Improving a real-time music alignment algorithm for opera performances

Oliver Hödl
Faculty of Computer Science
University of Vienna
Department for Smart and Interconnected Living
University of Applied Sciences Upper Austria
oliver.hoedl@univie.ac.at

Dennis Gubbels
Faculty of Computer Science
University of Vienna
Vienna, Austria
d.gubbels@degu.info

Oleksandr Shabelnyk
Faculty of Computer Science
University of Vienna
Vienna, Austria
alexsaber@gmail.com

Peter Reichl
Faculty of Computer Science
University of Vienna
Vienna, Austria
peter.reichl@univie.ac.at

2023 4th International Symposium on the Internet of Sounds | 979-8-3503-8254-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/IEEECONF59510.2023.10335462

***Abstract*—Music alignment is the process of matching two versions of a musical performance. We present an adaptation of a real-time music alignment algorithm to integrate within opera.guru, a mobile software solution to enhance the experience of attending an opera performance. Our algorithm is a proof-of-concept and is based on On-line Time Warping, a derivative of the well-studied Dynamic Time Warping, and offers effective improvements. Among those is a novel technique of cost matrix pre-processing called the Axes method. The enhanced solution was tested in simulated settings to measure the alignment precision of different methods. These tests showed good results in terms of computational performance and alignment error rate. However, our test dataset is of a limited scope and more thorough verification is required. Finally, we conducted user experience tests to study the applicability of our solution within the mobile solution in a real-world setting. According to these user tests, the alignment miscalculations proved to be mostly unnoticeable to the audience.**

Keywords—music alignment, On-line Time Warping (OLTW), opera performance

I. INTRODUCTION

Digital innovations can be seen across all possible fields including rather traditional ones such as opera. Even though live performances of classical music or operas are especially cherished for their authenticity, the fact that performers play or sing naturally and in front of the audience, modern technologies have found their application in this area as well. Music alignment counts as one such technology. It can serve as a helpful aid to both performers and audience [1, 2].

Music alignment refers to the process of matching two versions of a musical performance – typically note by note. This is a challenging and non-trivial task, as the tempo can constantly vary throughout the piece. We distinguish various types of alignment. In audio-to-score alignment, the audible performance is continuously matched with the corresponding score to "track" the current position. One of the possible utilities is to automatically turn the pages of a digital score during a performance [3] or the playback of digital accompaniment adapting to the pace of the performer [2, 4]. Audio-to-audio alignment can be further subdivided in alignment of simply two recordings and of a live performance to its prior recording. While matching two recordings their full length can be utilized, whereas no helpful "future" information is available during a live performance. This paper focuses on alignment of live polyphonic music, particularly operas, which imposes additional challenges compared with monophonic non-live music alignment.

In this paper, we analyse the current development in music alignment, as well as present a demonstrative example of applying and enhancing existing techniques based on Dynamic Time Warping in the context of opera.guru, a mobile software solution to enhance the experience of opera audience.

opera.guru [1] is a research project that enriches the experience of opera attenders. It offers them a mobile client application, which delivers aid, primarily in the form of text, to opera listeners and helps them to better follow the plot during a live opera performance. Following the plot of an opera can be challenging especially for those not regularly attending opera performances. At the same time, it is not necessary (or required by everyone) to follow an opera performance word by word to understand the plot (especially of someone focuses more on the music). With opera.guru, opera houses can decide whether they want to provide subtitles or plot summaries, and users can choose among different languages, on a mobile device during an ongoing opera performance. Optionally, media such as images or videos are supported. Not only does it help to overcome the language barrier, but it also makes operas more comprehensible for non-regular opera goers.

Though it mainly targets operas, opera.guru can be used during any live event. Along with the mobile applications for end-users, opera.guru offers a web-based content management system for administrators and operators.

During the recent developments of opera.guru, an automatic music alignment solution has been implemented, allowing the content to be delivered to the mobile clients in real-time fully automatically. It eliminates the need for the human operator to manually push the content, as it was necessary before.

In the remainder we present the implemented alignment algorithm, its integration into opera.guru, as well as the evaluation results from a technical and user experience point of view.

II. STATE OF THE ART

Some of the first developments in music alignment were introduced by Dannenberg [4] and Vercoe [5] who proposed live audio-to-score alignment algorithms in the context of digital accompaniment. These early approaches tracked specific monophonic instruments in symbolic form based on

[1] http://opera.guru (last access 31.07.2023)

string-matching techniques. Puckette [6] describes an algorithm based on an ordered note list where the notes played are detected using pitch tracking. Grubb & Dannenberg [7] were pioneers in tracking ensembles – polyphonic music. Their solution was based on separately tracking each individual musician using multiple microphones combined with a pitch-to-MIDI converter, however this approach proved not to be well scalable.

In 1998, Grubb presented a probabilistic framework for audio tracking [8, 9]. Later various stochastic models, introduced among others by Raphael [10] and Cano et al. [11] started adopting the so-called Hidden Markov Models (HMM). According to Rabiner & Juang [12], an HMM describes a system in a probabilistic fashion using a state-model. Different variations of these probabilistic methods have emerged, utilizing various techniques such as Specialised State Space [13], Segmental Conditional Random Fields [2], Dynamic Bayesian Networks and Particle Filtering [14], as well as Linear Dynamic Systems [15], however none of those seem to have achieved significant scientific resonance so far.

HMM-based solutions have been continuously improved over the years and current models are rather hybrid ones based on semi-Markov states instead, allowing explicit modelling of secondary factors such as tempo. Prominent and successful commercial systems based on hybrid HMM are Metronaut (Antescofo)[2] and Tonara[3].

On-line Time Warping (OLTW), introduced by Dixon, is a variant of Dynamic Time Warping (DTM) [16] for processing audio in real-time. DTW and HMM are not so different as it may appear at first glance; as argued by Fang, DTW can be modelled using an HMM [17]. The OLTW-based solutions for music alignment also saw several improvements, mostly related to additional systems to intervene in the case of common tracking issues. OLTW is particularly well suited for live audio-to-audio alignment and thus constitutes the cornerstone of our solution's implementation.

III. Conceptual Design and Basic Algorithm

Hereafter we examine the design of the proposed solution and highlight its main characteristics. To minimize the preparation time, the system has only two prerequisites:

A reference recording of the target performance as "compare-to" data. This can be a recording of a rehearsal or a previous performance which should ideally be coming from the production[4]. We are aware of possible major structural differences caused by improvisations such as secco recitative, handling of which is considered out of scope.

Timecodes, corresponding to the reference recording, must be set beforehand, meaning the system should know at what time to push data, e.g., textual aid, to the users.

Further we distinguish following important design characteristics:

- As opera.guru is primarily designed to deliver plot descriptions and not subtitles, the deliveries can tolerate a reasonable deviation within a few seconds.
- The target audio performance is deliverable as live microphone input.
- The software's architecture and interface allow easy integration with other software solutions and is independently deployable.

The solution is implemented based on the OLTW algorithm introduced earlier and is centred around a so-called cost matrix. Its axes represent the frames of the target and reference audio streams respectively, and the cells show the calculated cost of alignment between them. The cost is determined based on the selected audio feature using the Mel Frequency Cepstral Coefficients (MFCC) which turned out to be effective for our purposes, in line with findings from the literature [22].

The algorithm chooses the least expensive path throughout the matrix. We denote it as the alignment path. The audio features, i.e., cost, of each audio frame captured by the microphone are appended to the target buffer. Alongside which a pre-calculated reference buffer exists. Since our focus lies on the improvements of OLTW, and its application, its basic implementation is not discussed in further detail.

The technological stack comprises of Python programming language, Librosa library version 0.9.1 for calculating MFCC audio features and ffmpeg [18] for pre-processing audio files. The solution is a standalone application and communicates with opera.guru via a REST interface, which allows seamless integration with any other software system.

IV. Algorithm Improvements

Taking the OLTW algorithm as the basis we (1) improved heuristics for finding the most optimal cost path, and (2) developed a novel method to pre-process, i.e., optimize, the cost matrix.

A. *Cost-path-finding heuristics*

In the base implementation the algorithm always follows the cells in the cost matrix with the lowest cost. If there are significant differences between the target and reference streams, the lowest cost path may not be the actual correct alignment path. As shown in Figure 1, choosing only cells with the lowest cost can steer the alignment path in the wrong direction. At frame 500 the alignment is roughly correct, but at frame 820 the alignment path is completely wrong. To combat this problem, we introduce the following three heuristics for algorithm advancement.

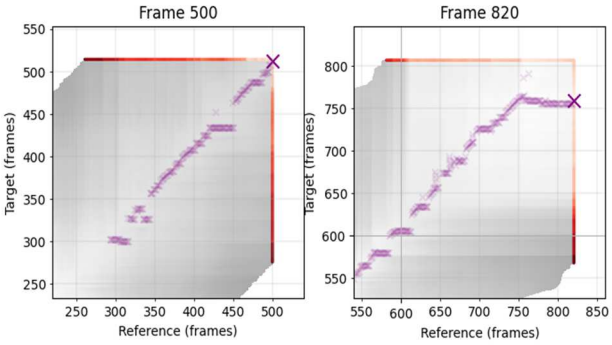

Fig. 1. Cost matrices showing how the single lowest cost point may lead to inaccurate alignment.

The *Minimum mean* heuristic calculates the average position of n adjacent lowest cost cells. A side effect of this approach is that the minimum mean point can lay further back

[2] https://www.metronautapp.com (last access 31.07.2023)

[3] https://www.tonara.com (last access 31.07.2023)

[4] By "production" we mean the same musicians, singers, conductor, etc.

from the current position, thus, not in the final row or column, as shown in Figure 2, frame 812. The Figure depicts the algorithm's behaviour using minimum mean with n = 5 compared to the standard implementation demonstrated previously on Figure 2. Even though frame 812 shows success by turning the alignment path towards the right one, other frames nearby, demonstrate that this approach is not always effective.

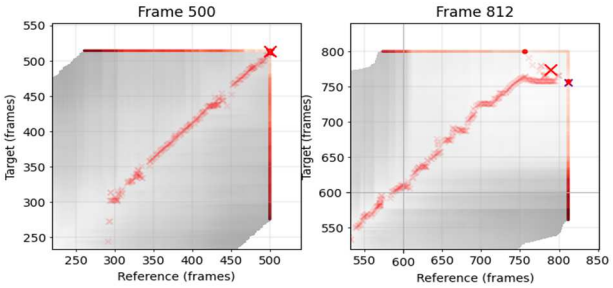

Fig. 2. Example demonstrating 5 minimal points and mean.

The *Rolling mean* averages across frames, rather than within a single frame. It takes the average position of the last n frames. Figure 3 shows the effect of applying rolling mean with n = 100. One of disadvantages of rolling mean is that it uses historic data and is therefore slow to react to changes in tempo. Additionally, fine details in alignment are smoothed out, affecting not only errors, but also useful information.

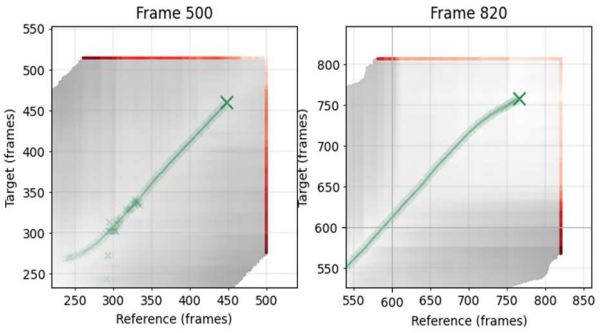

Fig. 3. Example demonstrating Rolling mean.

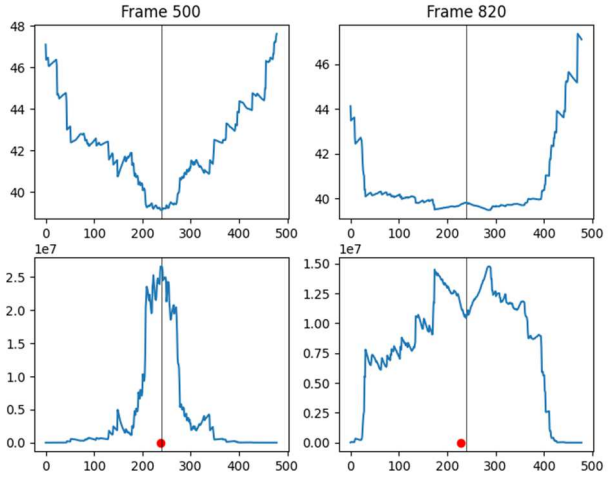

Fig. 4. Pre-calculation steps of Weighted mean.

The *Weighted mean* takes the cost-weighted average position of all cells in the last frame. The intuition for why the weighted mean should be an effective heuristic is that even when the alignment path is not clearly present, the cost matrix is largely symmetric around the correct alignment. Taking the weighted mean allows us to find this point of symmetry and find the alignment no matter how far the lowest cost points are removed from this alignment path. As a first step, the algorithm takes the cost values of the last column and the last row representing the last frame. The upper row on Figure 4 shows the result of this operation for three example frames. The vertical line denotes the current position; the lefthand side represents the final row and the righthand side the final column. In the next step the values are inverted so that the cells with the lowest cost get the highest weight. Additionally, the values are raised to the 8th power to highlight the low-cost values. Finally, the weighted mean position is calculated, as indicated by the red dot on the lower row in Figure 4. The position is projected back onto the cost matrix and taken as the advancement direction. Figure 5 shows the result of the Weighted mean.

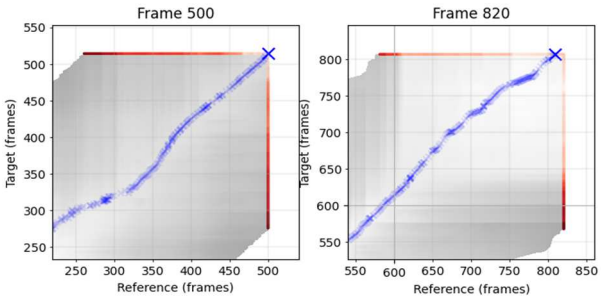

Fig. 5. Example demonstrating Weighted mean.

B. *Cost Matrix Pre-processing*

In music alignment, it is typical for a cost matrix to reveal distinguishable "lines" of low-cost values parallel to the axes, as can be seen on Figure 6 (see lefthand side matrix). These lines often cross the correct alignment path and therefore can steer the alignment path in the wrong direction. Next, we introduce two pre-processing methods to reduce the mentioned low-cost lines.

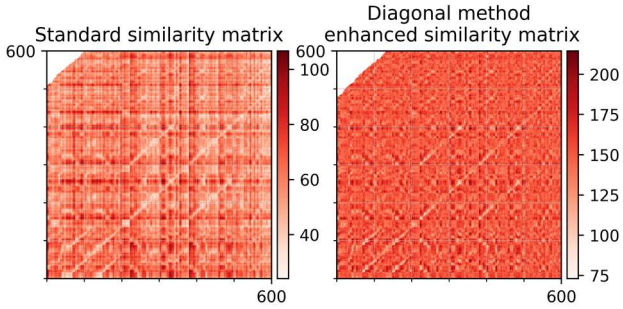

Fig. 6. Cost matrix before and after pre-processing.

The *Axes method*, developed fully by us, aims to eliminate the axis-parallel lines by raising the costs. It is achieved by using previous values along the axes. The processing is performed on every frame after the cost matrix is filled. It is comprised of three steps; the first two are described hereafter using the formulas and in Table 1:

$$M'[t,u] = offset - a \cdot M[t-1,u] - b \cdot M[t-2,u] - c \cdot M[t-4,u] - d \cdot M[t-8,u] - e \cdot M[t-16,u] \quad (1)$$

$$cost = M'[t,u] + offset - a \cdot M'[t-1,u] - b \cdot M'[t-2,u] - c \cdot M'[t-4,u] - d \cdot M'[t-8,u] - e \cdot M'[t-16,u] \quad (2)$$

TABLE I. EXPLANATION OF FOMULAS (1) AND (2)

Table Head	Explanations
M	More table copy
M´	Modified cost matrix
t	Target frame index. The indices are spaced by powers of 2 to increase coverage and reduce processing time
u	Reference frame index
offset	Constant to keep the result of the consecutive subtraction operation positive
a, b, c, d, e	The used values are 0.1, 0.25, 0.2, 0.15, 0.1

Finally, the costs are squared and divided by a constant to reduce the values. This helps to increase the weight of value differences and prevents the values from ending up negative. The Axes method is designed to satisfy the following rules shown in Table 2 which should be read as "if current cost is x and preceding costs are y, then resulting cost will be z".

TABLE II. RULES FOR AXES METHOD

Case	Current	Preceding	Resulting
1	Low	High	Low
2	High	High	Average
3	Low	Low	Average
4	High	Low	High

Consequently, after the pre-processing the resulting cost is low only in case 1, which occurs mostly on diagonal lines such as the alignment path.

Because the algorithm requires historic data, which are unavailable in the beginning, the first n rows and columns, in our case 16, are prefilled with value 1000, except for the diagonal, where the values are set to 0. We are aware that it results in incorrectly high costs around the diagonal due to applied squaring, but it proved to have no negative impact on the alignment correctness.

Müller and Kurt [19] introduced a pre-processing method we refer to as the *Diagonal method*. Though this method is designed for offline music alignment and is difficult to efficiently use in real-time, we included it for comparison purposes. The diagonal method consists of two steps:

$$dL(n,m) = \frac{1}{L}\sum_{L=0}^{L-1} d\big(t(n+l), u(m+l)\big) \quad (3)$$

$$dL^{min}(n,m) = min_r \frac{1}{L}\sum_{L=0}^{L-1} d\left(t(n+l), w_r\left(\frac{m}{r}+l\right)\right) \quad (4)$$

L is a length parameter, t and u are the target and reference audio streams; n and m are the indices. The second step accounts for tempo variability. This is done by explicitly calculating cost values for different simulated tempo ratios and choosing the lowest cost option. The tempo is simulated by changing the hop length, i.e., the index frequency. The variable w, used in the second step, is a set of reference streams at different simulated ratios r.

In our implementation of this method, we used 7 ratio choices in the range from 0.76 to 1.24 in equal steps of 0.08. The length parameter was set to 16. Figure 7 shows the result of applying the diagonal method to a similarity matrix.

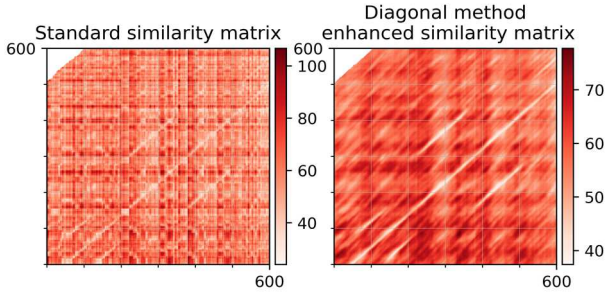

Fig. 7. Cost matrix before and after pre-processing.

V. EVALUATION

To evaluate the proposed solution, we have conducted two types of tests:

- Simulation: simulating recordings as live input, thus, without audience.
- User-centred: studying the user experience with a live audience.

As opera test data, recordings of Giuseppe Verdi's Rigoletto were chosen, originating from different productions (see Table 3), which makes the alignment process more challenging. During the transition between the first two scenes the recordings revealed significant discrepancy. Since handling applause, silence, etc. is out of scope of our solution, the reference recording was modified by cutting out a section 35 seconds long.

TABLE III. OPERA TEST DATA

Opera House	Director	Year	Start
Semperoper Dresden	Nikolaus Lehnhoff	2008	3:32
Vienna State Opera	Pierre Audi	2016	6:30

Additionally, tests were conducted using western classical music as a less challenging alternative to testing with opera performances. The test pair was Beethoven's ninth symphony, first movement, from years 1962 and 1983, both performed by Berlin Philharmonic Orchestra, conducted by H. von Karajan. We used annotation time stamps provided by Gadermaier & Widmer [22] as reference data for our evaluation.

A. The simulation of advancement heuristics

Figure 8 shows the deviation from manual alignment of the four previously discussed advancement heuristics in the simulated setting of Beethoven's ninth symphony. On the visualization the vertical axis represents the deviation, in seconds, from manual alignment. The horizontal axis shows the progression throughout the piece, again in seconds. Except for three sections with larger error spikes, all variants perform similarly well. At the start, the fluctuation seems to affect all variants. The spike in the middle, around the 300th second, shows the superiority of rolling mean and weighted mean. After the avoided upward spike, the rolling mean variant does show a prolonged downward error which is also avoided by the weighted mean variant. Just beyond the 500th second, all variants show a significant error spike, however the weighed mean variant demonstrates the smallest deviation. According to the results, the weighted mean heuristic showed the best precision.

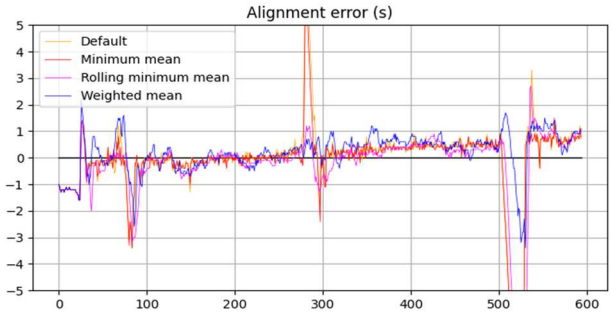

Fig. 8. Deviation from human-set alignment.

B. *Simulation of similarity matrix pre-processing*

The opera test data was used for testing the pre-processing methods to create more challenging conditions. Figure 9 demonstrates the accuracy of the two previously introduced diagonal and axes pre-processing methods, as well as no pre-processing, denoted as control, based on the standard OLTW advancement algorithm. Both the control and the diagonal methods lost the alignment path completely and were unable to recover, even though the diagonal method could perform correctly longer than the control one. The axes method proved to be able to follow the correct alignment until the end.

Fig. 9. Showing difference after applying pre-processing steps.

Another test of the same pre-processing methods, depicted on Figure 10, was conducted based on the weighted mean advancement heuristic. The axes method performed slightly worse compared to the previous test. The total deviation is 11.3 seconds, compared to 7.7 seconds before, although the accuracy decline is not particularly noteworthy. The performance of the diagonal method improved markedly.

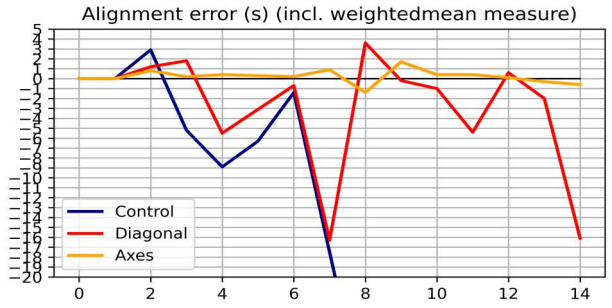

Fig. 10. Pre-processing steps combined with Weighted mean.

Figure 11 shows the cost matrices in the point in time when the diagonal method temporarily loses the correct alignment path. Even though the correct alignment path is strengthened in the "diagonal" matrix, vertical and especially horizontal lines of low-cost cells are still present. This steers the alignment in a wrong direction resulting in major deviation errors as seen before.

Overall, the results clearly show that matrix pre-processing, especially the axes method, can significantly improve the alignment accuracy.

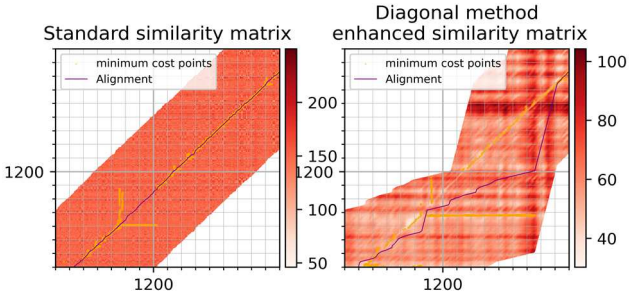

Fig. 11. Axes and Diagonal pre-processing method comparison.

C. *User experience tests*

A user experience test was held with a total of 93 students within a lecture at the University of Vienna, Austria. The audience was secretly divided into two groups. Following a random assignment, one group received content on their phones pushed by the automatic alignment solution, another one by a human operator. The audience was shown a section of the pre-recorded Rigoletto performance using the projector and the speakers of the lecture hall. The microphone, a Blue Yeti[5], used for the automatic alignment was placed in front of the room. After the test the students filled out a survey counting 38 responses for automated tracking and 44 responses for the manual tracking. 11 responses were discarded due to connectivity issues, connections from multiple devices, unserious answers, etc.

Table 4 shows the results of the survey conducted after the trial sessions. A scale from 1 to 5 was chosen; higher number represents higher satisfaction. Overall, according to the results, there is no difference in perceived subjective user experience between the two versions of audio alignment. Both methods perform well enough to keep the users' average satisfaction high, i.e., above 4.1.

TABLE IV. SURVEY RESPONSES

Nr	Question	Manual		Automated	
		Avg.	*Sd.*	*Avg.*	*Sd.*
1	How did you enjoy using the app?	4.2	0.9	4.1	0.9
2	Did you find the app helpful for following the plot of the opera scene?	4.8	0.7	4.7	0.6
3	Did you find it easy to keep track of both the projected opera scene and the app?	4.2	1.1	4.1	0.9
4	Overall, how good was the timing of messages shown in the app?	4.2	0.8	4.3	0.8
5	Overall, how consistent was the timing of messages shown in the app?	4.2	0.9	4.4	0.7

[5] https://www.bluemic.com/en-gb/products/yeti/ (last access 31.07.2023)

When the students were asked to determine which group they were in, respondents from both groups answered virtually identically. Although, we have to mention that while the solution worked well on the section tested, this is not guaranteed to be the case for the whole opera performance. Our simulated performance of the whole opera showed that a few short sections still produced large deviations of over 30 seconds, partially due to the mismatched recordings, but also partially due to shortcomings in the algorithm. Nevertheless, as a proof-of-concept of the user experience in relation to automatically show subtitles our algorithm worked well.

VI. Conclusions

In this paper we discussed the problem of real-time automated music alignment and developed an algorithmic solution to combat it. The solution utilizes MFCC audio feature extraction to make audio pieces comparable. Moreover, it is based on a well-known technique for audio-to-audio alignment called OLTW, which required further development to achieve the desired level of accuracy. Thus, we introduced and tested some new and some already known extensions to it. The evaluation showed that the Weighted mean method for the algorithm's advancement can provide a reasonably high alignment accuracy compared to other techniques. It calculates a weighted mean position over the cells in the final row and column of the cost matrix, thus, trading effectively a higher median error for a lower maximal one, which is preferrable in most cases.

Additionally, we have dived deep into increasing the alignment accuracy also by pre-processing the similarity matrix – the heart of the alignment algorithm. Among the tested approaches, the so-called Axes method yielded overall the best results by reducing the influence of lines parallel to the axes caused by repeated structures in the music.

According to the user experience trials, no major differences were reported using either manual or automated alignment. This proves, at least subjectively, that the introduced solution is suitable for real-world applications. Nonetheless, a thorough quantitative analysis using a richer dataset is required to be able to draw conclusion on general applicability of the solution.

VII. Future Work

We conducted tests on only one opera performance, which has more in common with polyphonic music. Therefore, the future work should yield in testing the solution with a larger dataset and with opera performances in their original form. Ideally, the opera pieces from various periods and styles should be tested to build a clear picture of the overall effectiveness and applicability of the automated system.

Previously mentioned research [20, 21] in automatic handling of applause and silence would be a meaningful extension to our solution, eliminating the need to manually edit the audio reference recording. Moreover, it makes the alignment robust in case either of those suddenly appears in the live performance.

Apart from the possibility to generally improve the alignment algorithm by introducing new techniques and approaches, another idea is to take advantage of the constraint that accuracy is only relevant at the synchronization points, meaning, when content should be pushed to the user devices.

Acknowledgment

Many thanks to Kaspar Lebloch and Hannes Weisgrab for the support in planning and conducting field studies. This work was funded by the Austrian Science Fund (FWF) [grant number WPK 126-G].

References

[1] M. Dorfer, A. Arzt and G. Widmer, "Towards Score Following in Sheet Music Images," in Proceedings of ICASSP, 2016.

[2] S. Sako, R. Yamamoto and T. Kitamura, "Ryry: A real-time score-following automatic accompaniment playback system capable of real performances with errors, repeats and jumps," in Proceedings of the International Conference on Active Media Technology, 2014.

[3] A. Arzt, G. Widmer and S. Dixon, "Automatic Page Turning for Musicians via Real-Time Machine Listening," in Proceedings of ECAI, 2008.

[4] R. B. Dannenberg, "An on-line algorithm for real-time accompaniment," in Proceedings of ICMC, vol. 84, pp. 193-198, 1984.

[5] B. Vercoe, "The synthetic performer in the context of live performance," in Proceedings of ICMC, 1984.

[6] M. Puckette and C. Lippe, "Score following in practice," in Proceedings of ICMC, 1992.

[7] R. B. Dannenberg and L. Grubb, "Automating ensemble performance," in Proceedings of ICMC, 1994.

[8] L. Grubb and R. B. Dannenberg, "A stochastic method of tracking a vocal performer," in Proceedings of ICMC, 1997.

[9] L. Grubb, "A probabilistic method for tracking a vocalist," Carnegie-Mellon University, Department of Computer Science, Pittsburgh, PA, 1998.

[10] C. Raphael, "Automatic segmentation of acoustic musical signals using hidden Markiv Models," in IEEE transactions on pattern analysis and machine intelligence, vol. 21, no. 4, pp. 360-370, 1999.

[11] P. Cano, A. Loscos and J. Bonada, "Score-performance matching using HMMs," in Proceedings of ICMC, 1999.

[12] L. Rabiner and B. Juang, "An introduction to hidden Markov models," in IEEE ASSP Magazine, vol. 3, no. 1, pp. 4-16, 1986.

[13] Z. Duan and B. Pardo, "A state space model for online polyphonic audio-score alignment," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011.

[14] F. Korzeniowski, F. Krebs, A. Arzt and G. Widmer, "Tracking rests and tempo changes: Improved score following with particle filters," in Proceedings of ICMC, 2013.

[15] G. Xia, Y. Wang, R. B. Dannenberg and G. Gordon, "Spectral Learning for Expressive Interactive Ensemble Music Performance," in Proceedings of ISMIR, 2015.

[16] S. Dixon, "Live Tracking of Musical Performances Using On-line Time Warping," in Proceedings of the 8th International Conference on Digital Audio Effects, 2005.

[17] C. Fang, "From Dynamic Time Warping (DTW) to Hidden Markov Model (HMM)", University of Cincinnati, 2009.

[18] S. Tomar, "Converting Video Formats with ffmped," Linux Journal, 2006.

[19] M. Müller and F. Kurth, "Enhancing Similarity Matrices for Music Audio Analysis," in Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing, 2006.

[20] C. Brazier and G. Widmer, "Towards Reliable Real-time Opera Tracking. Combining Alignment with Audio Event Detectors to Increase Robustness," in Proceedings of the 17th Sound Music Computing Conference, 2020.

[21] H. Sak, A. Seniour and F. Beaufays, "Long Short-term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," in Proceedings of Interspeech, 2014.

[22] T. Gadermaier and G. Widmer. "A Study of Annotation and Alignment Accuracy for Performance Comparison in Complex Orchestral Music," in Proceedings of ISMIR, pp. 769–775, Delft, The Netherlands, Nov. 2019.