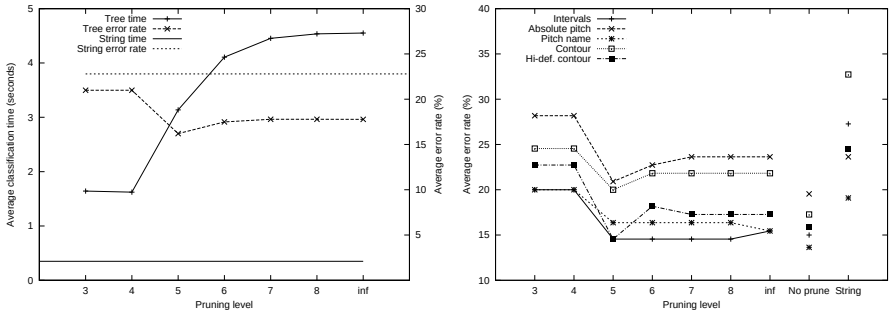

Fig. 7. Classification times and errors with different representations: (left) evolution of time and error rate versus tree pruning level (averaged for all the different labels). References for strings are plotted as horizontal lines. (right) error rates for the trees with the different labels. Errors for non-pruned trees and for strings are also displayed.

The performances for the five different kind of labels and for maximum tree levels ranging from 3 to 8, and without maximum level restriction (inf. in the graphs) were tested. Other experiments were to apply propagation without any pruning and the comparative performance of strings, coding both pitch and pitch plus duration sequences, as a reference.

The best error rate has been obtained with non pruned trees (see Table 1), but the high complexity of the distance calculation makes it very slow (38 s per sample in the 'real' corpus), making it unpractical. So we have focused in how much can we prune the trees keeping the error rates in a good level, always better that those for strings (see Fig. 7-left). A maximum level of 5 seems to be a good compromise between error and time.

In Fig. 7-right, the performance for the five codification labels is compared. The average errors for all the corpora are plotted for each kind of label. Note that the best results, apart from those obtained without pruning are obtained again for maximum level 5. Note also that trees perform better than strings.

Kind of labels	Pruned trees	Corpus	Non pruned trees	Strings
Absolute pitch	1.25	Latin	1.25	3.75
Pitch name	1.25	Latin	1.25	3.75
Contour	13.64	Real	11.87	12.48
Hi-def. contour	14.37	Latin	11.87	12.46
Interval	9.38	Latin	10.0	12.42

Table 1. Best tree classification error rates (in percentage) obtained for all the experiments, compared to those obtained for non pruned trees and strings.

6 Discussion and conclusions

Our results show that tree coding of melodies allows for better results than string coding. The addition of rhythmic information to string coding in order to improve classification rates is difficult, while tree coding naturally represents that information in its hierarchical structure.

Tree pruning has proved to be a good option in order to overcome the high time overhead of the tree edit distance, without significantly loosing classification accuracy. A maximum depth of 5 for pruning seems to be a good choice.

Preliminar experiments have been developed using polyphonic melodies and the results are promising, even better than those reported in this paper. We also plan to make use of the whole melody (not only 8 measures), developing some new methods for automatic extraction and segmentation of melodies.

Acknowledgements

This work has been funded by the Spanish CICYT project TAR; code TIC2000–1703–CO3–02.

References

1. Shasha S., Zhang K. Approximate Tree Pattern Matching. Pattern Matching Algorithms (1997) 341–371, Oxford University Press
2. Ó Maidin, D. A geometrical algorithm melodic difference. Computing in Musicology (1998) 65–72, MIT Press
3. Mongeau M., Sankoff D. Comparison of musical sequences. Computers and the Humanities **24** (1990) 161–175.
4. Smith L.A., McNab R.J., Witten I.H. Sequence-Based Melodic Comparison: A Dynamic-Programming Approach. Melodic Similarity. Concepts, Procedures, and Applications (1998) 1001–117. MIT Press and Center for Computing in the Humanities (CCARH), Stanford University
5. Lerdahl F. , Jackendoff R. A Generative Theory of Tonal Music (1983) MITP Cambridge, Massachusetts
6. Cruz-Alcázar P.P., Vidal-Ruiz E., Learning Regular Grammars to Model Musical Style: Comparing Different Coding Schemes. Proceedings of the 4th International Colloquium on Grammatical Inference (ICGI-98) LNAI **1433** (1998) 211-222.
7. Mitzenmacher M., Owen S. Estimating Resemblance of MIDI Documents. ALENEX (2001) 79–90.
8. Rizo D., Iñesta J.M. Tree-structured representation of melodies for comparison and retrieval Proc. of the Int. Workshop on Pattern Recognition in the Information Society (2002), 140–155.
9. Uitdenbogerd A.L., Zobel J. Manipulation of music for melody matching. In B. Smith and W. Eelsberg, editors, Proc. ACM International Multimedia Conference, Bristol, UK (1998) 235–240.
10. Kim Y. E., Chai W., Garcia R., Vercoe B. Analysis Of A Contour-Based Representation For Melody. Proc. International Symposium on Music Information Retrieval (2000).