Next Generation Performance Rendering — Exploiting Controllability

Keiji Hirata
NTT Communication Science Research Lab.
hirata@brl.ntt.co.jp

Rumi Hiraga
Bunkyo University
rhiraga@nefertiti.brl.ntt.co.jp

Abstract

We believe that the next-generation performance rendering system should be able to refine and improve a generated performance interactively, incrementally and locally through direct instructions in the natural language of a musician. In addition, the generated performance must reflect the musician's intention properly. For these purposes, we propose a new framework called two-stage performance rendering. The first stage translates a musician's instruction in natural language into the deviations of the onset time, duration and amplitude of structurally important notes and second stage spreads the deviations over surrounding notes. We demonstrate sample sessions using a prototype system that contains a gouping editor and a performance rendering engine.

1 Introduction

This paper discusses controllability in performance rendering (PR) and reports a prototype system that we have implemented based on our new framework.

Consider the following scenario of a piano lesson. A tutor gives a student direct instructions, such as "more passionate, here" or "even brighter for this phrase", whenever he/she thinks it best to refine the student's performance. Ideally, the student follows the instructions and changes his/her performance accordingly. The tutor, after having listened to the refined performance, may then issue other instructions. Observations of this scenario have given the authors the idea that a PR system should work like the tutor and student in the piano lesson, where the musician becomes the tutor, and the PR system is the student. That is, it should be possible to refine and improve a generated performance interactively, incrementally and locally through direct instructions in the natural language of the musician. The authors think that the solution is controllability.

To achieve a high level of controllability, we assert that a PR system should be able to (a) provide a user interface that allow a musician to specify how certain parts of a generated performance should be modified, (b) properly interpret the musician's instructions and (c) synthesize a natural performance that reflects these instructions.

For (a), the user interface has to give a feeling of direct manipulation to a musician. For (b), since instructions given in a natural language are usually subjective, equivocal, and even time-varying. the system should be able to be customized or personalized and be context-sensitive. As for (c), let us suppose a case in which a musician gives an instruction to play note Q louder in a particular part of a piece. If the system naively increases only the amplitude of Q, the generated performance may become unnatural. Considering the role of Q in the piece, the surrounding notes should also be played either louder or softer and even their agogics may have to be adjusted. Thus, to keep a generated performance natural, a PR system must maintain a certain musical consistency, which is represented in the form of the constraints regarding the agogics and dynamics for Q and the surrounding notes.

To meet these three requirements, this paper proposes a new framework called *two-stage performance rendering*.

2 Conventional Systems and Problems

Almost all conventional PR systems lack controllability, unfortunately. Given a score, these systems calculate the agogics and dynamics of all notes in the score all at once using rules, mathematical expressions and/or cases (which we call performance knowledge as a whole), extracted from real sample performances (S-performances for short) beforehand or taken from research results in musicology [Widmer 1993, Friberg 1991, Arcos et al. 1997, Igarashi et al. 2000]. In these systems, the relationships between the S-performances for extracting performance knowledge beforehand and the generated output is unclear. That is, a musician can hardly know which S-performances should be used to achieve a desired output; it is almost impossible to select appropriate S-performances that will leave some parts unchanged and modifying others. Thus, these systems cannot realize a cycle including the feedback of listening to the output and modification of a certain parts of the output that a musician does not prefer.

Another problem is that these systems cannot interpret the musician's subjective and qualitative in-

structions in the manner that the student in the piano lesson does. Such instructions are interpreted differently musician by musician. Hence, it is almost impossible to build a universal body of performance knowledge for generating a desired output.

3 Two-Stage Performance Rendering

3.1 Overview

In the two-stage performance rendering (Fig. 1), the first stage translates a musician's instruction into the agogics and dynamics of structurally important notes in a range and the second stage adjusts the surrounding notes. Here, a structurally

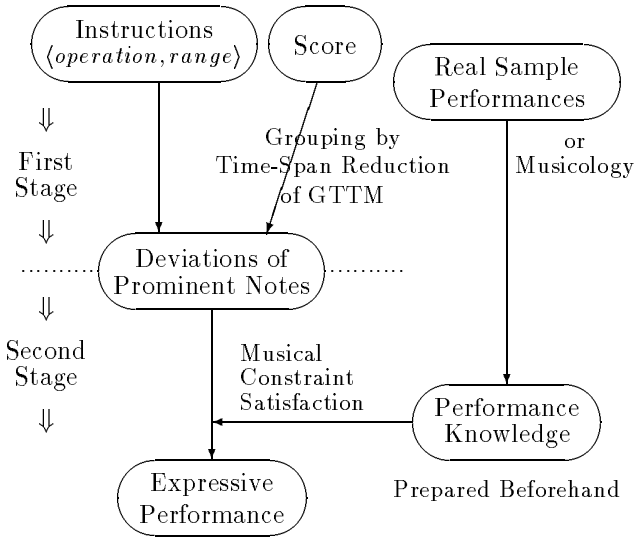

Figure 1: Framework of Two-Stage Performance Rendering

important note means a salient or prominent note in the context of the time-span (TS) reduction of GTTM [Lerdahl and Jackendoff 1983].

The inputs for two-stage performance rendering are a score to be performed, instructions given by the musician, and S-performances for extracting performance knowledge, which may be substituted with built-in rules or mathematical expressions derived from musicology. The output is an expressive performance of the score with the instructions issued. The musician's instructions specify an operation and the range of the score to which the operation is applied. The operations include faster, brighter, more passionate and so on.

The first stage maps the musician's subjective instructions in a natural language to the deviations of onset time, duration and amplitude (velocity) for every prominent note. The mapping is ad hoc and may be given in advance by a default heuristics of the PR system, manually provided by a musician, or acquired by a learning method for each musician and situation. On the other hand, notes included in a score are grouped hierarchically according to the TS reduction. Then, the reduction identifies prominent notes in groups at every level. The range of an instruction is mapped to the span of a group.

The second stage propagates the deviations set up to prominent notes at the first stage to their surrounding notes. At that time, the deviations of prominent notes are unchanged, and only agogics and dynamics of the surrounding notes are adjusted. This stage is introduced to bring about a musically natural performance. The performance knowledge for the propagation should be obtained in advance, and may be acquired from the analysis of S-performances with some musical theories, such as GTTM and I-R model [Narmour 1990]. The performance knowledge used here can be considered a constraint regarding the agogics and dynamics for prominent notes and their surrounding ones.

The two-stage performance rendering can be intuitively understood from the observation that trained musicians usually pay more attention to prominent notes than to the surrounding ones during performance.

3.2 Advantages

The two-stage performance rendering greatly owes the structural decomposition of a score to Desain and Honing (1992). However, since the two-stage performance rendering adopts the TS reduction of GTTM, a prominent note becomes available, and it follows that a musician can control an output as desired. Thus, it has the following advantages.

- Grouping notes on a score by GTTM enables a PR system to accept the musician's instructions localized to a part of the score.
- the musician's subjective knowledge used at the first stage is isolated from the universal musical knowledge used for musical constraint satisfaction at the second stage.
- The mapping of the first stage enables a PR system to adapt the preference and individuality of a musician.

4 Prototype System and Sample Sessions

We implemented a prototype system that employs the two-stage performance rendering. The implementation consists of a grouping editor and a PR engine. The system covers the music genre of solo piano tunes.

The grouping editor has a simple GUI to manipulate groupings of notes (Fig. 2). A subsidiary branch α represents a subsidiary group, and a main branch β a main group. By attaching α to β through the grouping editor, a musician can easily generate a new group one level higher in the TS reduction tree γ. Since this tree represents the musician's interpretation of a score, the editor facilitates transfer of a musician's intention to

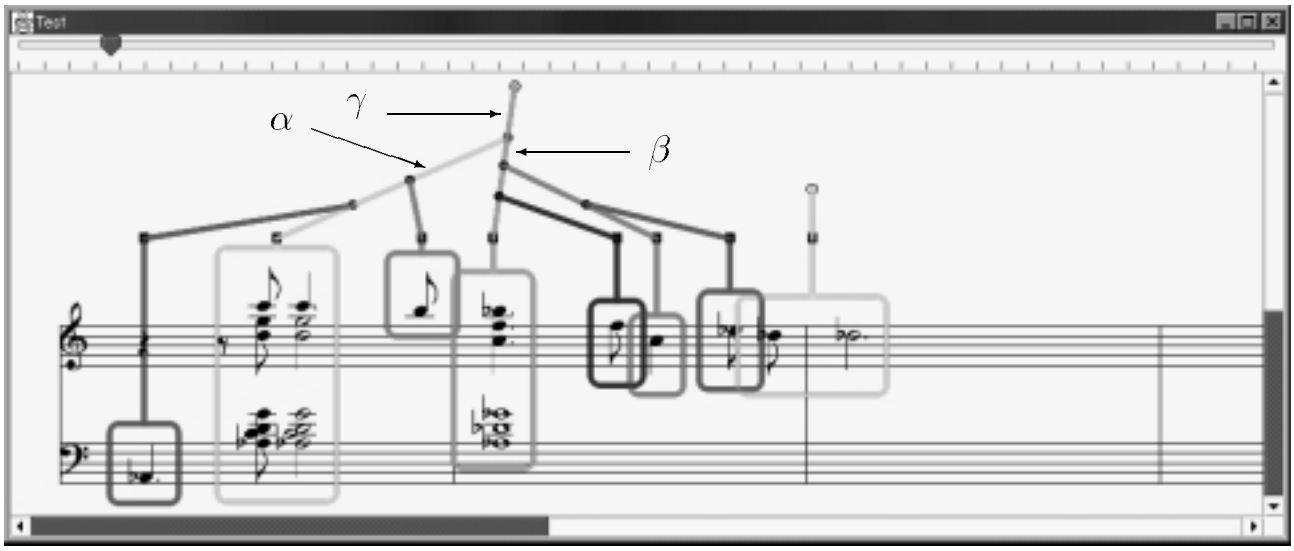

Figure 2: Grouping Editor Window

a PR system. Besides, since the editor is written in Java and can load and save a score and the associated grouping information in XML, the prototype system is highly portable.

In the current PR engine, for simplicity, the mapping of the first stage and the constraints of the second stage are given a priori; during operation, they do not change.

4.1 Sample 1: Vivace

Fig. 3 depicts how the system generates the output when a musician issues a sample instruction ⟨Vivace, bars 1 to 2⟩ on a score. The score has been analyzed based on the TS reduction beforehand; it contains three groups, α, β at a lower layer and γ at a higher layer, and α is subsidiary and β is primary. The prominent notes (chord) of α are **p**, those of β are **r**. Note that the results of the analysis are greatly dependent on a musician's intention and interpretation of the score and are not unique.

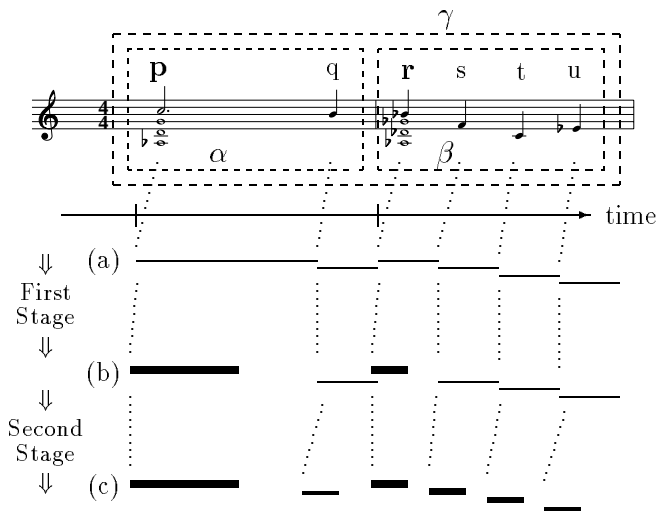

Figure 3: Sample 1: Vivace

In Fig. 3, (a), (b), and (c) represent the changes of onset time, duration, and amplitude of every note/chord included in the score fragment in the style of a piano roll (except for pitches); the horizontal axis represents time, and the amplitude of a note is represented by the thickness of a corresponding line segment. In the figure, (a) shows a mechanical (neutral) performance that follows the score exactly; (b) shows the situation after the first stage is finished and the deviations of only prominent chords of α and β are calculated; (c) shows the situation after the entire calculation is finished and the deviations of prominent chords are propagated to the surrounding notes.

In (b), since the instruction is vivace, the onset times of **p** and **r** are shifted forward by 10% of their original durations, and their durations are shorten by 40%. The values of 10% and 40% are not always chosen at the first-stage mapping; different values may be used for a different musician.

At the second stage, the deviations of **p** set in (b) are propagated to note q, and similarly, those of **r** to notes s, t and u. The process of the propagation is regarded as musical constraint satisfaction. Various performance knowledge for the musical constraint satisfaction can actually be used, and our prototype system assumes a uniform reduction of the durations of all surrounding notes by 60%. Meanwhile, in terms of onset time and duration, for simplicity, it assumes that the constraint is represented as a first-order polynomial function (Fig. 4). Consequently, the deviation of onset time for each note is proportional to the position of the note. Inversely, the amplitudes of notes

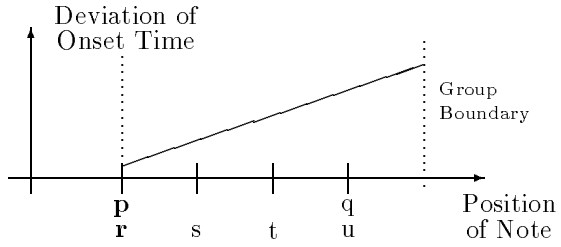

Figure 4: Function for Constraint

decrease proportionally to their positions.

Of course, the constraints for performance knowledge are not limited to such a simple function; more complicated expressions or algorithms can be adopted.

4.2 Sample 2: Graceful

Fig. 5 shows an example where a musician gives an instruction ⟨Graceful, bars 1 to 2⟩ on the same score fragment. Here, (a), (b) and (c) represent the same

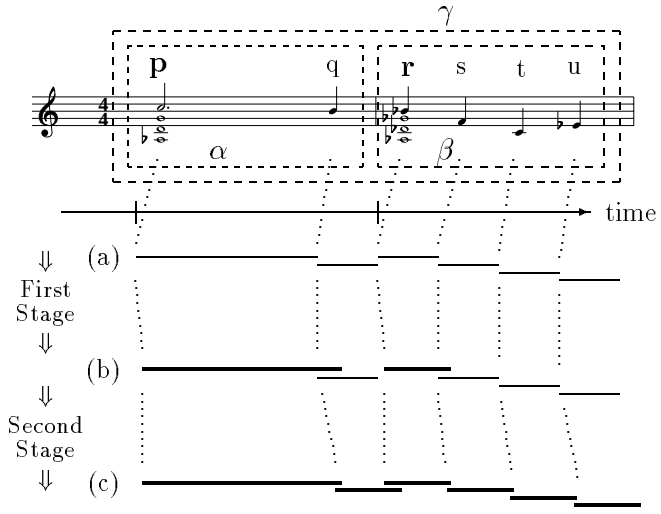

Figure 5: Sample 2: Graceful

snapshots. In order to realize a graceful performance, the durations of notes get longer throughout, and onset times are slightly shifted backward; these modifications correspond to the style of playing known as legato.

In (c), as for amplitude, the deviations of **p** and **r** are constantly propagated to the surrounding notes. As for onset time, a function similar to the previous sample (Fig. 4) is used.

5 Concluding Remarks

This paper proposed a framework for performance rendering with high controllability. Since this framework has a modular structure, to improve the quality of an output, we can separately examine various modules of the first-stage mapping, the second-stage performance knowledge, learning facilities for them and so on. We will consider the other combinations of these modules besides our current implementation.

A note usually belongs to more than one group, and it is in general unclear and more or less arbitrary to what extent what groups of different layers should contribute to generate the output. Thus, future work on the two-stage performance rendering will include development of:

- a learning method capable of adapting to an individual style and preferences for the first stage,
- an analysis method capable of discriminating between the first stage and the second stage knowledge extracted from real sample performances, and
- a method of combining the deviations of the layered groups.

References

[Arcos et al. 1997] Arcos, J. L., de Mántaras, R. L., and Serra, X. 1997. "SaxEx: a case-based reasoning system for generating expressive musical performances." *Proc. of ICMC*, pp.329–336. International Computer Music Association.

[Desain and Honing 1992] Desain, P., and Honing, H. 1992. "Towards a Calculus for Expressive Timing in Music." *Music, Mind and Machine*, pp.173–214. Amsterdam: Thesis Publishers.

[Friberg 1991] Friberg, A. 1991. "Generative Rules for Music Performance: A Formal Description of a Rule System." *Computer Music Journal* (15)2:56-71. The MIT Press.

[Igarashi et al. 2000] Igarashi, S., Koike, H., and Mizutani, T. 2000. "Structural Functions of Music and Creation of Interpretation Based on Them." 14th Annual Convention of Japanese Society for Artificial Intelligence (In Japanese).

[Lerdahl and Jackendoff 1983] Lerdahl, F. and Jackendoff, R. 1983. *A Generative Theory of Tonal Music.* The MIT Press.

[Narmour 1990] Narmour, E. 1990. *The Analysis and Cognition of Basic Melodic Structures.* The Univ. of Chicago Press.

[Widmer 1993] Widmer, G. 1993. "Understanding and Learning Musical Expression." *Proc. of ICMC*, pp.268–275. International Computer Music Association.