A Service-based System for Audio Context Recognition in Musical Performances

Carmine Colarusso
Dept. of Engineering
University of Sannio
Benevento, Italy
ccolarusso@unisannio.it

Gerardo Canfora
Dept. of Engineering
University of Sannio and CINI
Benevento, Italy
canfora@unisannio.it

Eugenio Zimeo
Dept. of Engineering
University of Sannio and CINI
Benevento, Italy
zimeo@unisannio.it

***Abstract*—This extended abstract presents an architecture and context recognition techniques for providing augmented reality (AR) content during the participation in a musical performance. The architecture is service-oriented and distributed across various environments and devices. Unlike traditional AR approaches based on video sources, here we propose audio recognition based on fingerprinting techniques integrated in dedicated services that aim at reducing the mismatch between the recognized audio fragments and pre-recorded performances. The system has been tested in a real theater context.**

***Index Terms*—Service architecture, Augmented Reality, Audio fingerprinting**

I. Introduction

Augmented reality (AR) combines real world with virtual computer-generated elements, enhancing users' perception of their environment through real-time access to contextualized content.

AR experiences can be delivered through different devices, including smartphones, tablets, smart glasses, and headsets. They exploit internal sensors and cameras to recognize the spatial context that is enhanced with additional virtual content. Typical approaches for spatial context recognition are marker-based (e.g, based on QR codes, specific images) or markerless (e.g, based on computer vision and other sensors to get and track environment features).

In this extended abstract, we introduce a completely different approach for augmented reality where context source is not visual but audio. In particular, we worked for enriching the experience of musical performance spectators with additional multimedia content related to listened audio. Examples are: opera libretto, musical score, subtitles, historical content related to the narrated context, etc. This content can be delivered both when the spectator is in the theater where the performance takes place and when he/she is remote.

To address this objective, on one side, we need to recognize audio patterns from live performances, on the other side, we need to match the recognized pattern with pre-recorded fingerprints extracted from a reference performance reproduction to understand where the recognized pattern is positioned within the overall musical performance.

We experimented with different audio recognition techniques and integrated the most performing one in a service-based architecture where a set of other services contributes to deliver the desired content by reducing possible errors in matching recognized audio with pre-recorded fingerprints. In the following sections, the service based architecture is presented and the main component, the *Fingerprinter*, is then discussed in details.

II. Architectural proposal

The service architecture (see Fig. 1) involves three main components:

- *AR Device*: a smartphone or AR-glasses on which the augmented contents will be displayed.
- *Context detector*: which in turn is composed of: *i)* a Sampler, to capture high-definition audio samples from the source; *ii)* a Fingerprinter, to pre-process audio samples to obtain a compact representation; *iii)* a Synchronizer, to correct the matching on the basis of possible time intervals in which the recognized audio could fall in;
- *Matching Service*: which exposes an interface to recognize audio samples. It could accept audio samples directly or pre-processed audio fingerprints. The Matcher compares the received fingerprint with the ones in a knowledge base (Fingerprint DB) and returns information about the opera and the offset (discrete time) correspondent to the submitted fragment.

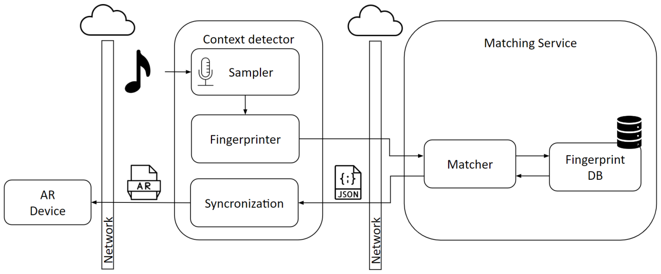

Fig. 1. AR Delivery Architecture

This work is a part of the CANTICO project funded by MUR

III. Fingerprinter

The main component of the architecture for context recognition is the Fingerprinter, i.e. the component that manages fingerprints. It could be placed either in the Context detector or in the Matching service, depending on the trade-off between computational resources and bandwidth. A fingerprint is a binary representation of audio samples optimized to be matched with an audio knowledge base.

A. *Fingerprints*

Multiple known techniques for fingerprint production have been experimented with.

1) Chromaprint: is an acoustic fingerprinting algorithm [1] that analyzes the spectral content of an audio signal by using Fast Fourier Transform (FFT) and by binning frequencies in the 12 pitch classes of musical notes (chromatic scale). Then, a binary string is obtained by applying a binary filter and quantization. When a query audio sample is provided, its fingerprint is generated using the same process, and then compared with the fingerprints stored in a database by using the Hamming Distance to find potential matches. This technique is not precise and requires working on whole traces but it represents the basis for the other techniques.

2) Dejavu: is an open-source audio fingerprinting and recognition library [2]. This technique starts with the search of local maximums on FFT extracted spectrograms that, together with the constellation of their neighbors, represent the fingerprints. Fingerprints are hashed into a compact representation and stored within a database. When a query audio sample is provided, it goes through the same preprocessing and query fingerprints are compared against the ones in the DB.

3) Computer Vision Based: this technique [3] involves applying computer vision to recognize musical tracks. The idea is to save an audio FFT spectrogram within a knowledge base, and then searching by measuring the Euclidean Distance between a feature extracted from a stored image and one extracted from a query audio sample spectrogram. The feature can be extracted by using SIFT or ORB techniques.

4) Cross Similarity Matrix: [4] proposes to extract a fingerprint that uniquely identifies an audio sample, even in the presence of alterations (e.g., key changes, instrument changes, vocals, etc.). The method begins by extracting the spectrogram of an audio source using the Constant Q-Transform (CQT). The benefits of this transformation are: *i)* a better approximation of the behavior of the human ear than FFT; *ii)* timbre invariance, providing a robust representation of the spectrum even if the same musical performance it is performed with different musical instruments; and *iii)* transformation of a musical key change, that is a logarithmic frequency shift, into a linear one. The spectrogram is then filtered using an adaptive threshold filter to binarize the obtained values. Hence, by sliding a fixed-length time window, with a certain overlap factor, over the binarized spectrogram, the 2DFT (2-Dimensional Fourier Transform) is computed for each window. The concatenation of the windows into a single matrix represents the fingerprint. Matching is performed by calculating the Euclidean distance between all possible pairs of sub-fingerprints, returning the closest one.

IV. Experimentation

All the analyzed techniques were tested to evaluate prediction accuracy: (a) to recognize the same fragment used during training (reference) and (b) to recognize live performances or covers. All techniques behave well in recognizing reference audio, as shown in Tab. I, while most of them fail with live performances.

TABLE I
Fingerprint technique accuracy comparison

	Chromaprint	Dejavu	Computer Vision Based	Cross Similarity Matrix
reference	-	100 %	93 %	100 %
live	-	18 %	66 %	**87 %**

The most performing solution based on III-A4 was chosen to implement the *Fingerprinter* component. We used a modified version of the methodology for the case study, which differs slightly from the original method because it is designed to recognize the whole representation after the performance, whereas our goal is to temporally locate an acquired sample inside a reference track. For this purpose, for the sample fingerprint production, we avoided the phase of sliding-window movement and concatenation, obtaining a fingerprint that consists of a single 2DFFT. To achieve this, the acquired sample must have the same length as the window used in the previous step.

For the original musical performance-matching phase, the obtained fingerprint is compared with the sequence of 2DFFTs that constitutes the fingerprint of the complete musical performance, and then the Matcher searches for the closest one by using the Euclidean distance.

A. *Live experimentation*

The proposed solution has been tested during a live performance ("Ascesa e caduta della città di Mahagonny" by Kurt Weil) at Teatro Regio of Parma. Specifically, the audio was recorded during the premiere and used as a reference, while the augmented architecture was employed during the second performance.

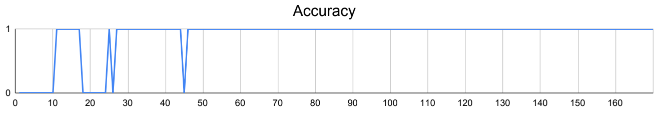

Fig. 2. Experimentation Accuracy during live performance (87%)

Fig. 2 shows, for each second, whether the predicted value matches the expected value (1) or not (0) during a performance fragment of 170 seconds length. The resulting prediction accuracy value is 87% affected by errors in the initial part of the test due to noises recorded during the performance.

REFERENCES

[1] D. Jang, C. D. Yoo, S. Lee, S. Kim, and T. Kalker, "Pairwise boosted audio fingerprint," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 995–1004, 2009.

[2] W. Drevo, "Audio fingerprinting with python and numpy, https://willdrevo.com/fingerprinting-and-audio-recognition-with-python/," 2013.

[3] M. Zanoni, S. Lusardi, P. Bestagini, A. Canclini, A. Sarti, and S. Tubaro, "Efficient music identification approach based on local spectrogram image descriptors," in *Audio Engineering Society Convention 142*, Audio Engineering Society, 2017.

[4] P. Seetharaman and Z. Rafii, "Cover song identification with 2d fourier transform sequences," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 616–620, IEEE, 2017.