FINE-TUNING MIDI-TO-AUDIO ALIGNMENT USING A NEURAL NETWORK ON PIANO ROLL AND CQT REPRESENTATIONS

Sebastian Murgul
Klangio GmbH
Karlsruhe, Germany
sebastian.murgul@klang.io

Moritz Reiser
University of Music Karlsruhe
Karlsruhe, Germany
moritz.reiser@t-online.de

Michael Heizmann
Karlsruhe Institute of Technology
Karlsruhe, Germany
michael.heizmann@kit.edu

Christoph Seibert
University of Music Karlsruhe
Karlsruhe, Germany
seibert@hfm-karlsruhe.de

June 27, 2025

ABSTRACT

In this paper, we present a neural network approach for synchronizing audio recordings of human piano performances with their corresponding loosely aligned MIDI files. The task is addressed using a Convolutional Recurrent Neural Network (CRNN) architecture, which effectively captures spectral and temporal features by processing an unaligned piano roll and a spectrogram as inputs to estimate the aligned piano roll. To train the network, we create a dataset of piano pieces with augmented MIDI files that simulate common human timing errors. The proposed model achieves up to 20% higher alignment accuracy than the industry-standard Dynamic Time Warping (DTW) method across various tolerance windows. Furthermore, integrating DTW with the CRNN yields additional improvements, offering enhanced robustness and consistency. These findings demonstrate the potential of neural networks in advancing state-of-the-art MIDI-to-audio alignment.

Keywords Music Information Retrieval · MIDI-to-Audio Alignment · Neural Network

1 Introduction

In the last decade, substantial progress has been made in the field of Music Information Retrieval (MIR) using deep learning techniques. A key downstream task is Automatic Music Transcription (AMT) [1], which seeks to convert audio recordings of music into a symbolic music representation like MIDI. Since many of the approaches rely on frame-wise annotations and annotating real world recordings is quite time-consuming, the use of synthesized MIDI data has become popular. While this offers a viable path to obtaining large training datasets, incorporating real-world audio could lead to further improvements. Using automatic MIDI-to-audio alignment can accelerate the annotation process and make real-world audio suitable for training.

Dynamic Time Warping (DTW) [2, 3] is one of the most established algorithmic methods and is considered the industry standard. DTW has been proposed by Dannenberg and Hu in 2003 and uses dynamic programming to optimize a monotonic alignment path by minimizing the cost between the aligned feature vectors. These feature vectors consist of chroma features derived from the spectrogram of the audio and the piano roll of the MIDI. The original method has been improved by several extensions [4, 5], and it has proven to be extremely efficient for aligning or matching recordings of music to the corresponding MIDI transcriptions. In 2009, Ewert et al. introduced the DLNCO features which improve the temporal accuracy without sacrificing the robustness [4]. The computational complexity of DTW is quadratic in the input length which becomes a major issue when dealing with longer audio recordings. Therefore, Prätzlich et al. proposed a Memory restricted Multiscale DTW (MrMsDTW) [5]. Besides the DTW method, Gingras et al. proposed a

three-step matcher algorithm based on pitch clustering in 2011 [6]. In 2014, Chen et al. introduced two more algorithmic approaches [7]. The first one converts the note sequences into strings and performs a string matching. The second one uses the divide and conquer principle to split the input note sequences into smaller units that are synchronized recursively. Besides the algorithmic approaches, neural networks have been used to improve the feature extraction and the DTW cost calculation. Kwon et al. use an RNN-based music transcription algorithm in order to extract chroma features from an audio's spectrogram which can be used in a DTW alignment with chroma features extracted from MIDI [8]. Wang et al. use a triplet pair architecture to learn features from low-level representation using a hinge loss function [9]. Agrawal et al. propose a Siamese network architecture which performs a feature extraction on the audio spectrograms and then estimates a distance between the two audio inputs [10]. A preprocessing pipeline incorporating DTW and a pre-trained transcription model is used in [11] and [12] to align unaligned real-world transcriptions for fine-tuning.

In terms of AMT dataset creation, DTW is used for fine alignment of the MAESTRO dataset [13] in order to account for any jitter in the audio or MIDI recording. In Pop2Piano, Choi et al. use DTW to synchronize piano arrangements to the corresponding original music video [14]. One major downside of using these techniques for dataset alignment is that they only consider one complete timeframe at once and cannot perform a frequency band specific alignment. Therefore, it can occur that for example the lower notes of a chord are aligned perfectly, but the higher notes are annotated too early.

In this work, a deep learning approach for fine-tuning loosely aligned MIDI annotations, which could be derived from DTW, is introduced. The model is based on a CRNN architecture and is trained to estimate an aligned piano roll given an unaligned piano roll and the performance audio. For training, a suited methodology for creating synthetic unaligned training examples from the well known piano transcription dataset MAPS [15] is presented. Finally, the synchronization results are evaluated with real world examples as a proof of concept.

2 Model Description

The two critical components for our approach are the feature extraction of the two cross-modal inputs and the temporal modelling in order to determine the relation between the unaligned and aligned notes.

2.1 Preprocessing

The MIDI files are converted to piano roll representations with pitches ranging from 21 to 108 and a temporal resolution of 100 frames per second. The velocity of the notes within the MIDI files is ignored in the piano roll creation, hence resulting in a binary image. In order to retain the individual note onsets of directly successive notes, small gaps with a width of one frame (10 ms) are inserted before an onset if needed. The input audio signal is resampled to 16 kHz. The CQT is calculated with a hop size of 160 using the librosa library [16]. To match the input piano roll, a total number of 88 frequency bins with 12 bins per octave is used, starting at a minimum frequency of 27.5 Hz. To highlight the contours in the CQT, the magnitude values are employed with a logarithmic scale. Since the audio and the unaligned MIDI file can be of different durations, the CQT and the piano rolls are zero padded to a common temporal length.

2.2 Network Architecture

The model architecture is visualized in Figure 1. It is organized in three blocks: two identical convolutional blocks for the feature extraction in the piano roll and the CQT images and one recurrent block that compares the extracted features in order to estimate an output piano roll. The architecture of the convolutional blocks is inspired by the model for singing transcription by Schwabe et al. [17]. The CQT and the piano roll share the same dimensions of $N \times 88$ where N represents the temporal dimension and 88 corresponds to the number of pitches. These images are processed by three two-dimensional convolutional layers (Conv2D) with a kernel size of 3. The first two convolutional layers consist of 16 filters, the third one uses 32 filters. ReLU is used as activation function after the convolutional layers. Max pooling of the pitch dimension with a size of 2 is inserted after each of the last two convolutional layers. The output of the last convolutional layer is fed into a fully connected layer (Dense). Batch Normalization is used after each convolutional layer to improve the feature stability and a dropout of 0.5 is employed after the intermediate Dense layers and the BiLSTM layer as a regularization method. The results of both strands are concatenated to a shape of $N \times 512$ where 512 corresponds to our embedding size. To improve the modelling of temporal constraints, the output is fed into a bidirectional LSTM layer (BiLSTM) with an input size of 512 and a hidden size of 256. The output of the BiLSTM is processed by another fully connected layer that returns a piano roll with the dimensions $N \times 88$. Using a sigmoid activation function allows the interpretation as probabilities.
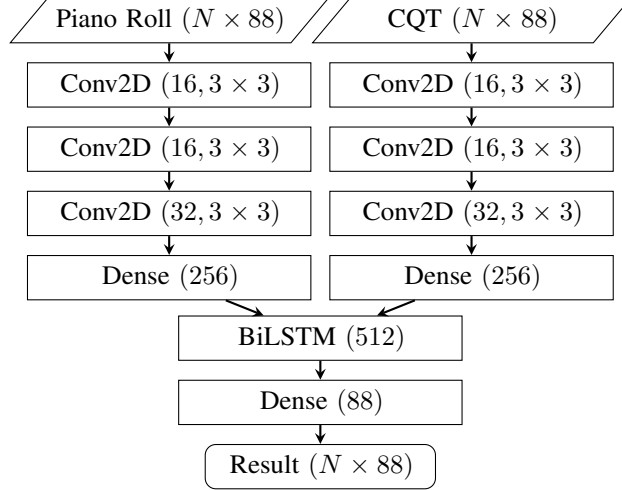
Figure 1: Architecture of the proposed CRNN model for MIDI-to-audio alignment. Two parallel convolutional branches extract spectral and temporal features from the unaligned piano roll and CQT input. The features are concatenated and passed through a BiLSTM layer and dense output layer to estimate the aligned piano roll as a binary activation map over time and pitch.

2.3 Postprocessing

To construct the resulting MIDI sequence from the returned piano roll, a postprocessing step is employed. Firstly, the piano roll is converted into a note sequence by thresholding the estimated piano roll with 0.5 and segmenting the detected note blocks for each pitch by searching for transitions. Since it is not guaranteed that the neural network returns the correct number of notes, the estimated notes are matched with the notes of the input MIDI file using the pitch and temporal order. After successful matching, the estimated notes' start and end times are used to update the input MIDI sequence. If a corresponding note is not found in the estimated sequence, the original unaligned note will be added to the resulting MIDI file instead.

3 Experiments

This section outlines the experimental setup, including the datasets used, the training procedure, and the evaluation configuration employed to assess the proposed model.

3.1 Datasets

3.1.1 Unaligned MAPS Dataset

To be able to train, validate, and test a neural network for synchronization, enough unsynchronized training data with dedicated labels is needed. Because such a dataset has not been available at the time of this work, the synthesized musical pieces of the MAPS database [15] are used. For each of the pieces, the corresponding perfectly aligned MIDI file is augmented to obtain a loosely aligned version. Firstly, the audio and MIDI files of each example are segmented into parts with a length of about 30 s each, yielding more training data and a more consistent sequence length. The individual lengths can vary since points in time without active notes are searched in order to find reasonable splitting points.

The onsets and offsets of all notes are slightly shifted to simulate the timing variations characteristic of human performance. Such timing variations can occur unintentionally, due to limitations in motor control or perceptual processes, but may also be deliberately employed as a means of musical expression or communication [18].

A possible reference for the maximum value of variation was derived from [19] with 50 ms for playing piano music and is used, for example, in [20] and [21]. However, given the complex interplay of perceptual, motor, and expressive factors influencing timing, and in order to accommodate a broader spectrum of timing deviations in our model, we chose to allow for a maximum onset/offset deviation of 100 ms. Figure 2 illustrates an example of such timing variations, showing how note boundaries are temporally displaced in human performance.
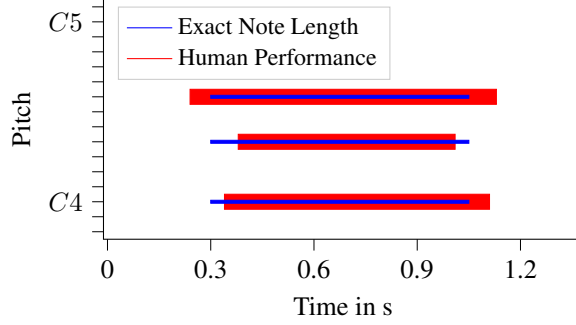
3

Figure 2: Illustration of timing deviations in a human piano performance. The blue bars represent ideal note lengths (as in a metronome-guided score), while the red bars show real-world timing deviations due to natural variation in note onsets and durations.

Besides shifting onsets and offsets, tempo fluctuations can also be simulated. Therefore, a scaling factor can be used to change the tempo of the unaligned MIDI files. This additional augmentation step is only applied for the experiments in Section 4.2.5.

The resulting unaligned MAPS dataset consists of triplet examples, each containing an audio, a perfectly aligned MIDI, and an artificially unaligned MIDI file. The dataset is split into three subsets. The 210 synthesized examples are used as training split, and the 60 real recordings of ENSTDkAm and ENSTDkCl are used as validation and test split respectively.

3.1.2 Real World Piano Recordings

The unaligned MAPS dataset is created in a way that the audio performances are temporally accurate representations while the corresponding MIDI files contain additional human timing variations. In the use case of recording a dataset however, the exact opposite is the case. To test how the network deals with the task of adjusting MIDI scores to match the human performance, six real world piano performances have been recorded. These were played on a digital piano and recorded by a Digital Audio Workstation (DAW), allowing the saving of MIDI data for usage as perfectly aligned labels. The temporally accurate MIDI file for each piece is taken from its digital musical score. During the recording of the pieces, a metronome was used, so only scenarios without tempo changes are considered here.

3.2 Model Training

To fit the model to the training data, the binary cross entropy loss is minimized using the Adam optimizer [22] with a learning rate of 0.001. The training is performed using a mini-batch size of 8 and is run for 50 to 200 epochs with early stopping. The model converges quite fast resulting in a training time on an NVIDIA T4 with 16 GB RAM of 2 h on average.

3.3 Evaluation Metrics

For the evaluation of the model, the absolute onset errors and the alignment accuracy in different tolerance windows are calculated. The absolute onset errors are obtained by comparing the note onset times for corresponding notes in the estimated and the ground truth MIDI files. The alignment accuracy is defined as the percentage of correctly aligned notes whose onset time deviation lies within a certain tolerance window. For ease of use, the implementation in the mir_eval library [23] is used. Note offsets are not considered in the evaluation, since they can have greater deviations, caused, for example, by reverb.

3.4 Baselines

In order to assess our model's performance, it is compared with the results of using the loosely aligned MIDI and DTW as a state-of-the-art method. For DTW, the implementation uses the synctoolbox by Müller et al. [24]. The DLNCO features [4] are extracted from the audio and the MIDI file respectively and the warping path is then determined by applying the MrMsDTW variant [5].

4 Results

In this section, we evaluate the performance of the proposed method on both the test split of the unaligned MAPS dataset and real-world piano recordings. The alignment quality is assessed using the evaluation metrics defined in Section 3.3. We first compare our model against established baselines, followed by a qualitative example from the MAPS dataset. Finally, we conduct an ablation study to analyze the impact of architectural choices, input features, and other key hyperparameters on model performance.

4.1 Comparison with Baselines

A comparison of the proposed CRNN model with the baseline methods is shown in Table 1. For the unaligned MAPS dataset, a maximum timing variation of 100 ms is applied. Therefore, when calculating the metrics on the unaligned MIDI, an alignment accuracy of 100 % is observed for the 100 ms tolerance window. For more strict windows, the results decrease resulting in a baseline alignment accuracy of 27.52 % in the 10 ms window. Using DTW leads to worse performance scores than the unaligned baseline. This is caused by the randomly generated unalignment of the notes, which does not necessarily follow a monotonic warping path. The proposed CRNN on the other hand leads to an improved alignment and increases the alignment scores significantly. The accuracy for the 10 ms window is increased by over 10 % and the mean absolute onset error decreases to 26.33 ms. When training the same CRNN model on a blind transcription task (without MIDI input), performance decreases compared to using MIDI scores.

For the real world piano recordings, the unaligned MIDI files achieve an accuracy of 20.67 % in the 10 ms tolerance window. By applying DTW this score can be improved to 41.53 %. The proposed CRNN performs slightly worse than DTW for the larger tolerance windows, but performs over 10 % better in the 10 ms range.

By applying the proposed CRNN on the alignment results of DTW, the overall best results are achieved for the unaligned MAPS dataset and the real world dataset. The results are improved by over 20 % for the 10 ms tolerance window in comparison to using DTW alone. When comparing the standard deviation for the real world piano recordings, the combined approach also leads to the most consistent results in that case.

Table 1: Comparison of our proposed CRNN model with the MrMsDTW using DLNCO features.

	Mean	Median	Std	≤ 100 ms	≤ 50 ms	≤ 25 ms	≤ 10 ms
Unaligned MAPS Test Split							
Unaligned	28.37 ms	24.07 ms	22.51 ms	100.00 %	80.74 %	57.79 %	27.52 %
DTW [24]	30.62 ms	24.40 ms	25.31 ms	97.74 %	79.90 %	54.18 %	24.79 %
CRNN (Blind Transcription)	28.07 ms	16.84 ms	32.88 ms	95.07 %	82.75 %	65.86 %	36.19 %
CRNN (Ours)	26.33 ms	15.33 ms	29.03 ms	96.68 %	82.69 %	66.78 %	40.78 %
DTW [24] + CRNN (Ours)	**25.07 ms**	**14.67 ms**	**28.81 ms**	**96.05 %**	**84.37 %**	**70.12 %**	**43.29 %**
Real World Piano Recordings							
Unaligned	30.56 ms	27.78 ms	19.69 ms	98.34 %	82.68 %	47.56 %	20.67 %
DTW [24]	16.78 ms	12.90 ms	14.86 ms	99.52 %	95.09 %	78.73 %	41.53 %
CRNN (Blind Transcription)	25.26 ms	17.60 ms	26.72 ms	96.78 %	87.34 %	63.02 %	33.30 %
CRNN (Ours)	19.03 ms	13.44 ms	19.29 ms	98.33 %	89.08 %	74.74 %	53.20 %
DTW [24] + CRNN (Ours)	**12.33 ms**	**7.73 ms**	**13.56 ms**	**99.54 %**	**95.63 %**	**86.17 %**	**62.02 %**

For a more detailed evaluation, Figure 3 shows a small excerpt of a synchronization result of the CRNN architecture for an exemplary MAPS test piece. It can be seen that the red coloring, corresponding to the calculated result, often fits the blue line, corresponding to the true value, and corrects the input values represented by the black box. But there are also mistakes visible in this example. It is noticeable that repeatedly played notes are sometimes interpreted as a single, long note. Thus, fewer onsets can be detected, leading to an increased number of false negatives and a lower recall of the network's raw output. Therefore, the postprocessing step in Section 2.3 is used to overcome this issue.

4.2 Ablation Study

The following ablation study focuses on the MAPS dataset due to its larger size and greater variability compared to the real-world piano recordings, enabling more meaningful and reproducible evaluations.
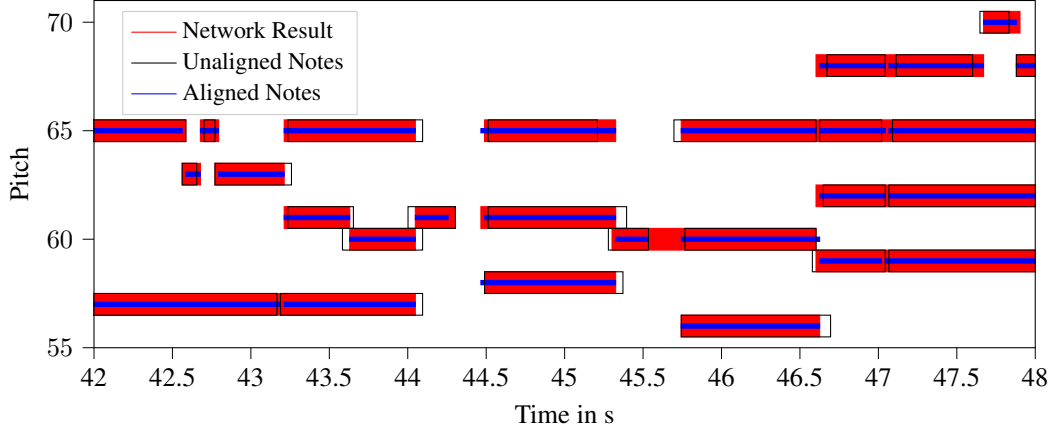
Figure 3: Alignment output of the proposed CRNN model on a MAPS test excerpt. Red blocks indicate the predicted aligned piano roll, blue lines mark ground-truth note onsets, and black boxes denote the original unaligned MIDI. The model effectively adjusts the MIDI timing to match the audio performance, correcting onset misalignments.

4.2.1 Model Architecture Variations

Results of different model architecture variations are displayed in Table 2. For the CNN variation, the RNN layer has been removed from the original CRNN architecture. This reduces the model's capabilities of processing the temporal relationship and leads to a significant drop of the alignment accuracies. A direct comparison between Gated Recurrent Unit (GRU) and LSTM cells in the bidirectional RNN layer shows that better results can be achieved when using the LSTM cells.

Table 2: Alignment results on MAPS validation split for various model architectures in percent.

Architecture	$\leq 50\,\text{ms}$	$\leq 25\,\text{ms}$	$\leq 10\,\text{ms}$
CNN	77.54	47.17	22.06
CRNN (GRU)	87.35	73.25	45.94
CRNN (LSTM)	**88.84**	**75.25**	**47.41**

4.2.2 Audio Features

Table 3 shows the impact of different input feature spectrograms on the alignment performance. Although the calculation of the CQT is not as computationally efficient as the computation of the Mel spectrogram, the CQT comes with the advantage of the same frequency bin arrangement as in the piano roll. Using the logarithmic magnitude scaled CQT leads to an improvement of about 2 % for the 10 ms tolerance window. The improvement for the other tolerance windows is less significant.

Table 3: Alignment results on MAPS validation split for various input representations with different magnitude scales in percent.

Input	Scale	$\leq 50\,\text{ms}$	$\leq 25\,\text{ms}$	$\leq 10\,\text{ms}$
Mel	Lin	88.77 %	74.07 %	45.54 %
Mel	Log	88.27 %	74.00 %	44.38 %
CQT	Lin	88.30 %	74.28 %	46.20 %
CQT	Log	**88.84 %**	**75.25 %**	**47.41 %**

4.2.3 Temporal Resolution

The temporal resolution of the piano roll and the input spectrogram is directly linked to the resolution of the output piano roll. Therefore, decreasing the temporal resolution limits the model's precision when it comes to aligning the MIDI. In Table 4, the impact of varying the number of frames per second (FPS) on the absolute onset error is shown. Increasing the resolution up to 100 FPS (10 ms per frame) has a positive effect on the performance. Increasing the resolution further does not lead to better results.

6

Table 4: Alignment results on MAPS validation split for different temporal resolutions.

FPS	≤ 50 ms	≤ 25 ms	≤ 10 ms
200	87.65 %	74.21 %	47.04 %
100	**88.84 %**	**75.25 %**	**47.41 %**
50	86.66 %	72.11 %	42.68 %
25	84.20 %	61.93 %	31.50 %

4.2.4 Maximum Human Timing Variations

For the experiments analyzed up to now, a maximum human timing variation of 100 ms is used in the unalignment procedure of the dataset creation. Experiments with real world musical pieces also yield an average human timing variation of 95 ms. While the extent of human timing variation can depend on the musician's proficiency, studies have shown that such variations are present in both professional and 5-year-old children's performances [25]. Therefore, different values for the human timing variation have been analyzed in Table 5. There is a correlation between the maximum human timing variation and the alignment performance. Smaller human timing variation results in a lower mean absolute error and a more consistent alignment result.

Table 5: Alignment results on MAPS validation split with different maximum human timing variation values.

Timing Variation	Mean	Median	Std
50 ms	18.24 ms	11.38 ms	20.78 ms
100 ms	21.60 ms	12.75 ms	26.16 ms
150 ms	24.15 ms	13.09 ms	30.49 ms
200 ms	25.84 ms	14.19 ms	32.86 ms

4.2.5 Tempo Changes

In this experiment, the tempo is changed by scaling the unaligned MIDI with a randomly sampled tempo factor in the range $[1 - \text{Max Tempo Factor}, 1 + \text{Max Tempo Factor}]$. With the introduction of tempo changes, the network's capabilities reach their limit. This is displayed in the test results shown in Table 6. It shows decreasing alignment accuracies for increasing ranges of tempo changes.

Table 6: Results on MAPS validation split with tempo changes.

Max Tempo Factor	Mean	Median	Std
0.00	21.36 ms	12.40 ms	26.35 ms
0.01	32.77 ms	18.70 ms	36.39 ms
0.05	44.20 ms	24.88 ms	47.80 ms
0.10	47.21 ms	27.54 ms	50.05 ms
0.20	48.61 ms	28.47 ms	50.67 ms
0.30	49.76 ms	30.72 ms	50.59 ms

5 Conclusion

A novel CRNN based approach for aligning piano rolls with performance audio spectrograms has been proposed. In order to synchronize piano recordings with their loosely aligned MIDI annotations, the model has been trained on a dataset compiled by augmenting existing piano transcription examples. To evaluate the model performance on real world data, six real world piano performances have been recorded. The synchronization system provides an improved alignment accuracy when compared with the unaligned MIDI files. Especially for the very strict tolerance window of 10 ms, the sole model outperforms the state-of-the-art approach DTW. When combining DTW with the proposed CRNN model, the best results can be achieved for the unaligned MAPS dataset and the real world piano pieces. In an ablation study, several model parameters have been researched and the limits of the model have been tested. The model performs best for the task of finely aligning MIDI notes that have already been loosely aligned in a preceding processing step. Hence, the best results can be achieved when performing DTW in advance.

In future work, this approach could be extended to other instruments using datasets like e.g. GuitarSet [26]. Additionally, applying the model during AMT training may improve onset precision. Exploring alternative architectures such as transformers could also further enhance performance.

References

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[2] R. B. Dannenberg and N. Hu, "Polyphonic audio matching for score following and intelligent audio editors," in *Proc. of the International Computer Music Conference (ICMC)*, San Francisco, USA, 2003, pp. 27–34.

[3] N. Hu, R. B. Dannenberg, and G. Tzanetakis, "Polyphonic audio matching and alignment for music retrieval," in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2003, pp. 185–188.

[4] S. Ewert, M. Müller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 1869–1872.

[5] T. Prätzlich, J. Driedger, and M. Müller, "Memory-restricted multiscale dynamic time warping," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 569–573.

[6] B. Gingras and S. McAdams, "Improved score-performance matching using both structural and temporal information from midi recordings," *Journal of New Music Research*, vol. 40, no. 1, pp. 43–57, 2011.

[7] C.-T. Chen, J.-S. R. Jang, and W. Liou, "Improved score-performance alignment algorithms on polyphonic music," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1365–1369.

[8] T. Kwon, D. Jeong, and J. Nam, "Audio-to-score alignment of piano music using rnn-based automatic music transcription," *arXiv preprint arXiv:1711.04480*, 2017.

[9] Y. Wang, S. Liu, and L. Guo, "An improvement on audio-to-midi alignment using triplet pair," in *1st International Workshop on Multimodal Understanding and Learning for Embodied Applications*, 2019, pp. 43–48.

[10] R. Agrawal and S. Dixon, "Learning frame similarity using siamese networks for audio-to-score alignment," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 141–145.

[11] X. Riley, D. Edwards, and S. Dixon, "High resolution guitar transcription via domain adaptation," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 1051–1055.

[12] B. Maman and A. H. Bermano, "Unaligned supervision for automatic music transcription in the wild," in *International Conference on Machine Learning*. PMLR, 2022, pp. 14 918–14 934.

[13] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the maestro dataset," *arXiv preprint arXiv:1810.12247*, 2018.

[14] J. Choi and K. Lee, "Pop2piano: Pop audio-based piano cover generation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[15] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.

[16] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," *Proceedings of the 14th Python in Science Conference*, vol. 8, pp. 18–25, 2015.

[17] M. Schwabe, S. Murgul, and M. Heizmann, "Dual task monophonic singing transcription," *J. Audio Eng. Soc*, vol. 70, no. 12, pp. 1038–1047, 2022.

[18] A. Penel and C. Drake, "Timing variations in music performance: Musical communication, perceptual compensation, and/or motor control?" *Perception & Psychophysics*, vol. 66, no. 4, pp. 545–562, 2004.

[19] C. Palmer, "Music performance," *Annual Review of Psychology*, vol. 48, no. 1, pp. 115–138, 1997.

[20] J. Devaney, "Estimating onset and offset asynchronies in polyphonic score-audio alignment," *Journal of New Music Research*, vol. 43, no. 3, pp. 266–275, 2014.

[21] F. Simonetta, S. Ntalampiras, and F. Avanzini, "Audio-to-score alignment using deep automatic music transcription," *2021 IEEE 23rd International Workshop on Multimedia Signal Processing*, pp. 1–6, 2021.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir_eval: A transparent implementation of common mir metrics." *Proceedings of the 15th International Society for Music Information Retrieval Conference*, vol. 10, pp. 367–372, 2014.

[24] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, "Sync toolbox: A python package for efficient, robust, and accurate music synchronization," *Journal of Open Source Software*, vol. 6, no. 64, p. 3434, 2021.

[25] A. Penel and C. Drake, "Sources of timing variations in music performance: A psychological segmentation model," *Psychological research*, vol. 61, pp. 12–32, 1998.

[26] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, "Guitarset: A dataset for guitar transcription." in *ISMIR*, 2018, pp. 453–460.