EXPRESSIVE PERFORMANCE RENDERING: INTRODUCING PERFORMANCE CONTEXT

Sebastian Flossmann, Maarten Grachten
Dept. Computational Perception
Johannes-Kepler-University, Linz
{sebastian.flossmann,maarten.grachten}@jku.at

Gerhard Widmer
Dept. Computational Perception
Johannes-Kepler-University, Linz.
The Austrian Research Institute
for Artificial Intelligence (OFAI)
gerhard.widmer@jku.at

ABSTRACT

We present a performance rendering system that uses a probabilistic network to model dependencies between score and performance. The score context of a note is used to predict the corresponding performance characteristics. Two extensions to the system are presented, which aim at incorporating the current performance context into the prediction, which should result in more stable and consistent predictions. In particular we generalise the Viterbi-algorithm, which works on discrete-state Hidden Markov Models, to continuous distributions and use it to calculate the overall most probable sequence of performance predictions. The algorithms are evaluated and compared on two very large data-sets of human piano performances: 13 complete Mozart Sonatas and the complete works for solo piano by Chopin.

1 INTRODUCTION

Research on performance modelling and rendering constantly faces the problem of evaluation. The RENCON-Project [2] addresses this by providing a platform to present rendering systems to a public audience and in the process rate and judge the 'naturalness' and expressiveness of the rendered performances.

In the following we present YQX, the system that won all three awards in the autonomous category of the RENCON08 that was hosted by the ICMPC in September 2008 [1] [2] . However successful, the system tended to sometimes produce unstable, 'nervous' sounding performances. In response to this we present two extensions to the system that both aim for smoother variations in the performance, while ideally increasing the similarity to human performances. This is in concordance with [7] who states that an average of several performances can sound more aesthetically appealing than the actual performances going into the average. To achieve this we incorporate the *performance context*, information about the performance currently rendered, into the decisions. In contrast to the original system, which makes ad hoc decisions based only on the score context at hand, this adds a dynamic component.

In the first extension we use the additional information locally: The prediction for the previous note influences the current prediction according to the relations found in the training data. In the second extension we use the additional information to calculate the sequence of predictions that is globally the most probable of all, given the probabilities learned. In a series of experiments we test whether and to what extent the renderings of both extensions are smoother and more consistent than the renderings of the original system.

2 RELATED WORK

Much research has been done on the modelling and synthesis of expressive music. Since e.g. [10] gives a very detailed overview, only a few more recent approaches shall be mentioned here. Grindlay and Helmbold [1] use hierarchical Hidden Markov Models to generate expressive tempo values based on score characteristics. The different levels of hierarchy are used to represent different phrases of the piece. Due to the sophisticated learning algorithm and the intuitive structure the learned model is easy to interpret. They report good generalisation to new scores and positive evaluation in listening tests. A more recent approach [8], also submitted to RENCON08, uses the technique of Gaussian Processes [6] to automatically learn a mapping between score and performance characteristics. The model aims at predicting a sequence of performance parameters that is optimal with re-

[1] Given the specific context of this paper - published results of RENCON08, existing system YQX, availability of demo videos - we consider it pointless to try to keep this submission strictly anonymous.

[2] Videos of YQX in action at RENCON08 can be found at `http://www.cp.jku.at/projects/yqx`. The test pieces were composed specifically for the contest by Prof. Tadahiro Murao.

SMC 2009, July 23-25, Porto, Portugal

gard to the whole piece. Although the approach differs from ours, the intended effect is similar to our second extension to YQX. However the authors report a rather weak generalisation to new scores (perhaps due to the lack of high-quality training data).

3 THE DATA

In Spring 1989, Nikita Magaloff, a Russian-Georgian pianist famous for his Chopin interpretations, performed the entire work of Chopin for solo piano that was published during Chopin's lifetime (op. 1 - op. 64) in six public appearances at the Vienna Konzerthaus. These concerts were recorded with a Bösendorfer computer-controlled grand piano. The data set comprises over 150 pieces with over 320.000 performed notes. The MIDI data were manually matched to symbolic scores derived from scanned sheet music. The result is a unique corpus containing precisely measured performance data for almost all notes Chopin has ever written for solo piano.

The second data collection we use for the evaluation of our models are 13 complete Mozart Piano Sonatas played by the Viennese pianist R. Batik, likewise recorded on a Bösendorfer computer piano and matched to symbolic scores. This data set contains roughly 106.000 performed notes.

4 FEATURES AND TARGETS

We aim at learning a mapping between the score notes with their local score context and the human performance in our corpus. The prediction, the application of the learned mapping to unknown music, will be note-wise: each note of the melody of the score will be assigned three numeric values, the *targets*, determining the performance of the note. The targets are the dimensions with which we describe a piano performance: *loudness, articulation* and *tempo*. In the following instead of tempo we will actually use *ioi ratio*, which is directly related. The characteristics of a note and its local score context are described by the *features* extracted from the score.

One of these features (IR-Arch, see below) is based on E. Narmour's Implication-Realization model of melodic expectation [5]. The theory constitutes an alternative to Schenkerian analysis, focused more on cognitive aspects of expectation than on musical analysis. The model analyses the musical progression of a piece and the expectations aroused in the listener's mind. One of the main claims of Narmour is that sequences of intervals, harmonies etc. either produce further expectations, a situation of *non-closure*, or not, a situation of musical *closure*. Calculating the distance of a melody note to the nearest point of closure provides clues about whether a note represents a phrase boundary or not.

In our experiments we use the following (very small) set of score features:

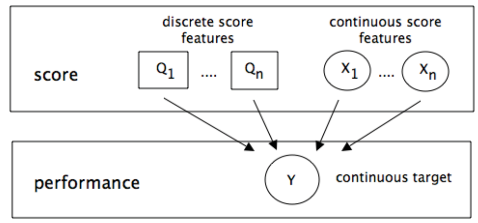

Figure 1. The probabilistic network forming the YQX system

Pitch Interval The interval between a melody note and its successor, measured in semitones.

Duration Ratio The logarithmic ratio between the score duration of a melody note and its successor.

I-R Arch The distance from the nearest point of closure, calculated from the Implication-Realization analysis.

The targets to be predicted are defined as follows:

IOI Ratio The logarithmic ratio of the length between two successive played notes and the length between the two corresponding score notes. A positive value indicates that the time between two notes is longer than notated, resulting in a decreased performance tempo.

Loudness The logarithmic ratio of the midi velocity of a note and the mean velocity of the performance. Thus positive values are louder than average, negative values softer.

Articulation This measures the amount of legato that is applied by a quotient of the gap between two successive played notes and the notated gap between the two corresponding score notes.

5 YQX - THE SIMPLE MODEL

YQX models the dependencies between score features and performance targets by means of a probabilistic network. The network consists of several interacting nodes representing the different features and targets. Each node is associated with a probability distribution over the values of the corresponding feature or target. A connection between two nodes in the graph implies a conditioning of one feature or target distribution on the other. Discrete score features (the set of which we call $\mathbf{Q}$) are associated with discrete probability tables, while continuous score features ($\mathbf{X}$) are modelled by gaussian distributions. The predicted performance characteristics, the targets ($\mathbf{Y}$), are continuously valued and conditioned on the set of discrete and continuous features. Figure 1 shows the general layout. The semantics is that of a linear gaussian model [4]. This implies that the case of a

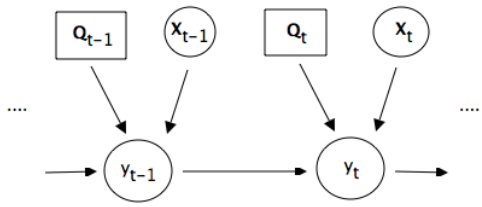

Figure 2. The network unfolded in time

continuous distribution parenting a continuous distribution is implemented by making the mean of the child distribution depend linearly on the value of the condition. In the following sets are marked with bold letters, vectors with an arrow super-scribed over the variable name.

Mathematically speaking this models a target y as a conditional distribution $P(y|\mathbf{Q}, \mathbf{X})$. Following the linear Gaussian model this is a gaussian distribution $\mathcal{N}(y; \mu, \sigma^2)$ with μ varying linearly with $\mathbf{X}$: given $\mathbf{Q} = \mathbf{q}$ and $\mathbf{X} = \overrightarrow{x}$ [3]

$$\mu = d_{\mathbf{q}} + \overrightarrow{k}_{\mathbf{q}} \cdot \overrightarrow{x},$$

where $d_{\mathbf{q}}$ and $\overrightarrow{k}_{\mathbf{q}}$ are estimated from the data by least squares linear regression. The average residual error of the regression is the σ^2 of the distribution. In effect we collect all instances in the data that share the same combination of discrete feature values and build a joint probability distribution of the continuous features and targets of those instances. This implements the conditioning on the discrete features $\mathbf{Q}$. The linear dependency of the mean of the target distribution on the values of the continuous features introduces the conditioning on $\mathbf{X}$. This constitutes the training phase of the model.

The prediction of the performance is done note by note. The score features of a note are entered into the network as evidence $\overrightarrow{x}$ and $\mathbf{q}$. The instantiation of the discrete features selects the appropriate probability table and the parameterisation $d_{\mathbf{q}}$ and $\overrightarrow{k}_{\mathbf{q}}$, the continuous features are used for calculating the mean of the target distribution μ. This value is used as the prediction for the specific note. As the targets are independent we create model and prediction for each target separately.

6 YQX - THE ENHANCED DYNAMIC MODEL

In the following we present two extensions of the system that both introduce a dynamic component by incorporating the prediction made for the preceding score note into the prediction of the current score note. Graphically this corresponds to first unfolding the network in time and then adding an arc from the target in time-step $t-1$ to the target in time-step t. Figure 2 shows the unfolded network. This should lead to smoother and more consistent performances with less abrupt changes and ideally to an increase of the overall prediction quality.

6.1 YQX with local maximisation

The first method is rather straight forward: We stick to the linear gaussian model and treat the additional parent (the target y_{t-1}) to the target y_t like an additional feature that we calculate from the performance data. In the training process the joint distribution of the continuous features, the target y_t and the target in the previous time-step y_{t-1} given the discrete score features, in mathematical terms $P(y_{t-1}, y_t, \overrightarrow{x}_t|\mathbf{q}_t)$, is estimated. That slightly alters the conditional distribution of the target y_t to $P(y_t|\mathbf{Q}, \mathbf{X}, y_{t-1}) = \mathcal{N}(y; \mu, \sigma^2)$ with [4]

$$\mu = d_{\mathbf{q},y_{t-1}} + \overrightarrow{k}_{\mathbf{q},y_{t-1}} \cdot (\overrightarrow{x}, y_{t-1}).$$

The prediction phase is equally straight forward. Just as in the simple model, the mean of $P(y_t|\mathbf{q}_t, \overrightarrow{x}_t, y_{t-1})$ is used as the prediction for the score note in time-step t. This is the value with the locally highest probability.

6.2 YQX with global maximisation

The second approach drops the concept of a linear gaussian model completely. In the training phase the joint conditional distributions $P(y_{t-1}, y_t, \overrightarrow{x}_t|\mathbf{q}_t)$ are estimated as before, but no linear regression parameters need to be calculated. The aim is to construct a sequence of predictions that maximises the conditional probability of the performance given the score features with respect to the complete history of predictions made up to that point.

This is calculated in analogy to the Viterbi-decoding in Hidden Markov Models (HMMs), where one tries to find the best explanation for the observed data [3]. Apart from the fact that the roles of evidence nodes and query nodes are switched, the main conceptual difference is that we have to deal with continuous instead of tabular distributions as used in the standard HMM setup. This rules out the dynamic programming algorithm usually applied but calls for an analytical solution, which we present in the following. Like the Viterbi algorithm the calculation is done in two steps: a forward and a backward sweep. In the forward movement the most probable target is calculated relative to the previous time-step. In the backward movement, knowing the final point of the optimal path, the sequence of predictions is found via backtracking through all time-steps.

6.2.1 The forward calculation

Let $\overrightarrow{x}_t, \mathbf{q}_t$ be the sets of continuous and discrete features at time t and N be the number of data points in piece. Let fur-

[3] We treat the real valued set of continuous score features like a vector

[4] The construct $(\overrightarrow{x}, y_{t-1})$ is a concatenation of the vector $\overrightarrow{x}$ and the value y_{t-1} leading to a new vector with a dimension $dim(\overrightarrow{x}) + 1$.

ther be α_t the probability distribution over the target values y_t to conclude the optimal path from time-steps 1 to $t-1$. By means of a recursive formula $\alpha(y_t)$ can be calculated for all time-steps of the unfolded network [5] :

$$\alpha(y_1) = p(y_1|\mathbf{x}_1, \mathbf{q}_1) \tag{1}$$
$$\alpha(y_t) = \max_{y_{t-1}\in\mathbb{R}} [p(y_t, y_{t-1}|\overrightarrow{x}_t, \mathbf{q}_t) \cdot \alpha(y_{t-1})] \tag{2}$$

This formula can be interpreted as follows: assuming that we know for all the target values y_{t-1} in time-step $t-1$ the probability of being part of the optimal path, we can calculate for each target value y_t in time-step t the predecessor that yields the highest probability for each specific y_t of being on the optimal path. In the backward movement we will start with the most probable final point of the path (the mean of the last α) and then backtrack to the beginning by choosing the best predecessors. As we cannot calculate the maximum over all $y_{t-1} \in \mathbb{R}$ directly, we need an analytical way to calculate $\alpha(y_t)$ from $\alpha(y_{t-1})$, which we will derive in the following. We will also show that $\alpha(y_t)$ remains gaussian through all time-steps.

In the following we will use the distribution $p(y_{t-1}|y_t, \overrightarrow{x}_t, \mathbf{q}_t) \propto \mathcal{N}(y_{t-1}; \mu_{t-1}, \sigma^2_{t-1})$ that can be calculated via conditioning from the joint conditional distribution $p(y_{t-1}, y_t, \overrightarrow{x}_t|\mathbf{q}_t)$ that is estimated in the training of the model. For details as to how this is done see e.g. [6]. As we will prove that the $\alpha(y_t)$ are gaussian, we will refer to the mean and variance by $\mu_{\alpha,t}$ and $\sigma^2_{\alpha,t}$.

The inductive definition of α (eq. 2) can be rewritten (the conditioning on $\mathbf{q}_t, \overrightarrow{x}_t$ is omitted for simplicity):

$$\alpha(y_t) = \max_{y_{t-1}\in\mathbb{R}} [p(y_{t-1}|y_t) \cdot \alpha(y_{t-1})] \cdot p(y_t) \tag{3}$$

Assuming that $\alpha(y_{t-1})$ is gaussian, the result of the product in brackets is gaussian $\mathcal{N}(y_{t-1}; \mu_*, \sigma^2_*)$ with a normalising constant z, that also is gaussian in either of the means of the factors:

$$\sigma^2_* = \frac{\sigma^2_{t-1} * \sigma^2_{\alpha,t-1}}{\sigma^2_{t-1} + \sigma_{\alpha,t-1}} \tag{4}$$
$$\mu_* = \sigma^2_* \left(\frac{\mu_{t-1}}{\sigma^2_{t-1}} + \frac{\mu_{\alpha,t-1}}{\sigma^2_{\alpha,t-1}} \right) \tag{5}$$
$$z = \frac{1}{\sqrt{2\pi|\sigma^2_{1-1} + \sigma^2_{\alpha,t-1}|}} e^{\left(\frac{-(\mu_{t-1} - \mu_{\alpha,t-1})^2}{2(\sigma^2_{t-1} + \sigma^2_{\alpha,t-1})} \right)} \tag{6}$$

Later on z will be multiplied with a gaussian distribution over y_t, hence z has to be transformed to a distribution over the same variable. By finding a y_t, such that the exponent in eq. 6 equals 0, we can construct a proper μ_z and σ^2_z. Note that the variable μ_{t-1} is dependent on y_t due to the conditioning of $p(y_{t-1}|y_t)$ on y_t.

$$z \propto \mathcal{N}(y_t; \mu_z, \sigma^2_z) \tag{7}$$
$$\mu_z = -\frac{\sigma^2_t \cdot (\mu_{t-1} + \mu_{\alpha,t-1}) + \mu_t \cdot \sigma^2_{t,t-1}}{\sigma^2_{t,t-1}} \tag{8}$$
$$\sigma^2_z = \sigma^2_{t-1} + \sigma^2_{\alpha,t-1} \tag{9}$$

As z is independent of y_{t-1} it is not affected by the calculation of the maximum:

$$\alpha(y_t) \propto \max_{y_{t-1}\in\mathbb{R}} [\mathcal{N}(y_{t-1}; \mu_*, \sigma^2_*)] \cdot \mathcal{N}(y_t; \mu_z, \sigma^2_z) \cdot p(y_t) \tag{10}$$
$$= \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \mathcal{N}(y_t; \mu_z, \sigma^2_z) \cdot p(y_t) \tag{11}$$

The factor $\frac{1}{\sqrt{2\pi\sigma^2}}$ can be neglected as it does not affect the parameters of the final distribution of $\alpha(y_t)$. The distribution $P(y_t)$ is gaussian by design and hence the remaining product again results in a gaussian and a normalising constant. As the means of both factores are fixed, the normalising constant this time is a single factor. The mean $\mu_{\alpha,t}$ and variance $\sigma^2_{\alpha,t}$ of $\alpha(y_t)$ follow:

$$\alpha(y_t) \propto \mathcal{N}(y_t; \mu_{\alpha,t}, \sigma^2_{\alpha,t}) \tag{12}$$
$$\sigma_{\alpha,t} = \frac{\sigma^2_t \cdot \sigma^2_z}{\sigma^2_t + \sigma^2_z} \tag{13}$$
$$\mu_{\alpha,t} = \sigma_{\alpha,t} \left(\frac{\mu_z}{\sigma^2_z} + \frac{\mu_t}{\sigma^2_t} \right). \tag{14}$$

Thus $\alpha(y_t)$ is gaussian in y_t, assuming $\alpha(y_{t-1})$ is gaussian. Since $\alpha(y_1)$ is gaussian, it follows that $\alpha(y_t)$ is gaussian for $1 \leq t \leq N$. This equation shows that the mean and variance of $\alpha(y_t)$ can be computed recursively over the mean $\mu_{\alpha,t-1}$ and variance $\sigma^2_{\alpha,t-1}$ of $\alpha(y_{t-1})$. The parameters of α_{y_1} equal μ_{y_1} and $\sigma^2_{y_1}$, which are the mean and variance of the distribution $p(y_1|\overrightarrow{x}_1, \mathbf{q}_1)$, and are estimated from the data.

6.2.2 *The backward calculation*

Once the mean and variance μ_t, σ^2_t of $\alpha(y_t)$ are known for $1 \leq t \leq N$, the optimal sequence $y_1, ..., y_N$ can be calculated:

$$y_N = \mu_{\alpha,N} \tag{15}$$
$$y_{t-1} = \operatorname*{argmax}_{y_{t-1}} \left[\mathcal{N}(y_{t-1}; \mu_*, \sigma^2_*\right] \tag{16}$$
$$= \mu_* \tag{17}$$

6.3 The Problem of Flatness

Both extensions presented above are designed to eliminate fast fluctuations from the predicted curves that, though small

[5] We use $\alpha(y_t)$ and $p(y_t)$ as an abbreviation of $\alpha(Y_t = y_t)$ and $p(Y_t = y_t)$, respectively

in amplitude, lead to unmusical irregularities in the rendered performances. The predicted curves are smooth, as we will show below, and suitable for rendering a consistent and principally acceptable performance. On the other hand, the flatter a curve, the more mechanical and unexpressive will the rendered performance sound. Based on the predictions we have, we can now try to counter this and superimpose peaks at selected points in the curves. To do this, a small set of note-level rules, automatically extracted by Widmer [9] from real performance data, is applied:

1. Staccato Rule: If two successive notes have the same pitch and the second of the two is longer, then the first note is played staccato.
2. Delay Next Rule: If two notes of the same length are followed by a longer note, the last note is played slightly delayed.
3. Trill Rule: If a trill is indicated in the score, the duration of the trill note is slightly prolonged.

7 RESULTS

We evaluated the algorithms on three data sets: The complete Chopin piano works, played by N. Magaloff and 13 complete Mozart Piano Sonatas, played by R. Batik, which were split into fast movements and slow movements. We first present the results of three-fold crossvalidations on the separate data sets and then take a detailed look at the predictions for an exemplary Mozart Sonata, and at the effects of the note-level rules. The quality of a predicted performance is measured by Pearson's correlation coefficient between the predicted curve and the curve calculated from the training data.

The crossvalidations, the results of which are given in table 1, show lower correlations on the Chopin data, implying that these data are much harder to predict than the Mozart pieces. This is probably due to the much higher variation that the limited information in the score features must account for. Testing with different sets of features shows that the prediction quality of a particular target depends greatly on the choice of features. As the goal of this paper is to compare different methods for introducing performance context, we restrict ourselves to one particular set of features and thereby refrain from choosing the best set for each target.

On the Mozart Sonatas the globally optimised algorithm shows a slight quality increase in predictive accuracy of the ioi ratios (12.5% for the fast movements and 10.9% for the slow movements) and loudness (10.0% and 7.0%). The slight decrease in average correlation for the articulation is not too surprising, as articulation is a rather local phenomenon that does not profit from long-term dependencies. For the Chopin data only a minor improvement in the prediction of the ioi ratios was registered (2.5%). The loudness in particular yielded low correlations. This is a problem we already encountered with the original YQX and that will need to be analysed in more detail in the future.

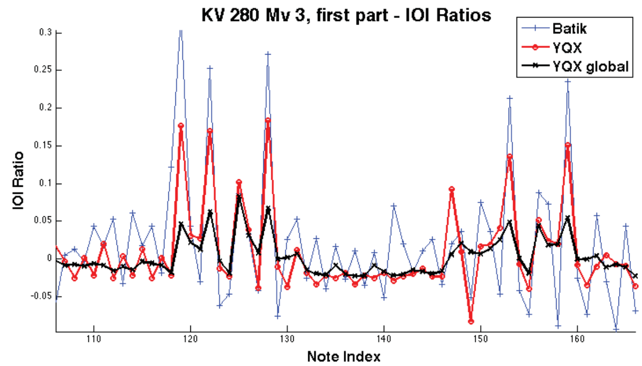

Figure 3. IOI Ratios predicted for bars 31 - 54 of K. 280, Mv.3

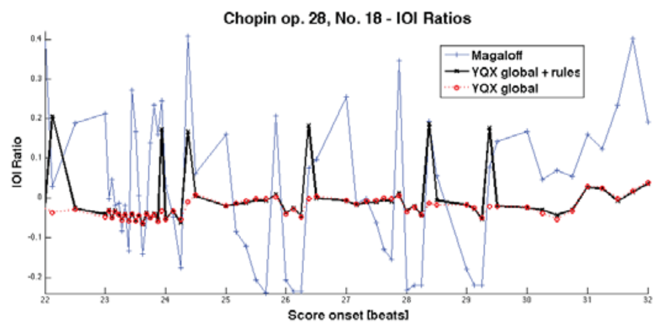

Figure 4. Effect of the note level rules applied to Chopin Prelude op.28 No. 18, bars 12 - 17

Figure 3 shows the ioi ratio predictions for bars 31 to 54 in the third movement of the Mozart Sonata K. 280. The original YQX algorithm exhibits small fluctuations that are rather uncorrelated with the human performance. This results in small, but noticeable irregularities in the rendered performance. In contrast to the human performance, which is anything but a flat curve, those make the result sound inhomogeneous instead of lively and natural. The globally optimised YQX eliminates those while still showing some of the peaks present in the human performance. The correlation for the movement was improved by 57.2% from 0.29 (YQX) to 0.456 (YQX global).

Figure 4 shows the effect of the note level rules described in section 6.3 on the ioi ratios predicted for Chopin Prelude op.28 No.18, bars 12 - 17. Instances of the Delay Next Rule occur at beats $24, 24.5, 26.5, 28.5$ and 29.5, all of which coincide with great contrasts in Magaloff's performance.

	Mozart fast			Mozart slow			Chopin		
	ioi	loud	art	ioi	loud	art	ioi	loud	art
YQX	0.233	0.171	0.323	0.339	0.217	0.200	0.160	0.108	0.323
YQX local	0.238	0.160	0.296	0.345	0.196	0.161	0.169	0.053	0.313
YQX global	0.262	0.188	0.319	0.376	0.232	0.190	0.164	0.075	0.316

Table 1. Results of the crossvalidations. The values shown are correlations of the predicted performance with a human performance

8 CONCLUSION

The automatic synthesis of expressive music is a very challenging task, especially regarding the evaluation of a system, as one cannot really judge the aesthetic qualities of a performance by numbers. An adequate measure of quality can only be provided by human judgement. The rendering system we present passed this hurdle in the RENCON 2008 and therefore poses a baseline for our current research. The two extensions we devised address the problem of unsteady performances by incorporating the current performance context into the predictions. This proved to be a tightrope walk: Finding a way to restrain the predicted curves on the one hand but not losing (ideally increasing) similarity to the original curves on the other hand.

Of the data we tested our algorithms on, the Mozart Sonatas form a simpler task than the Chopin data. We registered a considerable increase in similarity to the real performances while achieving our goal of smoother predictions. The Chopin data pose a harder nut to crack. Due to the vast amount of highly heterogeneous data that has to be accounted for by a very limited set of features we were not able to increase the prediction quality significantly.

We consider this a work in progress. There is still a long way to go to a machine-generated performance that sounds profoundly musical. The main goal in the near future will be to define a set of features that is capable of explaining data with a high degree of interpretational freedom, like the Chopin data. This will raise the problem of how to balance the predicted performances against the peaks superimposed by the note level rules. We also have to solve the problem of big tempo or loudness differences within pieces that affect the global mean, as this is the reference point for our predictions. A promising approach is to calculate the ioi ratios and loudness relative to a current mean and incorporate the mean tempo and loudness curves into the prediction process. The biggest challenge, however, will be to combine the model with phrase level predictions, as e.g. are made in [11].

9 ACKNOWLEDGEMENTS

This work is funded by the Austrian National Science Fund (FWF) under project number P19349-N15.

References

[1] GRINDLAY, G., AND HELMBOLD, D. Modeling, analyzing, and synthesizing expressive piano performance with graphical models. *Machine Learning 65*, 2-3 (June 2006), 361–387.

[2] HASHIDA, M. Rencon - performance rendering contest for computer systems. http://www.renconmusic.org/, September 2008.

[3] JUANG, B. H., AND RABINER, L. R. Hidden markov models for speech recognition. *Technometrics, 33*, 3 (August 1991), 251–272.

[4] MURPHY, K. *Dynamic Bayesian Networks: Presentation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

[5] NARMOUR, E. *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. University of Chicago Press, 1990.

[6] RASMUSSEN, C. E., AND WILLIAMS, C. K. I. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[7] REPP, B. The aesthetic quality of a quantitatively average music performance: Two preliminary experiments. *Music Perception 14*, 4 (1997), 419–444.

[8] TERAMURA, K., OKUMA, H., AND ET AL. Gaussian process regression for rendering music performance. In *Proceedings of the 10th International Conference on Music Perception and Cognition (ICMPC)* (2008).

[9] WIDMER, G. Machine discoveries: A few simple, robust local expression principles. *Journal of New Music Research 31*, 1 (2002), 37–50.

[10] WIDMER, G., AND GOEBL, W. Computational models of expressive music performance: The state of the art. *Journal of New Music Research 33*, 3 (2004), 203–216.

[11] WIDMER, G., AND TOBUDIC, A. Playing mozart by analogy: Learning multi-level timing and dynamics strategies. *Journal of New Music Research 32*, 3 (2003), 259–268.