Score Manipulation in PWGL using KSQuant

Kilian Sprotte$^1$, Mikael Laurson$^2$, and Mika Kuuskanka$^2$

$^1$ MKT, TU Berlin, Strasse des 17. Juni 135, 10623 Berlin, Germany
kilian.sprotte@gmail.com

$^2$ Center for Music and Technology, Sibelius Academy, P.O.Box 86, 00251 Helsinki, Finland
{laurson, mkuuskan}@siba.fi

Abstract. This paper presents the recent developments dealing with our metrical score representation format and how it can be used to produce and manipulate musical scores in PWGL. The base of these ideas is a special score format, called ENP-Score-notation, that allows to describe in textual form various score elements like beats, measures, voices and parts. This basic format can be extended using keywords to represent other score attributes such as pitch, enharmonics, expressions, and so on, resulting in complete scores. A novel PWGL library, KSQuant, uses a similar format: instead of the tree-based representation of the metric structure, KSQuant uses a flat onset-time representation. This format is ideal for many types of score manipulation operations where it is essential to preserve all score attributes.

1 Introduction

Many different text-based formats have been developed for representing the information in a musical score. Some typical examples are CMN, GUIDO, MusiXTEX [2], GNU LilyPond, and MusicXML [3]. Some Lisp-based representations are also able to preserve only a subset of musical information, such as the rhythm; these include the notation formats of both PatchWork [4] and OpenMusic [5]. Our textual format, ENP-Score-notation [7], also provides a way to generate and manipulate rich musical data in PWGL [6]. ENP-Score-notation is offered as an intermediate step between the score and the low-level file format. By using ENP-score-notation, a score can be converted into a more readable form that can also easily be converted back to an ENP score. This kind of symmetry is beneficial, as it allows users to view and modify score information without losing information.

The main topic of this paper is a novel PWGL user library, called KSQuant [1], developed by Kilian Sprotte. KSQuant is basically a quantizer, but unlike some of its predecessor libraries found in PatchWork, OpenMusic and PWGL (e.g. OMKant and GRhythm), it is strongly integrated in the PWGL system as it deals not only with rhythmic structures, but also with other score attributes.
Thus KSQuant is closely related to ENP-score-notation. The main difference between the systems is that whereas ENP-score-format deals normally with metrical structures, KSQuant is duration/onset-time oriented. This scheme allows KSQuant to be used as a high-level manipulation tool: it can realize besides quantizing tasks also other complex patch-level operations such as splitting and merging scores without losing the connection to other score parameters.

2 KSQuant Format

KSQuant operates with a format called 'simple-format'. Instead of using a tree-like structure like in ENP, events start at a certain time point and their duration is specified by the time that passes until the next event occurs. The ENP format is more expressive when it comes to the exact description of rhythmical structures – the simple-format can be seen as a possible reduction.

To this purpose, KSQuant provides mainly two conversion functions:

- **score2simple** transforms a given ENP-score into the simple-format. In order to do this, no additional parameters are required.
- **simple2score** is a much more complex function. It takes a lot of additional arguments to provide for information that is actually missing in the simple-format, such as time-signatures. Other parameters tune the quantization process that might need to be applied (Sect. 3).

![Fig. 1. A simple score using ENP that nevertheless demonstrates its ability to add notational attributes to a skeleton based on rhythm and pitch.](image)

As an example, let us consider the score in Fig. 1. It is quite minimal, but demonstrates some important properties of the ENP format. Concerning its rhythmical expressiveness, there is clearly more information inherent in the specific notation of the two triplets, than just a sequence of durations. Without changing them, the second triplet could be re-notated as a sixtuplet, or as two triplets with a tie in between. It also shows the use of some notational attributes. The accent is added to an entire chord, the x-note-head attribute affects individual notes only.

The textual description in ENP-score-notation format of this score is as follows:

```plaintext
(((1 ((1 :notes (67 60) :expressions (:accent))))
 (1 ((1 :notes (60)) (1 :notes (60)) (1 :notes (60))))
 (2 ((1.0 :notes (60)) (-1 :notes (60)))))
```
The tree-like structure used to represent the rhythmical groupings is clearly visible. When this score is converted using score2simple, the structure is translated to the following flatter representation in simple-format:

```
(((0.0 :notes (67 60) :expressions (:accent))
 (1.0 :notes (60))
 (1.333 :notes (60))
 (1.667 :notes (60))
 -3.0
 (4.0 :notes ((59 :note-head :x)))
 (4.667 :notes ((61 :note-head :x)))
 (5.333 :notes ((60 :note-head :x)))
 7.0))
```

The only rhythmical information that is left is the start-time of each chord, given as the first floating number in each list. A rest is represented by a negative start-time and at the end there is one additional time value that is exceptionally used to specify the end-time of the last chord.

On the other hand, all attribute information is exactly preserved as is. This allows the conversion processes to be reversible in the attribute domain, while being subject to change in the rhythmical domain in the case of simple2score.

3 KSQuant as a Quantizer

There has been a lot of ongoing research about Rhythm Quantization, especially in the context of Automatic Music Transcription [8][9]. Starting from the naive model of grid quantization, where each onset is quantized to the nearest grid point independent of its neighbours, more elaborate models have been developed that take the context into account. On the micro level the goal is then to successfully recognize expressive deviations, whereas tempo extraction on the macro level is performed.

KSQuant takes an intermediate position in this respect. Its quantization algorithm is currently not as sophisticated as it could be. The tempo, for instance, needs to be specified beforehand and can not automatically be extracted from the input. While there is certainly a potential for future work, it should be emphasized that the main goal of KSQuant is not to aid in Automatic Transcription, but to provide means for score transformations that are much more easily conducted on the simple-format representation than on the hierarchical tree representation.

---

3 technically, a single note object is also contained in a chord object
Nevertheless, the quantizer of KSQuant, as present in the function `simple2score`, features a number of parameters, whose default settings are shown in Fig. 2. Most important is the set of permitted divisions, `PermDivs` that is for convenience controlled by the parameters `MaxDiv` and `ForbiddenDivs`:

\[
PermDivs = \{d : d \in \{1, 2, ..., MaxDiv\}, d \notin ForbiddenDivs\}
\]

Together with the tempo, `PermDivs` establish a set of possible grids that is evaluated for every group (the group length depending on the time signature) by a simple heuristic employing the mean squared error of onset deviation in that group. The level of control provided by `PermDivs` often proves to be too limited. Therefore the user can still finetune the result using the argument `ForbiddenPatts`. Fig. 3 shows an example where the first measure is quantized using only `MaxDiv = 8` (i.e. `ForbiddenPatts` is nil). The second measure is, however, quantized with `ForbiddenPatts` set equal to `((1/16 3/32 3/32))` that forces the quantizer to change the second beat:

![Fig. 3. The effect of using the `ForbiddenPatt` argument.](image)

### 4 KSQuant Operations

This section enumerates some examples that demonstrates how KSQuant can be used to manipulate and build metrical scores. Often these manipulations would be tedious to perform using the standard ENP-score-notation format or other score building tools provided by PWGL.

Fig. 4 gives a typical quantizing patch example where start-times and pitch information is given in simple format to the `simple2score` box. The result is fed in turn to a `Score-Editor` box. Note that the simple format allows to specify the enharmonic spelling of the sixth note.
In Fig. 5, in turn, we have two input scores that are first merged to a single voice (to the left), and then appended resulting in a two measure score (to the right). We use here two KSQuant library boxes, simple-merge and simple-append, that accept as inputs the simple format.

Our final example, shown in Fig. 6, manipulates an input score by scaling all offset-times by 5/8 (see the scale input of the simple2score box). Note that also here the micro-tonal pitch material is preserved.
Fig. 6. An input score is scaled by 5/8 resulting in a new score.

5 ACKNOWLEDGEMENTS

The work of Mikael Laurson and Mika Kuuskankare has been supported by the Academy of Finland (SA 105557 and SA 114116).

References

1. Kilian Sprotte: KSQuant, a library for score manipulation and quantization http://kiliansprotte.de/ksquant/