ORIGINAL PAPER

Statistical piano reduction controlling performance difficulty

EITA NAKAMURA[1] AND KAZUYOSHI YOSHII[1,2]

We present a statistical-modeling method for piano reduction, i.e. converting an ensemble score into piano scores, that can control performance difficulty. While previous studies have focused on describing the condition for playable piano scores, it depends on player's skill and can change continuously with the tempo. We thus computationally quantify performance difficulty as well as musical fidelity to the original score, and formulate the problem as optimization of musical fidelity under constraints on difficulty values. First, performance difficulty measures are developed by means of probabilistic generative models for piano scores and the relation to the rate of performance errors is studied. Second, to describe musical fidelity, we construct a probabilistic model integrating a prior piano-score model and a model representing how ensemble scores are likely to be edited. An iterative optimization algorithm for piano reduction is developed based on statistical inference of the model. We confirm the effect of the iterative procedure; we find that subjective difficulty and musical fidelity monotonically increase with controlled difficulty values; and we show that incorporating sequential dependence of pitches and fingering motion in the piano-score model improves the quality of reduction scores in high-difficulty cases.

I. INTRODUCTION

Music arrangement involving a change of instrumentation (e.g. arrangement for piano, guitar, etc.) is an important process of music creation to increase the variety of music performances. Arranging a musical piece to change difficulty, for example, to make it playable for beginners, is also widely practiced. To automate these processes, systems for piano arrangement [1–5], guitar arrangement [6–8], and orchestration [9, 10] have been studied. This study aims at a system for piano reduction, i.e. converting an ensemble score (e.g. orchestral and band scores) into a piano score that can control performance difficulty and retain as much musical fidelity to the original score as possible (Fig. 1).

To computationally judge whether a musical score is playable, previous studies have developed conditions on the pitch and rhythmic content. For piano scores, conditions such as "there can be at most five simultaneous notes for each hand" and "simultaneous pitch spans for each hand must be <14 semitones (or so)" have been considered [2, 3]. However, these conditions cannot be thought of as necessary nor sufficient conditions for playable scores because in reality there can be a piano score with chords with more

than five notes and/or spanning a large pitch interval that are conventionally played as broken chords, and even scores without chords (melodies) can be unplayable in fast tempos. In fact, it is difficult to find a complete description of playable scores that is valid in every situation because the condition depends on player's skill and can change continuously with the tempo. A possible solution is to quantify performance difficulty and use it as an indicator of playable scores in each situation of skill level, tempo, etc. [11, 12].

As there is generally a trade-off between performance difficulty and musical fidelity to the original score, it is necessary to quantify musical fidelity and develop an optimization method. Music arrangers remove notes and shift pitches in an ensemble score for the piano reduction score to match a target difficulty level [2, 4]. From a statistical point of view, one can assign probabilities for these edit operations and use them to quantify musical fidelity. Following the analogy with statistical machine translation [13], if one can construct a model for the probability $P(R|E)$ of a reduction score R given an ensemble score E, the piano reduction problem can be formulated as optimization of $P(R|E)$ under constraints on difficulty values. A similar approach without controls of performance difficulty has been studied for guitar arrangement [7, 8].

To realize this idea, a statistical-modeling approach for piano reduction that can control performance difficulty has been proposed in a recent conference paper [4]. Following the thought that fingering motion is closely related to

[1]Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
[2]RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan

Corresponding author:
Eita Nakamura
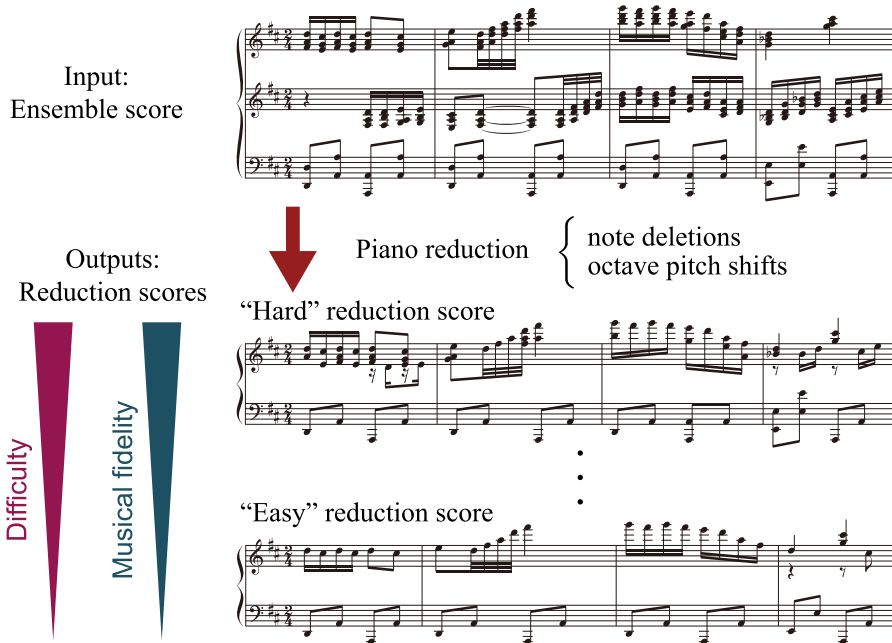Email: enakamura@sap.ist.i.kyoto-u.ac.jp

Fig. 1. Overview of the proposed system for piano reduction that can control performance difficulty.

the cost or difficulty of performance [14–16], quantitative measures of performance difficulty were developed based on a probabilistic generative model of piano scores incorporating fingering motion [17, 18]. To estimate the probability $P(R|E)$, a hidden Markov model (HMM) integrating the piano-score model and a model representing how ensemble scores are likely to be edited was constructed. A piano reduction algorithm was developed based on the Viterbi algorithm. While the potential of the method was suggested by the results of piano reduction for one example piece, formal evaluations, and comparisons with other approaches were left for future work. There was also a problem of the optimization method that the upper bound constraints on difficulty values were often not properly satisfied, due to the limitation of the Viterbi algorithm.

In this study, we extend the work of [4] and propose an improved piano reduction method using iterative optimization. We also carry out systematic evaluations on the difficulty measure and the piano reduction method. In particular, we evaluate difficulty measures in terms of their ability of predicting performance errors, which is to our knowledge the first attempt in the literature to objectively evaluating performance difficulty measures. Piano reduction methods are evaluated both objectively and subjectively: an objective evaluation is conducted to examine the effect of the iterative optimization strategy; an subjective evaluation is conducted to assess the quality of the generated reduction scores. The main results are:

- The proposed difficulty measures can be used as indicators of performance errors and measures incorporating the sequential nature of piano scores can better predict performance errors.

- The proposed iterative optimization method yields better controls of difficulty than the method in [4].
- Both subjective difficulty and musical fidelity of generated reduction scores monotonically increase with controlled difficulty values.
- By comparing methods based on different models, it is shown that incorporating sequential dependence of pitches and fingering motion in the piano-score model improves musical naturalness and the rate of unplayable notes of reduction scores in high-difficulty cases.

The following are limitations of the current system:

- Melodic and bass notes are manually indicated.
- Score typesetting, especially estimation of voices within each hand part, is currently done manually.

Automating these processes is an undeniable direction for future work. See Section IV.D for discussions.

The rest of the paper is organized as follows. In the next section, we discuss generative piano-score models and performance difficulty measures. In Section III, we present our method for piano reduction. In Section IV, we present and discuss results of evaluation of the piano reduction method. We conclude the paper in the last section.

II. QUANTITATIVE MEASURES OF PERFORMANCE DIFFICULTY

We formulate quantitative performance difficulty measures based on probabilistic generative models of piano scores. A generative model incorporating piano fingering and simpler models are described in Section II.A and performance difficulty measures are discussed in Section II.B.

A) Generative models for piano scores

1) Models for one hand

Let us first discuss models for one hand. A piano score is represented as a sequence of pitches $p_{1:N} = (p_n)_{n=1}^N$ and corresponding onset times $t_{1:N} = (t_n)_{n=1}^N$ (N is the number of musical notes). A generative model for piano scores (piano-score model) is here defined as a model that yields the probability $P(p_{1:N})$.

Simple piano-score models can be constructed based on the Markov model. The probability $P(p_{1:N})$ is factorized into an initial probability $P(p_1)$ and the transition probabilities $P(p_n|p_{n-1})$ as

$$P(p_{1:N}) = P(p_1) \prod_{n=2}^N P(p_n|p_{n-1}). \quad (1)$$

The simplest model is obtained by assuming that the initial and transition probabilities obey a uniform distribution over pitches. Writing $N_p = 88$ for the number of possible pitches, the model yields $P(p_{1:N}) = (1/N_p)^N$. Since this model yields the same probability for any piano score of the same length, it is here called a *no-information model*.

A more realistic model can be build by incorporating sequential dependence of pitches. For example, a statistical tendency called pitch proximity, that successive pitches tend to be close to each other, can be incorporated in initial and transition probabilities described with Gaussians:

$$P(p_1 = p) \propto Gauss(p; p_0, \sigma_p^2) + \epsilon, \quad (2)$$

$$P(p_n = p \mid p_{n-1} = p') \propto Gauss(p; p', \sigma_p^2) + \epsilon. \quad (3)$$

Here, $Gauss(\cdot; \mu, \sigma^2)$ denotes a Gaussian distribution with mean μ and standard deviation σ, p_0 is a reference pitch to define the initial probability, and ϵ is a small positive constant for smoothing the probability for pitch transitions with a large leap. We call this model a *Gaussian model*.

Although the Gaussian model can capture the tendency of pitch proximity, the simplification can lead to unrealistic consequences. First, pitch transitions involving 10 or 11 semitones have higher probabilities than octave motions, which opposes the reality [18]. Second, since the model does not distinguish white keys and black keys, it yields the same probability for piano scores transposed to any keys, which opposes the fact that "simpler keys" involving less black keys are more frequently used. In general, the difficulty or naturalness of a piano score changes when it is transposed to another key since the geometry of the piano keyboard requires different fingering motions [11]. To solve this, it is necessary to construct a model that describes fingering motions in addition to pitch transitions.

A model (called *fingering model*) incorporating fingering motions and the geometry of the piano keyboard has been proposed in [18]. The model is based on HMM, which has been first applied to the piano fingering model in [17]. In general, we can introduce a stochastic variable f_n representing a finger used to play the nth note. The variable f_n takes one of the following five values: 1 = thumb, 2 = index finger, . . . , 5 = little finger.[1] According to the model (Fig. 2), a fingering motion $f_{1:N} = (f_n)_{n=1}^N$ is first generated by an initial probability $P(f_1)$ and transition probabilities $P(f_n|f_{n-1})$. Next, a pitch sequence $p_{1:N}$ is generated conditionally on $f_{1:N}$: the first pitch is generated by $P(p_1|f_1)$ and the succeeding pitches are generated by $P(p_n|p_{n-1}, f_{n-1}, f_n)$, which describes the probability that a pitch would appear following the previous pitch and the previous and current fingers. Thus, the joint probability of pitches and fingering motion $P(p_{1:N}, f_{1:N})$ is given as

$$P(f_1)P(p_1|f_1) \prod_{n=2}^N P(f_n|f_{n-1})P(p_n|p_{n-1}, f_{n-1}, f_n).$$

In general, the parameters of the fingering model can be learned from music data with pitches and annotated fingerings. For want of a sufficient amount of data, the probability $P(p_n|p_{n-1}, f_{n-1}, f_n)$, which have $88^2 \cdot 5^2$ parameters, cannot be trained effectively in a direct way. We thus introduce simplifying assumptions to reduce the number of parameters. First, we assume that the probability depends on pitches through their geometrical positions on the keyboard (Fig. 2). The coordinate on the keyboard of a pitch p is represented as $\boldsymbol{\ell}(p) = (\ell_x(p), \ell_y(p))$. We also assume translational symmetry in the x-direction and time-inversion symmetry, which is expressed as

$$P(p_n = p \mid p_{n-1} = p', f_{n-1} = f', f_n = f)$$
$$= F(\ell_x(p) - \ell_x(p'), \ell_y(p) - \ell_y(p'); f', f)$$
$$= F(\ell_x(p') - \ell_x(p), \ell_y(p') - \ell_y(p); f, f'). \quad (4)$$

We also assume reflection symmetry between left and right hands. The above model can be extended to including chords, by sequencing the contained notes from low pitch to high pitch [16]. With the fingering model, one can estimate the fingering $f_{1:N}$ from a given sequence of pitches $p_{1:N}$ by calculating the maximum of the probability $P(f_{1:N}|p_{1:N}) \propto P(p_{1:N}, f_{1:N})$. This maximization can be computed by the Viterbi algorithm [19].

2) Models for both hands

A piano-score model with the left- and right-hand parts can be obtained by first constructing a model for each hand part and then combining the two models. If musical notes are already assigned to two hand parts, such a combined model can be obtained directly. On the other hand, if the part assignment is not given, as in the piano reduction problem, the model should be able to describe the probability for all cases of part assignment.

Such a model for piano music with unknown hand parts can be constructed based on the merged-output HMM [18, 20]. The idea is to combine outputs from two component Markov models or HMMs, respectively, describing the two hand parts. We here describe a model combining two fingering models. First, the hand part (left or right) associated

---

[1] In this study, we do not consider the possibility of finger substitutions where two or more fingers are assigned to a note.
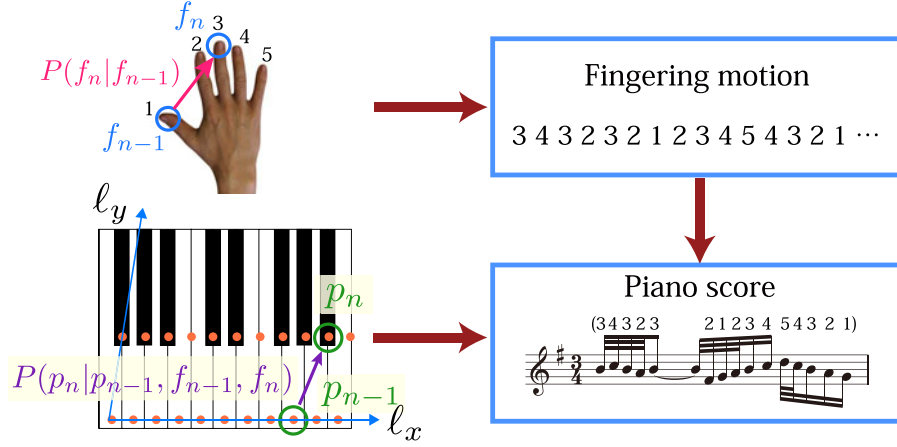
Fig. 2. Piano-score model incorporating fingering motion.

with a note p_n is represented by an additional stochastic variable $\eta_n \in \{L, R\}$. The generative process of η_n is described with a Bernoulli distribution: $P(\eta_n = \eta) = \alpha_\eta$ with $\alpha_L + \alpha_R = 1$. If η_n is determined, then the pitch is generated by the corresponding component model. For each $\eta \in \{L, R\}$, let $a^\eta_{f'f} = P^\eta(f|f')$ and $b^\eta_{f'f}(p', p) = F^\eta(\ell(p) - \ell(p'); f', f)$ denote the fingering and pitch transition probabilities of the component model. This process can be described as an HMM with a state space indexed by $k = (\eta, f^L, p^L, f^R, p^R)$ with the following initial and transition probabilities:

$$P(k_n = k \mid k_{n-1} = k')$$

$$= \begin{cases} \alpha_L a^L_{f'^L f^L} b^L_{f'^L f^L}(p'^L, p^L) \delta_{f'^R f^R} \delta_{p'^R p^R}, & \eta = L; \\ \alpha_R a^R_{f'^R f^R} b^R_{f'^R f^R}(p'^R, p^R) \delta_{f'^L f^L} \delta_{p'^L p^L}, & \eta = R, \end{cases}$$

$$P(p_n = p \mid k_n = k) = \delta_{pp^\eta}, \tag{5}$$

where δ denotes Kronecker's delta.

Using this model, one can estimate the sequence of latent variables $k_{1:N}$ from a pitch sequence $p_{1:N}$. This can be done by maximizing the probability $P(k_{1:N} \mid p_{1:N}) \propto P(k_{1:N}, p_{1:N})$. The most probable sequence $\hat{k}_{1:N}$ has the information of the optimal configuration of hands $\hat{\eta}_{1:N}$, which yields separated two hand parts and the optimal fingering for both hands ($\hat{f}^L_{1:N}$ and $\hat{f}^R_{1:N}$). For more details, see [18]. The Gaussian model and the no-information model can be similarly extended to models for both hands.

B) Performance difficulty

1) DIFFICULTY MEASURES

One can define a quantitative measure of performance difficulty based on the cost of music performance. From the statistical viewpoint, a natural choice is the probabilistic cost, which is the negative logarithm of a probability. To include the dependence on tempo, we define a performance difficulty as the time rate of the probabilistic cost

$$D(t) = -\ln P(\boldsymbol{p}(t))/\Delta t. \tag{6}$$

Here, Δt is a time width, $\boldsymbol{p}(t)$ is the sequence of pitches in the time range $[t - \Delta t/2, t + \Delta t/2]$, and $P(\boldsymbol{p}(t))$ is defined with one of the piano-score models in Section II.A. With the fingering model, one can use the joint probability of pitches and fingering to define a difficulty measure [18]:

$$D(t) = -\ln P(\boldsymbol{p}(t), \boldsymbol{f}(t))/\Delta t, \tag{7}$$

where $\boldsymbol{f}(t)$ denotes the fingering corresponding to the pitches $\boldsymbol{p}(t)$. If the fingering is unknown, one can substitute the maximum-probability estimate $\hat{\boldsymbol{f}}(t)$ in equation (7). For each note n with onset time t_n, we write $D(n) = D(t_n)$.

The difficulty measure can be defined for each hand part using the pitches in that hand part and a piano-score model for one hand, which is denoted by $D_L(t)$ or $D_R(t)$. In addition, the total difficulty can be defined as the sum of difficulties for both hands: $D_B(t) = D_L(t) + D_R(t)$. The quantity $D_B(t)$ can be relevant as well as $D_L(t)$ and $D_R(t)$ since the difficulty can be high even if difficulties for individual hand parts are not so high.

In previous studies [11, 12], features such as playing speed, note density, pitch entropy, hand displacement rate, hand stretch, and fingering complexity have been considered to estimate the difficulty level of piano scores. These features are incorporated in the above difficulty measures, although in an implicit manner. If one uses the no-information model, the difficulty measure takes into account the note density and playing speed. With the Gaussian model, pitch entropy and hand displacement rate, and hand stretch are incorporated in addition. With the fingering model, fingering complexity is further incorporated.

2) EVALUATION

To formally examine how the proposed measures reflect real performance difficulty, we study their relation with the rate of performance errors. We use a dataset [21] consisting of 90 MIDI piano performance signals of 30 classical musical pieces; for each piece, there are performances by three different players that are recorded in international
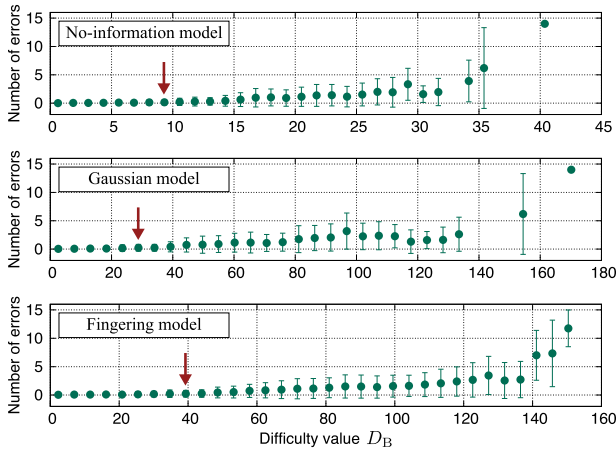
Fig. 3. Relations between the difficulty value D_B and the number of performance errors. Points and bars indicate means and standard deviations. Arrows indicate onsets of performance errors (see text).

Table 1. Accuracies of performance error prediction.

Model	Threshold $(D_R^{th}, D_L^{th}, D_B^{th})$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_w$	$\mathcal{P}_w$	$\mathcal{R}_w$
No-information	$(9, 10, 14)$	52.4	43.0	67.1	69.8	63.0	78.1
Gaussian	$(30, 30, 42)$	54.2	46.3	65.2	71.3	66.4	77.0
Fingering	$(41, 39, 53)$	53.9	49.1	59.8	70.6	69.3	73.8

Since more frequent errors indicate larger difficulty, we can also define the following weighted quantities:

$$\mathcal{P}_w = \frac{N'_{TP}}{N'_{TP} + N_{FP}}, \quad \mathcal{R}_w = \frac{N'_{TP}}{N'_{TP} + N'_{TN}}, \quad (8)$$

$$\mathcal{F}_w = \frac{2\mathcal{P}_w \mathcal{R}_w}{\mathcal{P}_w + \mathcal{R}_w}, \quad (9)$$

where N'_{TP} and N'_{TN} are obtained by weighting N_{TP} and N_{TN} with the number of performance errors. The results are shown in Table 1 where the thresholds are optimized with respect to $\mathcal{F}_w$ for each model. We see that the Gaussian model has the highest F measures, even though the differences are rather small. A possible reason is the relatively small size of the data used for training the fingering model. Since the Gaussian model has only one parameter σ_p to train, it has better generalization ability for such small training data. Such a trade-off between model complexity and the required amount of training data is common in many machine-learning problems. We thus use the difficulty measures defined with the Gaussian model in the following.

III. PIANO REDUCTION METHOD

In the statistical formulation of piano reduction, we try to find the optimal reduction score $\hat{R}$ that maximizes the probability $P(R|E)$ for a given ensemble score E. In analogy with the statistical approach for machine translation [13], we first construct generative models describing the probability $P(R)$ and $P(E|R)$, respectively, and integrate them for calculating $P(R, E) \propto P(R|E)$. We then derive optimization algorithms for piano reduction that take into account the constraints on performance difficulty values.

Prior to the main processing step, we convert an input ensemble score to a condensed score by removing redundant notes with the same pitch and simultaneous onset time (the number of such redundant notes is memorized and used later in the calculation of equation (13)). Although they are different strictly, we call such a condensed score an ensemble score in what follows. What is really meant by the symbol E is also a condensed score.

A) Model for piano reduction

To construct a generative model that yields the probability $P(R, E)$, we integrate a piano-score model describing the probability $P(R)$ and an *edit model* that describes the process yielding $P(E|R)$. As a piano-score model, we can

piano competitions. In the dataset, musical notes in a performance signal are matched to notes in the corresponding score and the following three types of performance errors are manually annotated: pitch error (a performed note with a corresponding note in the score but with a different pitch); extra note (a performed note without a corresponding note in the score); and missing note (a note in the score without a corresponding note in the performance). Timing errors are not annotated in the data and not considered in this study.

We calculate performance difficulty values for each onset time and calculate the number of performance errors in the time range of width Δt around the onset time. In the following, we set Δt to be 1 s. For the Gaussian model, $\epsilon = 4 \times 10^{-4}$ and p_0 is C3 (C5) for the left (right) hand. Other parameters of the Gaussian and fingering models are taken from a previous study [18] where a different dataset was used for training.

Figure 3 shows the relation between difficulty value D_B and the rate of performance errors for the three models. We see that for each model there is an onset (roughly, 10 for the no-information model, 30 for the Gaussian, and 40 for the fingering model) below which the average number of errors is almost zero and above which it gradually increases. This suggests that the difficulty measures can be used as indicators of performance errors.

For comparative evaluation, we predict performance errors by thresholding the difficulty values and measure the predictive accuracy for the three models. Using three thresholds D_L^{th}, D_R^{th}, and D_B^{th}, a prediction of performance errors at time t is defined positive if one of three conditions ($D_L^{th} > D_L(t)$, $D_R^{th} > D_R(t)$, and $D_B^{th} > D_B(t)$) is satisfied. We calculate the number of true positives N_{TP}, that of false positives N_{FP}, and that of true negatives N_{TN}, and the following quantities are used as evaluation measures:

$$\mathcal{P} = \frac{N_{TP}}{N_{TP} + N_{FP}}, \quad \mathcal{R} = \frac{N_{TP}}{N_{TP} + N_{TN}}, \quad \mathcal{F} = \frac{2\mathcal{P}\mathcal{R}}{\mathcal{P} + \mathcal{R}}.$$

use either the Gaussian model or the fingering model discussed in Section II.A.2, which statistically describes the naturalness of a generated (reduction) score.

For the edit model, we assign probabilities for edit operations applied to musical notes. As in [4], we focus on the two most common edit operations, note deletion, and octave pitch shift. As we model the inverse process of generating an ensemble score from a piano score, we introduce probabilities of note addition and octave pitch shift in the edit model. For each note in the ensemble score, the probability that it is an added note and not originated from the piano score is denoted by β_{NP} ("NP" for not played). In this case, the note's pitch p is drawn from a uniform distribution $c_{unif}(p)$. If it is originated from the piano score and the corresponding note has a pitch q, the probability of the note's pitch p denoted by $c_q(p) = P(p|q)$ is supposed to obey

$$c_q(p) = \begin{cases} 1 - 2\gamma_{oct}, & p = q; \\ \gamma_{oct}, & p = q \pm 12, \end{cases} \quad (10)$$

where γ_{oct} denotes the probability of an octave shift.

We can integrate the fingering model in Section II.A.2 and the edit model in the following fashion based on the merged-output HMM (Fig. 4), which leads to tractable inference algorithms. For each note in the output ensemble score, indexed by m, we introduce a stochastic variable ξ_m that can take one of three values $\{NP, L, R\}$. It is generated from a discrete distribution as $P(\xi_m = \xi) = \beta_\xi$, where parameters β_ξ obey $\beta_{NP} + \beta_L + \beta_R = 1$. If $\xi_m = NP$, then its pitch p_m has a probability $P(p_m = p) = c_{unif}(p)$. If $\xi_m = L$ or R, then its pitch is generated from the component fingering model of the corresponding hand part and may undergo an octave shift. Writing f^L, p^L, f^R, and p^R for the finger and pitch variables of the two component fingering models, the latent state of the merged-output HMM is described by a set of variables $r = (\xi, f^L, p^L, f^R, p^R)$. The transition and output probabilities are defined as

$$P(r_m = r \mid r_{m-1} = r')$$

$$= \begin{cases} \beta_{NP}\delta_{f^L f'^L}\delta_{f^R f'^R}\delta_{p^L p'^L}\delta_{p^R p'^R}, & \xi = NP; \\ \beta_L a^L_{f'^L f^L} b^L_{f'^L f^L}(p'^L, p^L)\delta_{f^R f'^R}\delta_{p^R p'^R}, & \xi = L; \\ \beta_R a^R_{f'^R f^R} b^R_{f'^R f^R}(p'^R, p^R)\delta_{f^L f'^L}\delta_{p^L p'^L}, & \xi = R, \end{cases}$$

$$P(p_m = p \mid r_m = r) = \begin{cases} c_{unif}(p), & \xi = NP; \\ c_{p^L}(p), & \xi = L; \\ c_{p^R}(p), & \xi = R. \end{cases} \quad (11)$$

The model indeed generates a piano score specified by $(p^L_m, p^R_m)_m$ and an ensemble score specified by $(p_m)_m$.
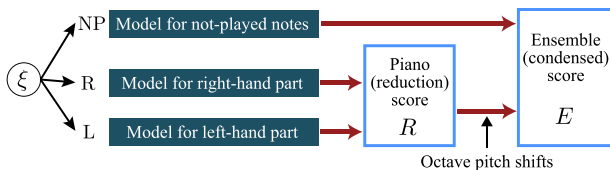

Fig. 4. Generative process of the model for piano reduction.

In the process of piano reduction, which is explained in the next section, the parameter β_{NP} represents how much notes in the ensemble score are removed. Thus, properly adjusting β_{NP} is crucial to control the performance difficulty of resulting reduction scores. Roughly speaking, if the note density around a note is high, it is necessary to remove more notes around that note by setting β_{NP} large. In addition, some notes like melodic notes and bass notes are musically more important than others and should have a small probability of deletion, or small β_{NP} in the present model. These conditions can be realized in the following form of $\beta_{NP}(m)$, which depends on each note m:

$$\beta_{NP}(m) = (1 - \zeta(m))e^{-\kappa h(m)}, \quad (12)$$

where $h(m) \geq 0$ represents the musical importance of note n, $\kappa > 0$ is a coefficient to control the effect of $h(m)$, and $\zeta(m) \in [0, 1]$ is a factor to control the overall rate of note deletion. If $\zeta(m) \simeq 1$, $\beta_{NP}(m) \simeq 0$ and almost all notes remain in the reduction score. If $\zeta(m) \simeq 0$, $\beta_{NP}(m) \simeq 1$ unless $\kappa h(m)$ is large (i.e. note m is musically important), so most musically unimportant notes will be removed.

In addition to the importance of melodic and bass notes, it is not difficult to imagine that pitches in an ensemble score that are played simultaneously by multiple instruments are musically important. Thus, the following form is used for defining musical importance $h(m)$:

$$h(m) = \mathbb{I}(m \in \mathcal{M}) + \mathbb{I}(m \in \mathcal{B}) + a\, Mult(m), \quad (13)$$

where $\mathbb{I}(\mathcal{C}) = 1$ if a condition $\mathcal{C}$ is true and 0 otherwise, $\mathcal{M}$ denotes the set of melodic notes, $\mathcal{B}$ denotes the set of bass notes, and $Mult(m)$ is the *multiplicity* of note m, defined as the number of notes in the ensemble score having the same pitch and onset time as note m excluding m itself. The parameters κ and a are adjustable parameters, and $\zeta(m)$ is adjusted according to target difficulty values as explained in the next section.

B) Algorithms for piano reduction

Let us derive algorithms for piano reduction based on the model in Section III.A and the difficulty measures in Section II.B. The piano reduction problem is here formulated as finding a reduction score R that maximizes $P(R|E)$ for a given ensemble score E with constraints on R's performance difficulty values. Specifically, we impose the following constraints for each note n in R:

$$[D_L(n) < \widetilde{D}_L] \wedge [D_R(n) < \widetilde{D}_R] \wedge [D_B(n) < \widetilde{D}_B], \quad (14)$$

where $\widetilde{D}_L$, $\widetilde{D}_R$, and $\widetilde{D}_B$ are some target difficulty values.

Without the constraints (14), finding the maximum of $P(R|E)$ is a basic inference problem for HMMs and can be achieved with the Viterbi algorithm [19]. However, the constraints (14) cannot be easily treated because difficulty values at each note depends on the existence of other notes in the time range of Δt, which violates the Markovian assumption for the Viterbi algorithm. In other words, if we know

appropriate values of $\zeta(m)$ in equation (12) for controlling difficulty values, the optimization problem is directly solvable, but finding those values is not easy.

In the following, we present two strategies for optimization. In a previous study [4], appropriate values of $\zeta(m)$ were estimated and the Viterbi algorithm was applied once to obtain the result. A slight extension of this *one-time optimization* method is presented in Section III.B.1. On the other hand, if one can apply the Viterbi algorithm iteratively, it would be possible to find appropriate values of $\zeta(m)$ from tentative results, by starting from $\zeta(m) = 1$ and gradually lessening it. This *iterative optimization* method is developed in Section III.B.2.

1) ONE-TIME OPTIMIZATION ALGORITHM

In [4], appropriate values of $\zeta(m)$ were estimated by matching the expected difficulty values to the target values with the following equation:

$$\zeta(m) = \min\left\{ \frac{\widetilde{D}_L}{D_L(m)}, \frac{\widetilde{D}_R}{D_R(m)} \right\}, \tag{15}$$

where $D_L(m)$, etc., represent the difficulty values calculated for the ensemble score at its mth note. One can include a factor involving $D_B(m)$ in the above equation in general.

We can generalize this method by introducing a scaling factor ρ and modifying equation (15) to

$$\zeta(m) = \rho \min\left\{ \frac{\widetilde{D}_L}{D_L(m)}, \frac{\widetilde{D}_R}{D_R(m)} \right\}. \tag{16}$$

By choosing the value of ρ, one can control the expected average of resulting difficulty values. For example, one can use a maximum value of ρ that can satisfy the constraints (14) for most outcomes.

2) ITERATIVE OPTIMIZATION ALGORITHM

For iterative optimization, the Viterbi algorithm is applied in each iteration to obtain a tentative reduction score $R^{(i)}$, with tentative values of $\zeta^{(i)}(m)$ (i is an index for iterations). For each note n in $R^{(i)}$, we calculate the difficulty values $D_L^{(i)}(n)$, $D_R^{(i)}(n)$, and $D_B^{(i)}(n)$. If the constraints (14) are not all satisfied at note n, then we lessen the values of $\zeta(m)$ for all notes m in the ensemble score around n within the time range of width Δt as

$$\zeta^{(i+1)}(m) = \lambda \zeta^{(i)}(m) \tag{17}$$

with some constant $0 < \lambda < 1$.

The iterative algorithm is initialized with $\zeta^{(i=1)}(m) = 1$ for all notes m. The algorithm ends when the constraints (14) are satisfied at every note in the reduction score, or the number of iterations exceed some predefined value i_{max}. For efficient and stable computation, the Viterbi algorithm at iteration $i + 1$ is applied only to those regions of the ensemble score where the constraints (14) are not still satisfied at iteration i. Specifically, we first construct a set of notes m in the ensemble score whose onset time t_m is included in the range $[t_n - \Delta t/2, t_n + \Delta t/2]$ around some onset time t_n in

the reduction score for which the difficulty constraints are not satisfied. This set is then split into a set Ψ of isolated regions of notes. For each such isolated region, the Viterbi algorithm is applied with fixed boundary states at one note before the beginning of the region and one note after the end.

The iterative algorithm is summarized as follows.

(i) Initialize $\zeta^{(i=1)}(m) = 1$ and apply the Viterbi algorithm to the whole ensemble score.
(ii) Calculate difficulty values and obtain regions Ψ where the constraints (14) are not satisfied. Exit if Ψ is empty or $i \geq i_{max}$.
(iii) Update the control factor $\zeta(m)$ as in equation (17) and apply the Viterbi algorithm to each region of Ψ. Increment i and go back to step (ii).

IV. EVALUATION OF PIANO REDUCTION ALGORITHMS

A) Setup

To evaluate the piano reduction algorithms, we prepared a dataset of orchestral pieces of Western classical music. The dataset consists of 10 pieces by different composers and with different instrumentations; each piece has a length of around 20 bars. The list of the pieces is available in the accompanying webpage.[2]

We compare one-time optimization algorithms and iterative optimization algorithms based on the Gaussian model and the fingering model; in total, we have four methods labeled as *One-time Gaussian*, *One-time Fingering*, *Iterated Gaussian*, and *Iterated Fingering* methods. The parameters of the piano-score models are set as in Section II.B.2. The other parameters are set as follows: $a = 0.01$, $\kappa = 11$, $\lambda = 0.85$, $\gamma_{oct} = 0.001$, and $\beta_R(m) = \beta_L(m) = (1 - \beta_{NP}(m))/2$ where $\beta_{NP}(m)$ is set as in equation (12). Difficulty values are calculated with the difficulty measures using the Gaussian model with $\Delta t = 1\,s$. These parameter values were fixed after some trials by one of the authors and there is room for further optimization.

As a baseline method, we also implement a method based on a simple piano-score model (called the *distance* model) that takes into account the distance between each note in the ensemble score and its closest melodic or bass note, but not sequential dependence of pitches. Specifically, for each note m in the ensemble score, the closest melodic or bass notes $CMB(m)$ is obtained by first searching in the direction of onset time and then in the direction of pitches. Then the probability of its pitch p_m is given as

$$P(p_m) \propto Gauss(p_m; CMB(m), \sigma_p^2). \tag{18}$$

Integrating this piano-score model into the piano reduction model in Section III.A and using the iterative optimization algorithm, a baseline *Iterated Distance* method is obtained.

---

[2]http://pianoarrangement.github.io/demo.html

B) Quantitative evaluation of difficulty control

We first examine the effect of the iterative optimization algorithms in controlling the difficulty values of output reduction scores, in comparison with the one-time optimization algorithms. We run the four algorithms, *One-time Gaussian*, *One-time Fingering*, *Iterated Gaussian*, and *Iterated Fingering*, for the test dataset with three sets of target difficulty values $(\widetilde{D}_L, \widetilde{D}_R, \widetilde{D}_B) = (15, 15, 30)$, $(30, 30, 40)$, and $(40, 40, 50)$. For the scaling factor ρ for the one-time optimization algorithms, we test values in $\{0.1, 0.2, \ldots, 1.0\}$. For the iterative optimization algorithms, i_{max} is set to 50. To evaluate a reduction score R, we compute difficulty values $(D_L(n)$ etc.) for each note n in R and calculate the following measures:

- Mean difficulty values $\{\overline{D}_L, \overline{D}_R, \overline{D}_B\}$:

$$\overline{D}_L = \frac{1}{\#R} \sum_{n \in R} D_L(n) \quad etc. \tag{19}$$

- Maximum difficulty values $\{D_L^{max}, D_R^{max}, D_B^{max}\}$:

$$D_L^{max} = \max_{n \in R}\{D_L(n)\} \quad etc. \tag{20}$$

- Out-of-range rate (proportion of regions where difficulty values exceed target values) $\{A_{out}^L, A_{out}^R, A_{out}^B\}$:

$$A_{out}^L = \frac{\#\{n \in R \mid D_L(n) > \widetilde{D}_L\}}{\#R} \quad etc. \tag{21}$$

- Additional-note rate (proportion of notes in the reduction score other than melodic and bass notes) A_{add}:

$$A_{add} = \frac{\#R - \#\mathcal{M} - \#\mathcal{B}}{\#\mathcal{M} + \#\mathcal{B}}. \tag{22}$$

Variations of difficulty values of the reduction scores by the *One-time Gaussian* method are shown in Fig. 5, with corresponding values for the *Iterated Gaussian* method. Here, for simplicity, only difficulty values for both hands
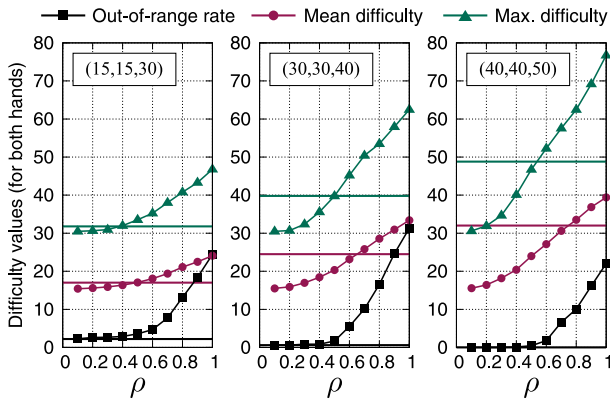
$(\overline{D}_B, D_B^{max}$, etc.) are shown. It is observed that for those values of ρ where A_{out}^B is equivalent to that for the iterative optimization method, $\overline{D}_B$ and D_B^{max} are smaller compared with the iterative optimization method. This means that with the same level of satisfaction for the difficulty constraints, results of the iterative optimization method have larger difficulty values on the average, which is a desired property. On the other hand, if ρ is increased sufficiently, it is possible for the one-time optimization algorithm to achieve the same level of $\overline{D}_B$ as the iterative optimization method, but then A_{out}^B is larger, meaning that the difficulty constraints are less strictly satisfied. Analyses of difficulty values for each hand and comparison between *One-time Fingering* and *Iterated Fingering* methods reveal similar tendencies.

The results for all three kinds of difficulty values (for each of two hands and for both hands) are shown in Table 2. Here, for one-time optimization methods, results are shown for the smallest value of ρ such that all three out-of-range rates exceed those for the corresponding iterative optimization methods. In addition to the same tendencies as found in the above analysis, one can observe that for the same level of satisfaction of difficulty constraints, the iterative optimization methods yield larger additional-note rates than the corresponding one-time optimization methods. These results indicate that the iterative optimization methods are more appropriate for controlling difficulty values.

Even for iterative optimization algorithms, the out-of-range rates can be non-zero, especially for small target difficulty values. One reason for this is that for some pieces the minimal reduction score with only melodic and bass notes has difficulty values larger than the target values. Another reason is the greedy-like nature of the iterative optimization algorithms: when some regions of the reduction score is fixed and used as boundary conditions for updates, the Viterbi search sometimes cannot reduce notes even for smaller values of $\zeta(m)$. Comparing iterative optimization methods in cases of target difficulty values $(30, 30, 40)$ and $(40, 40, 50)$, we find that while the *Iterated Gaussian* method has the largest additional-note rate, it has the least values for most difficulty evaluation measures. If the additional-note rate increases with the fidelity to the original ensemble score, this indicates the *Iterated Gaussian* method has the ability to efficiently increase the fidelity while retaining low difficulty values. This is probably because the Gaussian model is used for calculating difficulty measures.

C) Subjective evaluation

We conduct a subjective evaluation experiment to evaluate the quality of reduction scores by the proposed algorithms.[3] In particular, we examine how much of the additional notes (notes other than melodic and bass notes) are actually playable and how the musical quality such as fidelity and difficulty changes with varying target difficulty values. For

Fig. 5. Difficulty metrics for the *One-time Gaussian* method for varying ρ, for three cases of target difficulty values $(\widetilde{D}_L, \widetilde{D}_R, \widetilde{D}_B)$ indicated in the insets. Difficulty values are those for both hands $(\overline{D}_B, D_B^{max}$, etc.) and horizontal lines indicate corresponding values for the *Iterated Gaussian* method.

[3]Readers who wish to have access to the raw experimental data and source code should contact the authors.

Table 2. Comparison of average values of difficulty metrics for reduction scores. Triplet values in parentheses indicate one for left-hand part, right-hand part, and both hand parts, from left to right.

Algorithm	Target difficulty	Mean difficulty	Max. difficulty	Out-of-range rate (%)	A_{out} (%)
One-time Gaussian	(15, 15, 30)	(10.0, 5.4, 15.4)	(22.5, 14.6, 30.5)	(18.6, 7.3, 2.3)	7.1
Iterated Gaussian	(15, 15, 30)	(11.0, 6.1, 17.0)	(22.3, 15.5, 31.8)	(18.2, 7.2, 2.2)	20.3
One-time fingering	(15, 15, 30)	(12.9, 9.1, 22.0)	(30.7, 27.8, 50.5)	(30.0, 18.0, 20.9)	30.9
Iterated fingering	(15, 15, 30)	(12.7, 8.4, 21.1)	(29.0, 23.9, 46.5)	(27.5, 15.7, 14.9)	31.7
Iterated distance	(15, 15, 30)	(11.9, 6.1, 18.0)	(28.0, 15.7, 37.3)	(23.4, 7.4, 5.2)	21.8
One-time Gaussian	(30, 30, 40)	(10.4, 5.5, 15.9)	(23.2, 15.4, 30.7)	(0.7, 0, 0.6)	11.4
Iterated Gaussian	(30, 30, 40)	(16.2, 8.3, 24.5)	(30.0, 21.2, 39.8)	(0.4, 0, 0.6)	62.3
One-time fingering	(30, 30, 40)	(13.2, 9.4, 22.7)	(31.8, 28.6, 51.2)	(6.5, 5.8, 11.6)	33.4
Iterated fingering	(30, 30, 40)	(16.3, 10.6, 26.9)	(34.3, 28.6, 50.9)	(3.6, 3.0, 6.3)	60.1
Iterated distance	(30, 30, 40)	(17.8, 8.3, 26.0)	(35.9, 21.7, 44.8)	(2.4, 0, 2.3)	61.9
One-time Gaussian	(40, 40, 50)	(13.4, 7.0, 20.4)	(30.6, 19.2, 40.1)	(0.1, 0, 0.1)	39.0
Iterated Gaussian	(40, 40, 50)	(20.9, 11.1, 32.0)	(36.8, 27.8, 48.8)	(0, 0, 0)	98.3
One-time fingering	(40, 40, 50)	(13.5, 9.5, 22.9)	(32.4, 29.2, 51.7)	(2.8, 3.4, 5.7)	34.7
Iterated fingering	(40, 40, 50)	(20.2, 13.6, 33.8)	(40.1, 33.1, 54.9)	(1.7, 1.0, 1.6)	88.9
Iterated distance	(40, 40, 50)	(22.1, 10.3, 32.4)	(42.5, 27.3, 53.6)	(0.8, 0, 0.8)	88.3

this, we asked professional piano arrangers to evaluate the piano reductions generated by the *Iterated Fingering*, *Iterated Gaussian*, and *Iterated Distance* methods with three sets of target difficulty values (15, 15, 30), (30, 30, 40), and (40, 40, 50). Two music arrangers participated in the evaluation and each reduction score was evaluated by one of them. Evaluators are provided manually typeset reduction scores, the input condensed scores, and corresponding audio files of the 10 tested musical pieces, which are uploaded to the accompanying demo page.[4] The evaluation metrics are as follows:

- Musical fidelity (10 steps; 1: not faithful at all, . . ., 10: very faithful) — How the reduction score is faithful to the original ensemble score in terms of music acoustics.
- Subjective difficulty (10 steps; 1: very easy, . . ., 10: very difficult) — How difficult the reduction score is for playing with two hands.
- Musical naturalness (10 steps; 1 very unnatural, . . ., 10: very natural) — How natural the reduction score is as a piano score.
- Number of unplayable notes N_{unp} — How many notes and which notes should be removed from the reduction score to make it playable by a skillful pianist. We define the unplayable-note rate A_{unp}, a quantity normalized by the number of additional notes:

$$A_{unp} = \frac{N_{unp}}{\#R - \#\mathcal{M} - \#\mathcal{B}}. \quad (23)$$

Results are summarized in Fig. 6, where statistics (mean and standard deviation) are shown for each evaluation metrics and for each method. The results in Figs 6(a) and 6(b) indicate that subjective difficulty and musical fidelity monotonically increase with the additional note rate, which confirms the ability of the proposed methods for controlling

performance difficulty. For these two quantities, few differences can be found in the results for the three methods. The result in Fig. 6(c) shows that musical naturalness tends to decrease when increasing the additional-note rate. This can be understood from the fact when $A_{add} \simeq 0$ the reduction score consists mostly of melodic and bass notes, which should have high naturalness, and for larger A_{add} it becomes more demanding for the models to retain naturalness. For the highest difficulty case with target difficulty values (40, 40, 50) and $A_{add} \sim$ 90–100%, the *Iterated Gaussian* and *Iterated Fingering* methods outperform the baseline *Iterated Distance* method. This suggests the importance of incorporating sequential dependence of pitches in the piano score model for improving musical naturalness.

The result in Fig. 6(d) shows that, especially in the high difficulty regime, the unplayable-note rate is reduced by incorporating sequential dependence of pitches in the piano score model and even further so by incorporating the fingering motion. This suggests that although the same difficulty measure is used and it is not a perfect measure for describing the real difficulty of a piano score, a better piano score model can generate reduction scores with less unplayable notes.

D) Example results and discussions

Examples of piano reduction scores obtained by the *Iterated Fingering* method are shown in Fig. 7, together with the results of the subjective evaluation (the evaluation scores are given for the whole piece including the part not shown in the figure).[5] We see that results for larger target difficulty values have more harmonizing notes and are given larger fidelity and subjective-difficulty values. In the cases with $(\widetilde{D}_L, \widetilde{D}_R, \widetilde{D}_B) = (15, 15, 30)$ and (30, 30, 40), all notes are playable in the shown section and the latter case has a larger

[4]http://pianoarrangement.github.io/demo.html

[5]See   http://pianoarrangement.github.io/demo.html   for   more examples with sound files.
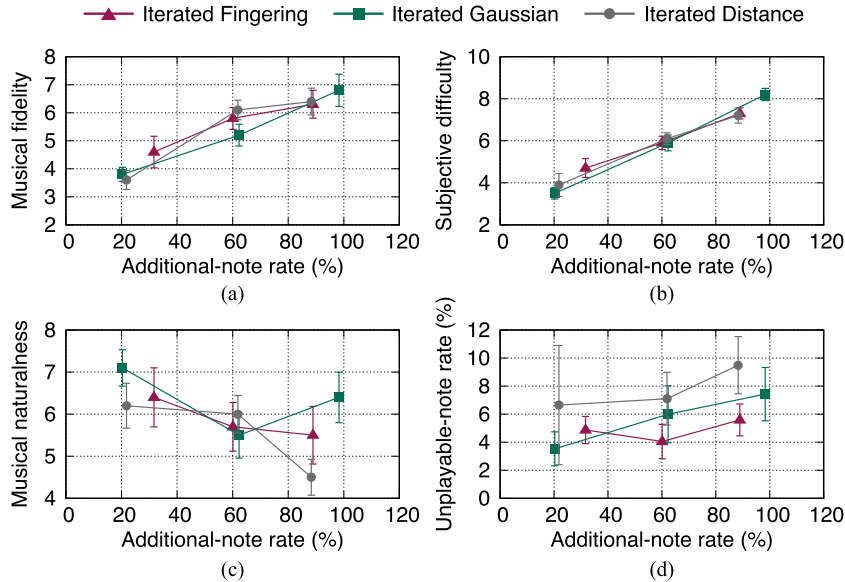
Fig. 6. Subjective evaluation results. For each method, the average results for the three sets of target difficulty are indicated with points. Bars indicate their standard errors.

musical-naturalness value. On the other hand, there are several unplayable notes in the case with $(\widetilde{D}_L, \widetilde{D}_R, \widetilde{D}_B) = (40, 40, 50)$, which leads to a smaller musical-naturalness value.

We were informed from the evaluators (professional music arrangers) that keeping more notes in a reduction score does not always improve musical naturalness. One reason is that flexibility for performance expression can be reduced by adding too many notes. We have therefore two important directions to further improve the piano reduction methods. One is to construct a more precise fingering model and difficulty measures based on it. However, as we discussed in Section II.B.2, a more complex model typically requires more training data for appropriate learning. Since a large-scale fingering dataset is currently not available, construction of such a dataset is also an important issue. Another is to incorporate more musical knowledge in the piano reduction model, particularly on harmonic aspects (e.g. completion of chordal notes and voice leading) and cognitive aspects (e.g. restricting notes over melodic notes to avoid mishearing of melodies).

Other left issues are identification of melodic and bass notes and score typesetting for reduction scores, which are manually done currently. As for the identification of melodic and bass notes, a simple method of taking the instrument part with the highest (lowest) mean pitch as the melody (bass) part for each bar can reproduce 40.6% (57.0%) of the indications in our test data. While this calls for a more refined method for automatically estimating the melodic and bass notes, we noticed that the choice is also subjective and it may be important to leave room for user preferences.

Finally, since the evaluation is subjective, it is also important to look at multiple ratings given by different evaluators. Such a large-scale subjective evaluation would be significant

for revealing finer relations between human's evaluation and the model's prediction.

V. CONCLUSION

We have described quantitative measures of performance difficulty for piano scores and a piano reduction method that can control the difficulty values based on statistical modeling. We followed the quantification of performance difficulty using statistical models proposed in [18] and found that the difficulty values can be used as indicators of performance errors. For the current amount of training data, we also found that the difficulty measures based on the Gaussian model yields the best accuracy of predicting performance errors. The problem of piano reduction is formalized as a statistical optimization problem following the framework of [4], and we improved the optimization method by proposing an iterative method. We confirmed the efficacy of the iterative optimization method and the algorithms are shown to be able to control subjective difficulty and musical fidelity. It was also found that incorporating sequential dependence and fingering motion in the piano-score model by using the Gaussian and fingering model improves generated reduction scores in terms of musical naturalness and the rate of unplayable notes. Directions for further improvements were also discussed.

Whereas it has been assumed that the same difficulty measures apply universally for all players in this study, they can be different for individual players depending on, for example, the size of hands. In the present framework, part of such individuality can be expressed by adapting the fingering model to individual players. This model adaptation can be realized in principle if one has a sufficient amount

Fig. 7. Examples of piano reduction scores obtained by the *Iterated Fingering* method (Wagner: Prelude to Die Meistersinger von Nürnberg). For clear illustration, only the first nine bars from a 27-bar excerpt in the test data are shown. Unplayable notes indicate those identified by the evaluator.

of musical scores that have been already played by an individual player. Another interesting direction is to adapt an individual's fingering model using the frequency of errors in his/her performance data, which could reduce the amount of necessary data.

A limitation of the present model is that timing errors and other rhythmic aspects are not considered. Rhythmic features may become important especially in polyrhythmic passages in which the left- and right-hand parts have contrasting rhythms (e.g. two against three rhythms). In such cases, the sum of difficulty values for the two hands may underestimate the total difficulty. To properly deal with these problems, it would be necessary to incorporate a performance timing model and interdependence between the two hands into the present framework.

The present formulation of combining a musical-score model and an edit model can also be applied to other forms of music arrangement if one replaces the piano fingering model with an appropriate score model of the target instrumentation/style and adapt the edit model for relevant edit operations. For example, if we combine a score model for jazz music and a proper edit model, it would be possible to develop a method for arranging a given piece in the rock music style (or other styles) into a piece in the jazz style.

Although this study has focused on piano arrangement, the framework can also be useful for music transcription [22]. In music transcription, musical-score models play an important role to induce an output score to be an appropriate one that respects musical grammar, style of target music, etc. [23, 24]. Especially in piano transcription, results

of multi-pitch detection contain a significant amount of spurious notes (false positives), which often make the transcription results unplayable [25]. By integrating the present piano-score model and an acoustic model (instead of the edit model) and applying the method for optimization developed in this study, one can impose constraints on performance difficulty of transcription results and reduce these spurious notes.

FINANCIAL SUPPORT

STATEMENT OF INTEREST

None.

REFERENCES

[1] Chiu, S.-C.; Shan, M.-K.; Huang, J.-L.: Automatic system for the arrangement of piano reduction, in *IEEE Int. Symp. on Multimedia*, San Diego, California, 2009, 459–464.

[2] Onuma, S.; Hamanaka, M.: Piano arrangement system based on composers' arrangement processes, in *Int. Computer Music Conf.*, New York, 2010, 191–194.

[3] Huang, J.-L.; Chiu, S.-C.; Shan, M.-K.: Towards an automatic music arrangement framework using score reduction. *ACM T. Multim. Comput. Commun. Appl.*, **8** (1) (2012), 8:1–8:23.

[4] Nakamura, E.; Sagayama, S.: Automatic piano reduction from ensemble scores based on merged-output hidden Markov model, in *Int. Computer Music Conf.*, Denton, Texas, 2015, 298–305.

[5] Takamori, H.; Sato, H.; Nakatsuka, T.; Morishima, S.: Automatic arranging musical score for piano using important musical elements, in *Int. Sound and Music Computing Conf.*, Aalto, 2017, 35–41.

[6] Tuohy, D.R.; Potter, W.D.: A genetic algorithm for the automatic generation of playable guitar tablature, in *Int. Computer Music Conf.*, Barcelona, 2005, 499–502.

[7] Hori, G.; Yoshinaga, Y.; Fukayama, S.; Kameoka, H.; Sagayama, S: Automatic arrangement for guitars using hidden Markov model, in *Int. Sound and Music Computing Conf.*, Copenhagen, 2012, 450–456.

[8] Hori, G.; Kameoka, H.; Sagayama, S: Input-output HMM applied to automatic arrangement for guitars. *J. Inform. Process. Soc. Japan*, **21** (2) (2013), 264–271.

[9] Maekawa, H.; Emura, N.; Miura, M.; Yanagida, M.: On machine arrangement for smaller wind-orchestras based on scores for standard wind-orchestras, in *Int. Conf. on Music Perception and Cognition*, Bologna, 2006, 268–273.

[10] Crestel, L.; Esling, P.: Live orchestral piano, a system for real-time orchestral music generation, in *Int. Sound and Music Computing Conf.*, Espoo, 2017, 434–442.

[11] Chiu, S.-C.; Chen, M.-S.: A study on difficulty level recognition of piano sheet music, in *IEEE Int. Symp. on Multimedia*, Irvine, California, 2012, 17–23.

[12] Sébastien, V.; Ralambondrainy, H.; Sébastien, O.; Conruyt, N.: Score analyzer: Automatically determining scores difficulty level for instrumental e-learning, in *Int. Conf. on Music Information Retrieval*, Porto, 2012, 571–576.

[13] Brown, P.F.; Pietra, V.J.D.; Pietra, S.A.D.; Mercer, R.L.: The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, **19** (2) (1993), 263–311.

[14] Parncutt, R.; Sloboda, J.A.; Clarke, E.F.; Raekallio, M.; Desain, P.: An ergonomic model of keyboard fingering for melodic fragments. *Music. Percept.*, **14** (4) (1997), 341–382.

[15] Hart, M.; Tsai, E.: Finding optimal piano fingerings. *UMAP J.*, **21** (1) (2000), 167–177.

[16] Al Kasimi, A.; Nichols, E.; Raphael, C: A simple algorithm for automatic generation of polyphonic piano fingerings, in *Int. Conf. on Music Information Retrieval*, Vienna, 2007, 355–356.

[17] Yonebayashi, Y.; Kameoka, H.; Sagayama, S.: Automatic decision of piano fingering based on a hidden Markov models, in *Int. Joint Conf. on Artificial Intelligence*, Hyderabad, 2007, 2915–2921.

[18] Nakamura, E.; Ono, N.; Sagayama, S.: Merged-output HMM for piano fingering of both hands, in *Int. Conf. on Music Information Retrieval*, Taipei, 2014, 531–536.

[19] Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77** (2) (1989), 257–286.

[20] Nakamura, E.; Yoshii, K.; Sagayama, S.: Rhythm transcription of polyphonic piano music based on merged-output HMM for multiple voices. *IEEE/ACM Trans. Audio, Speech, Language Process.*, **25** (4) (2017), 794–806.

[21] Nakamura, E.; Yoshii, K.; Katayose, H.: Performance error detection and post-processing for fast and accurate symbolic music alignment, in *Int. Conf. on Music Information Retrieval*, Suzhou, 2017, 347–353.

[22] Benetos, E.; Dixon, S.; Giannoulis, D.; Kirchhoff, H.; Klapuri, A.: Automatic music transcription: challenges and future directions. *J. Intell. Inf. Syst.*, **41** (3) (2013), 407–434.

[23] Raczyński, S.; Vincent, E.; Sagayama, S.: Dynamic Bayesian networks for symbolic polyphonic pitch modeling. *IEEE Trans. Audio, Speech, Language Process.*, **21** (9) (2013), 1830–1840.

[24] Ycart, A.; Benetos, E.: Polyphonic music sequence transduction with meter-constrained LSTM networks, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Calgary, 2018, 386–390.

[25] Nakamura, E.; Benetos, E.; Yoshii, K.; Dixon, S.: Towards complete polyphonic music transcription: Integrating multi-pitch detection and rhythm quantization, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Calgary, 2018, 101–105.

Eita Nakamura received his Ph.D. degree in physics from the University of Tokyo, Tokyo, Japan, in 2012. After having been a Postdoctoral Researcher at the National Institute of Informatics, Meiji University, and Kyoto University, Kyoto, Japan, he is currently a Research Fellow of Japan Society for the Promotion of Science. His research interests include music modeling and analysis, music information processing, and statistical machine learning.

Kazuyoshi Yoshii received his M.S. and Ph.D. degrees in informatics from Kyoto University, Kyoto, Japan, in 2005 and 2008, respectively. He had been a Senior Lecturer and is currently an Associate Professor at the Graduate School of Informatics, Kyoto University, and concurrently the Leader of the Sound Scene Understanding Team, RIKEN Center for Advanced Intelligence Project, Tokyo, Japan. His research interests include music analysis, audio signal processing, and machine learning.