Automating Audio Description for Live Theater

Using Reference Recordings to Trigger Descriptive Tracks in Real Time

Dirk Vander Wilt[*]
New York University
New York, NY
dirk.vanderwilt@nyu.edu

Morwaread Mary Farbood[*]
New York University
New York, NY
mfarbood@nyu.edu

ABSTRACT

Audio description is an accessibility service used by blind or visually impaired individuals. Often accompanying movies, television shows, and other visual art forms, the service provides spoken descriptions of visual content, allowing people with vision loss the ability to access information that sighted people obtain visually. At live theatrical events, audio description provides spoken descriptions of scenes, characters, props, and other visual elements that those with vision loss may otherwise find inaccessible.

In this paper we present a method for deploying pre-recorded audio description in a live musical theater environment. This method uses a reference recording and an online time warping algorithm to align audio description with live performances, including a process for handling unexpected interruptions. A software implementation that is integrated into an existing theatrical workflow is also described. This system is used in two evaluation experiments that show the method successfully aligns multiple recordings of works of musical theater in order to automatically trigger pre-recorded, descriptive audio in real time.

CCS CONCEPTS

• **Human-centered computing** → **Accessibility technologies**; • **Applied computing** → *Performing arts*; *Sound and music computing*.

KEYWORDS

audio description, blind, visually impaired, accessible, musical theater, time warping

ACM Reference Format:
Dirk Vander Wilt and Morwaread Mary Farbood. 2019. Automating Audio Description for Live Theater: Using Reference Recordings to Trigger Descriptive Tracks in Real Time. In *Audio Mostly (AM'19), September 18–20, 2019, Nottingham, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3356590.3356603

1 INTRODUCTION

Audio description (AD) is an accommodation used by people who are visually impaired or blind. It is a spoken description of the visual elements of an accompanying work, providing an accessible alternative to obtaining information that sighted individuals at the same time and place may obtain visually. Therefore, it is temporal as well as descriptive. At live theatrical events that provide AD, such as some Broadway shows, patrons use headphones and a personal receiving device, so as not to disrupt other theatergoers. For an overview on the art and practice of AD, see Fryer [6] and Snyder [13].

Live theatrical events pose an interesting problem for AD services. Like fixed media, the descriptions must be timed appropriately so as not to disrupt other aural aspects of the show (musical numbers, dialogue, etc.) [6, 13]. However, since repeated live performances are by design never identical, a fixed AD track cannot be aligned in advance. In live situations, either a professional audio describer describes the visual elements in real time, or a system is created to allow pre-recorded AD to be triggered [8]. Developing and deploying this type of service is expensive and time-consuming. A recent study showed that media producers view AD as "a costly service with no revenue potential," [11]. According to Szarkowska [14], "A lengthy preparation process and high production costs are among the greatest obstacles to the wider availability of audio description."

This paper proposes an inexpensive and novel method to trigger AD for a live theatrical performance by only using audio obtained from a show's previous performance. The process described here warps the audio from the live show in real time to a reference recording using an established online time warping algorithm [4]. This method is a step towards

[*] Music and Audio Research Laboratory (MARL), Department of Music and Performing Arts Professions, Steinhardt School of Culture, Education, and Human Development

AM'19, September 18–20, 2019, Nottingham, United Kingdom

ACM ISBN 978-1-4503-7297-8/19/09…$15.00
https://doi.org/10.1145/3356590.3356603

being able to reduce the cost of deploying live theatrical audio description, thus making it more available to visually impaired people.

2 METHOD

Live audio alignment has been successful in music score following [1] and audio-transcription alignment [7] tasks. In this implementation, a reference recording is aligned to live input using an online time warping algorithm [4]. During the live alignment, descriptive audio tracks based on their pre-aligned position in the reference recording are also aligned and played back to blind and visually-impaired audience members.

First, a performance of the entire production is recorded in advance. Relevant features from that audio are then extracted and stored in frames of vectors. A second audio track is also created, containing only the AD which aligns to that recorded performance. The descriptive track is broken up into multiple smaller tracks such that one sub-track contains the audio for a single described event within the recorded performance. The points where each sub-track should be triggered are marked by frame number and stored in array F, where $F(1...x)$ represents the frame at which AD number x should be triggered. Once the marks, descriptive track, and extracted features of the reference recording are obtained, the live alignment may begin.

In the system described here, Mel-frequency cepstrum coefficients (MFCCs) are extracted from both the live input and reference recording. MFCCs are used to analyze audio for human speech and music, both of which apply in this system. The code implemented to extract MFCCs is based on [5]. In addition to 13 MFCC coefficients per frame, a first-order difference from the previous frame is appended totaling 26 features per frame. The system uses a Hamming window and 40 Mel filters. MFCC feature vectors are extracted with a frame length of 100 ms and a hop size of 40 ms. For a description of MFCC feature extraction, see [9] and [10].

Online Time Warping

Dynamic Time Warping (DTW) uses dynamic programming to recursively compare two time series U and V of lengths m and n. The output is an m- by- n matrix D where each element $D(x, y)$ is a cumulative (in this case Euclidean) distance between $U(x)$ and $V(y)$. The value of each cell is the cumulative “path cost" from $D(1, 1)$ up to that point [12]. Every cell in the matrix is calculated by obtaining the distance between $U(i)$ and $V(j)$, and adding it to the least of one of three adjacent (previous) cells. In this way, the cumulative path cost between $D(1, 1)$ and the current cell is determined. The smaller the cumulative cost, the better the match up to that point. When the whole matrix is calculated through $D(m, n)$, backtracking the smallest distance back to $D(1, 1)$ will be the path which relates the closest points of U to V. This algorithm requires U and V to be fully available in advance and runs in quadratic time, and so is unsuitable for a live application.

An online time warping algorithm, developed by Dixon [4], requires only one of the series to be fully available in advance, while the other series may be obtained in real time. The algorithm outputs a similar matrix D, but it builds as the input is received, one row and/or column at a time, and only in the forward direction. Also, since it is only able to estimate the total size of the resulting matrix (the total length of the live input is unknown), it is instead bounded by a constant, which is determined in advance. Thus, it does not have the advantage of being able to backtrack from future input.

In online time warping, whenever a new frame of input is received in real time as $U(t)$, where t is both the current live input frame and the total number of input frames received so far, the system must determine whether to add another row, or column, or both, to matrix D. It does this by checking all the path cost's previous c elements of the current row t and column j of the matrix. If the lowest cost is in that row, it increments the row. If the lowest cost is in that column, it increments the column. If the lowest cost is $D(t, j)$, the current cell, it increments both. Also, if a row or column has been incremented $MaxRunCount$ times successively, it then increments the other, thus preventing the system from running away in one direction. This implementation sets $c = 500$ and $MaxRunCount = 3$ as described in [4].

Indices t and j are pointers to the current real-time position in U and V. At any point $U(t)$ (the current frame in the real-time input), the value of j is the current estimated location in V. Since index t is the current live input frame, it will always increment steadily. Index j, however, will increment based on where the online time warping algorithm estimates the current temporal location to be in the reference recording. AD is inserted based on the real-time current value of j.

Interruptions

In addition to adapting to a performance in real time, automatic AD must also account for unexpected interruptions. If an intermission, audience applause, or other such break is not included in the reference recording (which will likely be the case if a reference recording of a show's rehearsal is used), then the marks will mis-fire until sufficient time has passed for the marks to re-align successfully once again. The longer the unexpected interruption, the longer it will take for the marks to re-align. Thus, a mechanism is needed to handle these situations accordingly.

To identify interruptions, the system proposed here is trained to listen for specific audio identifiers that are representative of typical interruptions. In addition to calculating t (live input frame) and j (position in reference) with online time warping, incoming frames of live input will also be checked against these interruption frames. If the live input is closer to the interruption frames, then the system will stop incrementing j. Once stopped, the system will check against the next frames of the reference recording ($j + 1, j + 2$, etc.) with the live input. If it determines that the live input more closely matches the reference, then j will increment and the alignment will continue. During pauses, t will continue to increment since live input is still being received.

Training data is a short audio clip of representative audio that occurs during interruptions (room ambience, audience chatter, etc.). It is obtained in a manner identical to obtaining the reference recording and live input (same sample rate, hardware setup, etc.). The length of the training data need not be very long. In this implementation, multiple audio clips are obtained to represent audio from typical theatrical interruptions, such as audience murmur during show intermissions. Each clip is then converted into a feature set in the exact manner as the reference recording, and stored in S, where $S(n)$ is the feature vector at frame n.

Detecting interruptions during live alignment works as follows: Each time the online time warping algorithm is about to increment j, a check against the interruption feature set is performed. The system finds the path cost between $U(t-c...t)$ and $S(1...1+c)$. In this way, the system determines the cost of aligning the previous c frames of live input with c frames of the interruption feature set. Then, the system finds the path cost between $U(t-c...t)$ and $V(j...j+c)$, which determines the cost of aligning the previous c frames of live input with the upcoming (future) c frames of the reference recording. If the cost is lower with aligning the live input to the reference, then there is likely no interruption. Conversely, if the cost is lower with aligning the live input with the interruption features, then there is likely a pause. The cost is calculated by finding the distance between two feature vectors at point c, then adding it to the distance between feature vectors at point $c + 1$, etc., until the full cost has been calculated.

If it has been determined that an interruption is likely at frame t, then a count of how many consecutive frames of a likely pause is incremented by 1. If the number of consecutive "pause likely" frames increases beyond a set threshold, then the system will not increment j (meaning the live alignment is effectively paused). Otherwise, the system will increment j and the live alignment continues. Similarly, once a pause has been initiated, the system will remain paused until a threshold of consecutive "pause not likely" events has been reached. Increasing or decreasing the consecutive count threshold is a way to adjust the sensitivity of the system.

3 THE ALIGNMENT PROCESS

When the show begins, V contains the reference recording features, U is empty, $t = 1$ and $j = 1$, which are references to the indices of the first frames of U and V, respectively. Each time t increases (meaning the live recording has progressed by one frame), the new value of j is determined, based on the online time warping algorithm. If the algorithm determines that U is progressing faster than V at that moment, then increment t by 1 while a new frame of live input is received. If U is slower than V, then increment j by 1. If they are both moving at the same speed, then both t and j are incremented. Index j will keep incrementing (or not increment if the input is slower) until it matches t's estimated location, and a new t (live input frame) is obtained. The AD number x is triggered when $j = F(x)$. In this way, the descriptive tracks are able to align with the live performance based on the online time warping's estimation of the current index j of the reference.

Since the actual size of the matrix is unknown, an empty square matrix of 40,000-by-40,000 was created in the current implementation, which holds approximately 25 minutes of data on either t or j given the feature extraction parameters presented earlier. During the alignment, when one index is incremented up to the size of the matrix, the matrix is reset and the path cost returns to 0. In this manner, the alignment can run indefinitely, and the calculated path cost does not increase indefinitely. Other techniques may be used in different implementations, including smaller matrix sizes or a matrix that follows a window around the warp path and whose values are overwritten as the tracking progresses.

MFCCs for each minute of audio were extracted in less than 2 seconds while running on a 2.7 GHz MacBook Pro using a C/C++ implementation. Offline tests of the online algorithm were able to process one hour of alignment (including extraction and matrix calculation) in about 5 minutes. This process therefore runs comfortably in a real-time scenario.

4 EVALUATION

To evaluate this method, two different audio recordings of the same theatrical productions were used, with one recording as a reference and the other as live input. Markers were placed manually in both recordings to represent specific moments in the production (such as lines of dialogue or musical cues). The algorithm was then run in real time and the mark locations found during the alignment were compared to the actual mark locations in the live input.

In the first evaluation experiment, two recordings of Gilbert and Sullivan's *H.M.S. Pinafore* were used: a D'Oyly Carte recording from 1949 and a Malcolm Sargent recording from 1958. In both recordings 213 specific points were marked;

Table 1: Results of the two evaluation experiments. Marks refers to the total number of annotated marks for each show; values in the < 1, 2, 5 sec columns indicate the percentage of marks found within 1, 2, and 5 seconds of the ground truth; St. dev refers to standard deviation of the differences between the found marks and the ground truth.

	Marks	< 1 sec	< 2 sec	< 5 sec	St. Dev
Pinafore	213	75.59%	85.92%	93.90%	2.68 sec
Blonde	169	69.23%	78.70%	86.98%	3.32 sec

these points were meant to simulate where AD may be triggered. This experiment used the D'Oyly Carte version as the reference and the Malcolm Sargent version as the live input.

After completing the alignment, the ground truth was compared with the marks automatically located by the algorithm. A total of 161 marks (76%) were found less than 1 second from the mark's actual location in the reference; 183 marks (86%) were found less than 2 seconds from the actual location; and 200 marks (94%) were less than 5 seconds. The mean difference between the marks indicated by the algorithm and the ground truth was 1.2 seconds (SD = 2.68 seconds).

To test the algorithm in a more realistic situation, a second experiment using two recordings with notable, audible differences were obtained: the Broadway (2007) and London (2010) cast recordings of *Legally Blonde, The Musical*. The London version was recorded in front of a live audience and contains audience noise, laughter, ambience, etc. that is not present in the Broadway recording. The only alteration made to the recordings was the removal of one track from the London version because it was out of order in the Broadway recording.

In both versions of *Legally Blonde*, 169 locations were manually marked, and the alignment was run in real time using the London recording as the reference, and the Broadway recording as the live input. The results showed that 117 marks (69%) were found within 1 second of the reference, 133 (79%) were within 2 seconds, and 147 (87%) were within 5 seconds. The mean difference between the found marks and the ground truth was 1.79 seconds (SD = 3.32 seconds).

In both experiments, the total duration of each recording was over an hour, and the algorithm was able to keep up with the long live input, and automatically correct itself after moments of difference between the reference and the live input. If there is a “mistake" between productions and the AD becomes misaligned, the algorithm may correct itself as the production progresses. For example, the longest difference between reference and live for all experiments was about 21 seconds, which occurred during a significant period of divergence between the two recordings of *Legally Blonde*. However, the algorithm was back to correctly aligning once again less than 2 minutes later, with the next marks falling 120 ms from the ground truth.

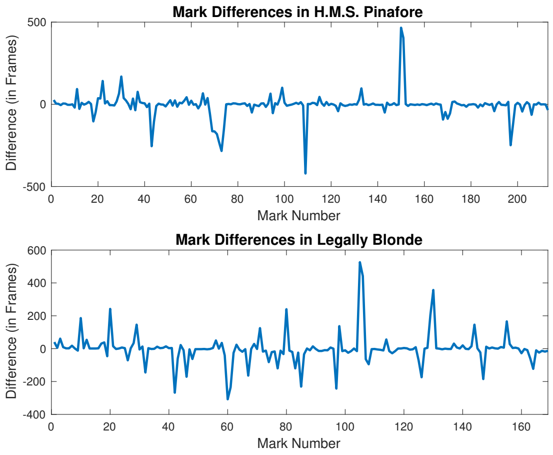

Figure 1: Accuracy of all marks shown for *H.M.S. Pinafore* and *Legally Blonde*, shown as deviations from ground truth in frames. X axis indicates mark number; Y axis indicates difference in number of frames (25 frames = 1 second).

Within the context of a live theatrical environment, these results show that most (79-86%) AD will be triggered within two seconds of the actual event occurring. These metrics indicate that theatergoers would be able to follow the visual elements of the production in a timely way.

Evaluating Pause Detection

To evaluate the process of pausing the algorithm when an interruption is detected, two 3-minute audio clips were obtained to represent two theatrical environments that could indicate a cue to pause. The first recording was of a small, empty theater. The second recording was of a large library atrium to represent a Broadway theater, including quiet audience chatter and activity, such as what would occur during an intermission. The experiment was performed twice, once for each audio clip. The first 30 seconds of each recording was used as training data, and the remaining 2.5 minutes was inserted after the overture of *H.M.S. Pinafore* (approximately 5 minutes into the recording) in the live input only. This emulates an interruption in the live performance that is not present in the reference recording, and thus makes the reference and live input substantially mis-aligned.

The algorithm used a threshold of 10 frames (0.4 seconds of consecutive input) to determine if there will be a pause. The number of frames *c* of which to check the cumulative path cost at each frame was set to 50. In this implementation, a random 50-frame subsection of the interruption training data was used at each frame, so as not to bias a particular part of the 30-seconds of training data. In these experiments, when

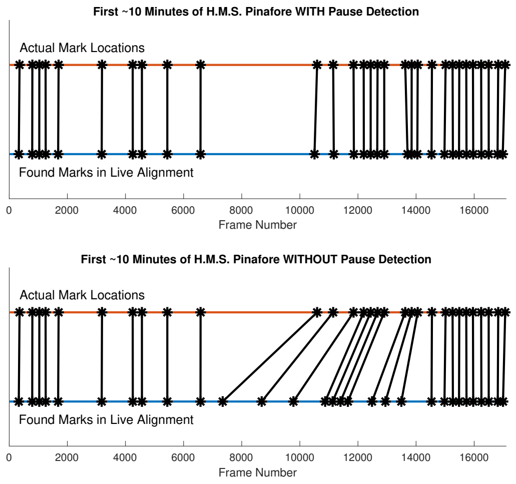

Figure 2: Accuracy of alignment in *H.M.S. Pinafore*'s first 10 minutes with pause detection (top) and without pause detection (bottom).

determining the lowest cost, a weight of 2 was applied to the calculation of the pause training data only. Instead of using $if(PausePath < LiveInputPath)$ to determine if a pause is likely, $if(PausePath * 2 < LiveInputPath)$ was used. This way, the algorithm favors the alignment continuing unless there is strong indication that an interruption is occurring. In a real-world situation, the training data would be obtained at the venue prior to the performance. The weight applied, if any, would be determined by how sensitive to interruptions the show technicians would like the program to be.

First, the results of a live alignment without the interruption checking is obtained, using the modified audio file as the live input. Then the same alignment was run again, this time with the interruption checking enabled. The results show that the mark accuracy after the interruption greatly increased when the algorithm was paused. Although both tests show that the system is able to correct itself and re-align eventually, adding a check for interruptions allows the audio description to remain consistently aligned, with no period of misfires beyond what was noted in the previous evaluation. Conversely, without adding this check, the system required about 2.3 minutes before it was correctly firing marks once again, which is about the same length as the interruption itself. For the first 50 marks (approximately 20 minutes), the average accuracy with interruption-detection enabled was 1.32 seconds, whereas without the detection the accuracy was 16.32 seconds.

5 SOFTWARE IMPLEMENTATION

Software implementations to increase the availability of audio description generally take the form of automating some task of AD creation or deployment, thus decreasing cost and complexity, and ultimately increasing availability. For example, the *CineAD* system [3] uses information from existing closed captions and a teleplay or screenplay of a fixed film or television program to generate a descriptive script automatically. Alternately, *LiveDescribe* [2] assists amateur describers with creating video AD in part by analyzing a video and allowing describers to record descriptive audio only during breaks in dialogue.

For a real-world setting, we propose both a software implementation and workflow that takes into account existing theatrical technology (audio recording) as well as established music information retrieval techniques (feature extraction and online time warping) to align AD in real time. The previous sections of this paper have discussed the algorithm and parameters of the software; this section describes how the software may be ideally used in a live setting.

Computer and Theatrical Requirements. The software in this implementation runs comfortably on a 2012-era MacBook Pro with a solid-state hard drive and 16 gigabytes of memory. The computer must be able to obtain two channels of mono input simultaneously—one input to capture the reference recording and (later) the live input, and the other to capture the live audio describer. Importantly, both channels must be isolated from each other such that they do not capture audio from each other, meaning the describer must be listening with headphones so that the reference recording is not captured on the descriptive audio track. The computer must also have a single mono audio output channel to transmit the AD to audience members during the live performance.

Software Interface. The software's core functions are *Record* and *Begin Show* features. *Record* activates the two mono inputs simultaneously to capture both the reference recording and the descriptive audio. The software records the reference's alignment to the AD by detecting when the speaker begins to talk, and the corresponding frame number is retained for the particular mark. Alternately, a third function, *Add Mark*, can be manually activated each time the describer wishes to add a new mark. The software may have supplemental editing features such as the ability to replace portions of the reference recording.

Capturing the Reference, Audio Description, and Interruption Training Data. With the software running, the setup configured as above, and the show commencing, the live describer activates *Record* on the interface and the system begins to capture both the performance and describer's voice on separate tracks. The system records the entirety of both mono inputs, while also capturing the sample numbers for each audio described event. By the end, the system will have all relevant data needed to play back the description to future audiences, and a human describer is no longer needed. If interruption detection is to be used, the training data may be obtained at any time prior to the live audio alignment.

Providing AD to Audiences. A theatrical technician activates the *Begin Show* button as the production commences. At this point, the mono output (to the wireless receivers of audience members) and the mono input for the live recording (which is the same input as for the reference) is activated. The online time warping process is activated, and the AD is triggered at the correct moment.

6 FUTURE WORK

In this paper we presented an automatic approach to triggering pre-recorded audio description during a live musical theater performance using an online time warping algorithm. The method is able to correct itself and adapt to variations between the reference and live performance, which is necessary for an effective real-time method. We also presented a method for automatically pausing the algorithm during moments of unanticipated interruptions. Although more work is needed to refine this process, it shows promise. The evaluation experiments showed that significant differences such as changes in casting, script, and instrumentation, are already handled robustly.

The software implementation of the system described here can be integrated into a theater's existing setup with minimal interference. For example, the system is able to capture, process, and deploy AD independent from the existing software or hardware controls, since it only uses an audio signal which is readily available in the performance space. No other setup or configurations are required. The simplicity of the method's technical setup and its overall flexibility provide a new way to make theater experiences for visually impaired audience members more inclusive and accessible.

Pause detection is important for theatrical works, especially given the need to respond to unanticipated audience interruptions. Longer online alignments, particularly those with multiple parts (such as breaks of indeterminate length between set pieces) need a system to respond to those events. In a truly automated, hands-off system for audio description deployment, the system needs to be robust enough to handle a wide range of interruptions that are common in theater and other long-duration live works. Further work is needed, for example, with testing additional features and different methods for detecting cues for pauses, including exploring techniques that don't require training data.

Given the proliferation of accessibility on personal computing devices, using a smartphone to align and deliver the description would improve the overall success of the system. Being able to track a live performance from a mobile device without having to be connected to a wireless transmission mechanism would allow AD to be completely in control of the user, not reliant on the setup of the theater.

Audio description is a relatively new but quickly expanding accommodation for those with visual impairments. While it is becoming more common in film and television, AD for live theatrical performances remains rare. Decreasing the cost and complexity of creating and deploying AD would increase its availability, thus making the enjoyment of live theater more accessible to blind and visually impaired individuals.

7 ACKNOWLEDGEMENTS

The authors would like to thank Juan Pablo Bello for his assistance, the Music and Audio Research Laboratory (MARL) at NYU Steinhardt, and the anonymous reviewers who provided invaluable insights.

REFERENCES

[1] Andreas Arzt. 2016. *Flexible and robust music tracking.* Ph.D. Dissertation. Johannes Kepler University, Linz.

[2] Carmen Branje and Deborah I. Fels. 2012. LiveDescribe: Can Amateur Describers Create High-Quality Audio Description? *Journal of Visual Impairment & Blindness* 106, 3 (2012). https://doi.org/10.1177/0145482X1210600304

[3] Virginia P. Campos, Tiago M. U. de Araújo, Guido L. de Souza Filho, and Luiz M. G. Gonçalves. 2018. CineAD: a system for automated audio description script generation for the visually impaired. *Universal Access in the Information Society* (2018). https://doi.org/10.1007/s10209-018-0634-4

[4] Simon Dixon. 2005. Live tracking of musical performances using online time warping. In *Proceedings of the 8th International Conference on Digital Audio Effects.* 92–97.

[5] S. Pavankumar Dubagunta. 2016. A simple MFCC extractor using C++ STL and C++11. https://github.com/dspavankumar/compute-mfcc

[6] Louise Fryer. 2016. *An Introduction to Audio Description: A Practical Guide.* Routledge. https://doi.org/10.4324/9781315707228

[7] Nat Lertwongkhanakool, Natthawut Kertkeidkachorn, Proadpran Punyabukkana, and Atiwong Suchato. 2015. An Automatic Real-time Synchronization of Live Speech with Its Transcription Approach. *Engineering Journal-Thailand* 19, 5 (2015), 81–99. https://doi.org/10.4186/ej.2015.19.5.81

[8] Elena Litsyn and Hagai Pipko. 2019. System and method for distribution and synchronized presentation of content. US Patent App. 16/092,775.

[9] Beth Logan. 2000. Mel Frequency Cepstral Coefficients for Music Modeling. In *ISMIR*, Vol. 270. 1–11.

[10] Lindasalwa Muda, Mumtaj Begam, and I. Elamvazuthi. 2010. Voice Recognition Algorithms using Mel Frequency Cepstral Coeffient (MFCC) and Dynamic Time Warping (DTW) Techniques. *Journal of Computing* 2, 3 (2010), 138–143.

[11] Malgorzata Plaza. 2017. Cost-effectiveness of audio description production process: comparative analysis of outsourcing and 'in-house' methods. *International Journal of Production Research* 55, 12 (2017), 3480–3496. https://doi.org/10.1080/00207543.2017.1282182

[12] Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Transactions on Acoustics Speech and Signal Processing* 26, 1 (1978), 43–49. https://doi.org/10.1109/TASSP.1978.1163055

[13] Joel Snyder. 2014. *The Visual Made Verbal: A Comprehensive Training Manual and Guide to the History and Applications of Audio Description.* American Council of the Blind, Inc.

[14] Agnieszka Szarkowska. 2011. Text-to-speech audio description: towards wider availability of AD. *Journal of Specialised Translation* 15 (2011), 142–162.