DISCOVERY OF REPEATED VOCAL PATTERNS IN POLYPHONIC AUDIO: A CASE STUDY ON FLAMENCO MUSIC

Nadine Kroher[1], *Aggelos Pikrakis*[2], *Jesús Moreno*[3], *José-Miguel Díaz-Báñez* [3]

[1] Music Technology Group
Univ. Pompeu Fabra, Spain

[2] Dept. of Informatics
Univ. of Piraeus, Greece

[3] Dept. of Applied Mathematics II
Univ. of Sevilla, Spain

ABSTRACT

This paper presents a method for the discovery of repeated vocal patterns directly from music recordings. At a first stage, a voice detection algorithm provides a rough segmentation of the recording to vocal parts, based on which an estimate of the average pattern duration is computed. Then, a pattern detector which employs a sequence alignment algorithm is used to yield a ranking of pairs of matches of the detected voiced segments. At a last stage, a clustering algorithm produces the final repeated patterns. Our method was evaluated in the context of flamenco music for which symbolic metadata are very hard to produce, yielding very promising results.

***Index Terms*—** Pattern discovery, flamenco music.

1. INTRODUCTION

The development of algorithms for the automated discovery of repeated melodic patterns in musical entities is an important problem in the field of Music Information Retrieval (MIR) because the extracted patterns can serve as the basis for a large number of applications, including music thumbnailing, database indexing, similarity computation and structural analysis, to name but a few.

Recently, a related task, titled “Discovery of Repeated Themes and Sections” was carried out in the context of the MIREX evaluation framework [1] and provided a state-of-the-art performance evaluation of the submitted algorithms. Most solutions to this task have so far used a symbolic representation of the melody extracted from a score as a basis for analysis [2]. However, when applying a state-of-the-art symbolic approach to automatic transcriptions of polyphonic pieces, [3] report a significant performance decrease.

In our study, we have chosen to focus on the automatic discovery of repeated melodic patterns in flamenco singing. This task poses several challenges given the unique features of this music genre [4]. In contrast to other music genres, flamenco is an oral tradition and available scores are scant, almost limited to manual guitar transcriptions. Recently, an algorithm to automatically transcribe flamenco melodies was developed [5] and used in the context of melodic similarity [6] and supervised pattern recognition [7]. However, the reported accuracy of symbolic representations when compared to manually annotated ground truth are still very low (note accuracy below 40%). Furthermore, most symbolic-based approaches rely on transcriptions quantised to a beat grid. However, in flamenco, irregular accentuation and tempo fluctuations increase the difficulty of rhythmic quantisation. Therefore, the system described in [5] outputs a note representation which is not quantised in time.

We propose an efficient algorithm for unsupervised pattern discovery, which operates directly on short-term features extracted from the audio recording, without computing a symbolic interpretation at an intermediate stage. This type of analysis can be also encountered in the context of structural segmentation [8], [9], [10], [11], where, in contrast to our targeted short motifs, an audio file is segmented into long repeating sections that capture the form of a music piece. In [12], a structural analysis technique is adopted to extract shorter repeated patterns from monophonic and polyphonic audio and the work in [13] uses dynamic time warping for inter- and intra-recording discovery of melodic patterns based on pitch contours.

Our method is applied on the analysis of the flamenco style of *fandangos de Huelva*, in which pattern repetition is a frequent phenomenon, mainly due to the folk nature of this style and its popularity in festivals. The discovered repeated patterns can be readily used for the establishment of “characteristic signatures” in groups of flamenco songs. In addition, they can play an important role in inter-style studies for the discovery of similarities among different musical entities and in ethnomusicological studies which aim at tracking the evolution of the cultural aspects of flamenco styles over the years [14]. The research contribution of our approach lies in the development of an efficient algorithm for the discovery of vocal patterns directly from the music recording (circumventing the need for symbolic metadata) and its application in the field of flamenco music.

The paper is structured as follows: the next section presents the singing voice detection algorithm, Section 3 describes the pattern duration estimator, Section 4 presents the pattern discovery algorithm which operates on the extracted voiced segments and Section 5 describes the evaluation approach. Finally, conclusions are drawn in Section 6.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

2. VOCAL DETECTION

As we are targeting repeated patterns in vocal melodies, we first detect sections in which the singing voice is present based on low-level descriptors which exploit the limited instrumentation of the music under study (mainly vocals and guitar). Note that related methods that detect vocal segments [15], [16] have so far mainly focused on commercial Western type music (where instrumentation varies a lot) and use machine learning algorithms to discriminate between voiced and unvoiced frames. Of course, such approaches may be alternatively used when the instrumentation becomes more complex.

The proposed vocal detector is based on the fact that when analysing the spectral domain, we observe an increased spectral presence in the range 500Hz-6kHz due to the singing voice (compared to pure instrumental sections). We therefore extract the spectral band ratio, $b(t)$, of the normalised spectral magnitude, $|X(f,t)|$, using a moving window size of 4096 samples and a hop size of 128 samples (assuming a sampling rate of 44100Hz), as follows:

$$b(t) = 20 \cdot log10 \left(\frac{\sum_{f \geq 500}^{f \leq 6000} |X(f,t)|}{\sum_{f \geq 80}^{f \leq 400} |X(f,t)|} \right) \quad (1)$$

where $X(f,t)$ is the Short-time Fourier Transform of the signal. As we are mainly dealing with live stereo recordings, where the voice is usually more dominant on one channel due to the singer's physical location on stage, we extract $b(t)$ for both channels and select the channel with the higher average value. Furthermore, we extract the frame-wise root mean square (RMS) of the signal, $rms(t)$, over the same windows and estimate its upper envelope, $rms_{Env}(t)$, by setting each RMS value to the closest local maxima, thus resulting into a piece-wise constant function.

We now detect singing voice sections by combining the information that is carried out by the previously extracted spectral band ratio and the RMS envelop. Specifically, $b(t)$ is first shifted to a positive value by adding the minimum value of the sequence and it is then weighted by the respective RMS value. The resulting sequence, $v(t)$, is normalised to zero mean. We then assume that positive values of $v(t)$ correspond to voiced frames and negative values to unvoiced ones. In other words, our voicing function, $voicing(t)$, is the sign function, i.e., $voicing(t) = sgn(v(t))$. Obviously, $voicing(t)$ outputs binary values, which are then smoothed with a moving average filter (30ms long). The resulting sequence, $c(t)$, takes values in $[0,1]$ and can be interpreted as a confidence function for the segmentation algorithm in Section 4. An overview of the process is given in Figure 1.

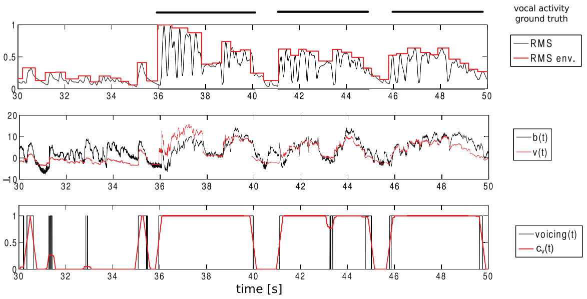


Fig. 1. Vocal detection overview.

3. PATTERN DURATION ESTIMATION

The detected vocal segments are used to estimate a mean pattern duration for each music recording which will be fed to the pattern detector (Section 4). Due to the rhythmic complexity of this type of music and the non-trivial relation between accentuation in accompaniment and vocal melody, common beat estimation methods are not suitable for providing estimates of pattern durations. Therefore, we proceed to defining a vocal onset detection function, $p(t)$, assuming that strong vocal onsets coincide with large (positive) changes in the vocal part of the spectrum and also with a volume increase. To this end, for each frame, the spectral band ratio difference value, $\Delta b(t)$, is computed, by summing $b(t)$ over all frames within a segment of length l_w=435 ms, before ($b_{prev}(t)$) and after ($b_{post}(t)$) each time instance, t:

$$\Delta b(t) = (b_{post}(t) - b_{prev}(t)) \cdot b_{post}(t)) \quad (2)$$

In a similar manner, the RMS envelope difference function, $\Delta rms_{Env}(t)$, is computed using the previous mid-term windows. The combined onset detection function, $p(t)$, is then defined as $p(t) = \frac{\Delta b(t)}{\overline{\Delta b}} \cdot \frac{\Delta rms_{Env}(t)}{\overline{\Delta rms_{Env}}} \cdot voicing(t)$

We then define that vocal onsets coincide with those local maxima of $p(t)$ that exceed twice the average of $p(t)$ over all frames. Subsequently, we estimate a set of possible pattern durations by analysing the distances between estimated vocal onsets (starting points) in a histogram with a bin width of 0.1 seconds. We assume that the peak bin of the histogram corresponds to a short rhythmical unit and we take its smallest multiple larger than 3 seconds as the average pattern duration, dur_{MIN}.

4. PATTERN DETECTION

After the voicing confidence function, $c(t)$, and the estimated pattern duration have been computed, we proceed to detecting pairs of similar patterns and then use a clustering scheme to create clusters of repeated patterns. We first apply a simple segmentation scheme on sequence $c(t)$. Namely, any subsequence of $c(t)$ that lies between two subsequences of zeros is treated as an audio segment containing singing voice, provided that its duration is at least half the estimated pattern duration length.

At the next step, we extract the chroma sequence of the audio recording [17] using a short-term processing technique (window length and hop size are 0.1 s and 0.02 s, respectively), normalize each dimension of the chroma vector to zero mean and unit standard deviation and preserve the chroma subsequences that correspond to the previously detected voiced segments. Due to the microtonal nature of the music under study, a 24-bin chroma vector representation has been used. We adopted a chroma-based representation because pitch-tracking methods on this type of music corpora have so far exhibited error prone performance and in addition, the chroma vector has shown to provide good results on music thumbnailing applications [17]. Note that our method does not exclude the use of other features or feature combinations. The output of the feature extraction stage, is a set of M sequences, $X_i, i = 1, \dots, M$, of 24-dimensional chroma vectors (of possibly varying length).

4.1. Pairwise matching

We then examine pairwise the extracted chroma sequences using a sequence alignment algorithm. The main characteristics of this algorithm are that **(a)** it operates on a similarity grid, **(b)** it uses the cosine of the angle of two chroma vectors as a local similarity measure, and **(c)** it uses a gap penalty for horizontal and vertical transitions among nodes of the grid. The result of the sequence alignment procedure can be a matching of subsequences, which is a desired property in our case, because there is no guarantee that the extracted voice segments are accurate with respect to duration and time offset.

To proceed with the description of the sequence alignment algorithm and for the sake of simplicity of notation, let $X = \{x_i, i = 1, \dots I\}$ and $Y = \{y_j, j = 1, \dots J\}$ be two chroma sequences that are being aligned. We assume that X is placed on the horizontal axis of the matching grid. Also, let $s(j, i)$, be the local similarity of two vectors y_j and x_i, defined as the cosine of their angle, $s(j,i) = \frac{\sum_{k=1}^{L} y_j(k) x_i(k)}{\sqrt{\sum_{k=1}^{L} y_j^2(k)}\sqrt{\sum_{k=1}^{L} x_i^2(k)}}$, where $L = 24$.

We then construct a JxI similarity grid and compute the accumulated similarity at each node. To achieve this, dynamic programming is used. Specifically, the accumulated similarity, $H(j, i)$, at node (j, i) of the grid, is defined as

$$H(j,i) = \max \begin{cases} H(j-1, i-1) + s(i,j) - G_p, \\ H(j, i-k) - (1 + kG_p), \;\; k = 1, \dots, G_l, \\ H(j-m, i) - (1 + mG_p), m = 1, \dots, G_l, \\ 0 \end{cases} \tag{3}$$

where $j \geq 2$, $i \geq 2$, G_p is the gap penalty and G_l is the maximum allowed gap length (measured in number of chroma vectors). In Section 5, we provide recommended values for G_p and G_l for the corpus under study. Note that a diagonal transition contributes the quantity $s(i,j) - G_p$, which can be positive or negative, depending on how similar y_j and x_i are. Furthermore, each deletion (vertical or horizontal) introduces a gap penalty equal to $(1 + k \times G_p)$, where k is the length of the deletion (measured in number of frames).

For each node of the grid, we store the winning predecessor, $W(j, i)$. If $H(j, i)$ is zero for some node, $W(j, i)$ is set equal to the fictitious node $(0, 0)$. Upon initialization $H(j, 1) = max\{S(j, 1) - G_p, 0\}, j = 1, \dots, J$, and $H(1, i) = max\{S(1, i) - G_p, 0\}, i = 1, \dots, I$. In addition, $W(j, 1) = (0, 0), j = 1, \dots, J$, and $W(1, i) = (0, 0), i = 1, \dots, I$.

After the whole grid has been processed, we locate the node that has accumulated the highest (matching) score, and perform backtracking until a $(0, 0)$ node is reached. The resulting best path reveals the two subsequences that yield the strongest alignment. The matching score is then normalized by the number of nodes in the best path sequence. In this way, the matching score is not biased against shorter paths.

If the lengths of both subsequences corresponding to the best path do not exceed half the estimated pattern length, we select the node with the second largest accumulated score and perform again the backtracking procedure. This is repeated until we detect the first pair of subsequences that exhibit sufficient length. If no such subsequences exist, the original chroma sequences, X and Y are considered to be irrelevant.

After the pairwise similarity has been computed for all voiced segments, we select the K higher values, where K is a user defined parameter (in our study the best results were obtained for $K = 15$). The respective best paths reveal the endpoints (frame indices) of the subsequences that were aligned. We therefore end up with a set, P, of K pairs of patterns,

$$P = \{\{(t_{11}, t_{12}), (t_{13}, t_{14})\}, \dots, \{(t_{K1}, t_{K2}), (t_{K3}, t_{K4})\}\}$$

where $\{(t_{i1}, t_{i2}), (t_{i3}, t_{i4})\}$ denotes that the pattern (chroma sequence) starting at frame index t_{i1} and ending at frame index t_{i2} has been aligned with the pattern starting at t_{i3} and ending at t_{i4}.

4.2. Pattern clustering

The goal of this last stage is to exploit the relationship of the extracted pattern pairs by means of a simple clustering algorithm. We propose a simple, frame-centric clustering scheme. This is not an optimal scheme in any sense but it is of low computational complexity and yields acceptable performance. The investigation of more complicated approaches is left as a topic of future research.

The proposed scheme is based on the observation that a frame (chroma vector) can be part of one or more pattern pairs. To this end, we assume that the i-th pattern pair is represented by its index, i. Therefore, a set of such indices (frame label) can be directly associated with each feature vector. For example, if the m-th chroma vector is encountered in the 3rd and 4th pattern pairs, the respective frame label will be the set $\{3, 4\}$. In general, by simple observation of the extracted pattern pairs, the m-th frame will be assigned the label $c_m = \{l_1, l_2, \ldots, l_m\}$. If a frame has not been part of any pair, the respective label is set equal to the empty set.

In this way, we generate a sequence, C, of frame labels, i.e, $C = \{c_1, c_2, \ldots, c_N\}$, where N is the length of the music recording (measured in frames). We then define that a subsequence of C starting at frame i and ending at frame j forms a *maximal segment* if

$$c_k \cap c_{k+1} \neq \emptyset, \forall i \leq k \leq j-1 \tag{4}$$

$$c_{i-1} \cap c_i = \emptyset \;\; or \;\; c_{i-1} = \emptyset \tag{5}$$

$$c_j \cap c_{j+1} = \emptyset \;\; or \;\; c_{j+1} = \emptyset \tag{6}$$

All maximal segments can be easily detected by scanning sequence C from left to right: condition (5) is used to detect candidate starting points, condition (4) is used for expanding segments and condition (6) serves to terminate the expansion of a segment to the right. Each time a maximal segment is completed, its label is set equal to the union of the labels of all its frames. After all maximal segments have been formed, we assign to the same cluster all segments with the same label. In this way, we expect that each cluster will contain segments which represent repetitions of a prototypical pattern. Figure 4.2 presents the output of our method for a music recording, including ground truth and the estimated starting points (with stars). Circles mark errors. Repeated patterns 3 and 4 failed to be discovered and pattern 2 was mistakenly clustered with pattern 1. The latter is due to the fact the pattern 2 is very similar to pattern 1 even when perceived by a human listener.

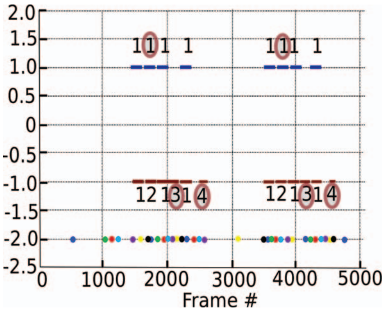


Fig. 2. Discovered patterns and ground truth (bottom).

5. EVALUATION

We evaluate our system on a corpus consisting of 11 recordings of ”fandangos”, a flamenco singing style. Flamenco experts manually annotated the repeated patterns in each track. The number of patterns per track varies between 3 to 7, with 7 out of 11 tracks exhibiting 4 different repeated patterns. The number of instances per pattern varies from 2 to 9, with most patterns exhibiting 2 or 3 instances. Pattern durations lie in the range $[1.5, 5.8]$ s, with the majority of patterns being at least 3 s long.

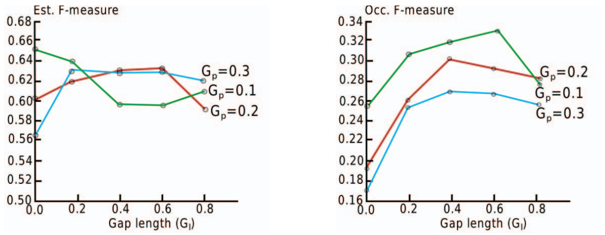
In order to evaluate the performance of the proposed method, we follow the approach adopted by the previously mentioned MIREX task and compare to the audio-based approach in [12] (referred to as NF-14 in the presented results). It should be mentioned that this baseline method is not targeting the singing voice in particular and assumes a constant tempo, which is not necessarily a valid assumption for the genre under study.

It has to be noted that, although the MIREX task evolves around MIDI annotations and synthetic audio, it defines two categories of performance measures that can be readily applied in our study. The first category includes establishment precision, Pr_{Est}, establishment recall, R_{Est} and establishment F-measure, F_{Est}. The second category includes occurrence precision, Pr_{Occ}, occurrence recall, R_{Occ} and occurrence F-measure, F_{Occ}. The term establishment means that a repeated pattern has been detected by the algorithm, even in the case when not all instances of the pattern have been discovered. On the other hand, the occurrence performance measures quantify the ability of the algorithm to retrieve all occurrences of the repeated patterns. For details on the computation of these performance measures, the reader is referred to [3] and the aforementioned MIREX competition task [1].

As it was described in Section 4, our method uses three parameters during the pattern detection stage, namely the gap penalty, G_p, the gap length, G_l, and the number, K, of highly ranked pair matches. Figure 3 presents the establishment and occurrence F-measures for different value combinations of G_p and G_l, assuming $K = 15$. It can be seen that a good trade-off between F_{Est} and F_{Occ} can be achieved when $G_p = 0.1$ and $G_l = 0.6$. For this combination of values, $F_{Est} \approx 0.60$, and $F_{Occ} \approx 0.33$. Table 1 presents how parameter K affects the performance measures and gives a comparison to the baseline method. It can be observed that $K = 15$ is indeed a reasonable choice for this parameter. Furthermore, it can be seen that for both establishment and occurrence, the baseline method exhibits a slightly higher precision, but since its recall is low, the resulting F-measures are inferior to our approach. For our method, establishment and occurrence recall are higher than their precision counterparts. This means that the method is capable of detecting the annotated repeated patterns to the expense of certain noise in the results.

	K=10	**K=15**	K=20	NF-14
Pr_{Est}	0.43	0.48	0.50	0.63
R_{Est}	0.71	0.80	0.78	0.37
F_{Est}	0.54	0.60	0.61	0.47
Pr_{Occ}	0.23	0.23	0.22	0.30
R_{Occ}	0.32	0.56	0.50	0.07
F_{Occ}	0.27	0.33	0.31	0.11

Table 1. Performance measures for different values of K ($G_p = 0.1$ and $G_l = 0.6$). NF-14 is the baseline method.



(a) Establishment F-measure

(b) Occurrence F-measure

Fig. 3. Performance curves for different values of G_p over G_l (in seconds), when $K = 15$.

6. CONCLUSIONS

This paper presented a computationally efficient method for the discovery of repeated vocal patterns directly from the music recording. Our study focused on flamenco music genre of "Fandangos", for which state-of-the-art pitch extraction algorithm provide noisy results, making the music transcription task (MIDI transcription) a hard one. The proposed method can be seen as a voice detection module followed by a pattern detector, in the heart of which lies a sequence alignment algorithm. Our evaluation study has indicated that the proposed approach performs satisfactorily and the reported evaluation results are in line with the performance of algorithms working on symbolic data for the MIREX task of repeated pattern finding. By adapting the vocal detection to a given instrumentation, the approach can be adapted to other singing traditions with similar characteristics.

Acknowledgments. This research was partly funded by the Junta de Andalucía, project COFLA: Computational Analysis of flamenco Music, FEDER-P12-TIC-1362 and the Spanish Ministry of Education, project SIGMUS, TIN2012-36650.

REFERENCES

[1] J. S. Downie, "The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research," *Acoustical Science and Technology*, vol. 29, no. 4, pp. 247–255, 2008.

[2] B. Jansen, W. B. de Haas, A. Volk, and P. van Kranenburg, "Discovering repeated patterns in music: state of knowledge, challenges perspectives," in *Proc. CMMR*, 2013.

[3] T. Collins, S. Boeck, F. Krebs, and G. Widmer, "Bridging the audio-symbolic gap: The discovery of repeated note content directly from polyphonic music audio," in *53rd AES Conference: Semantic Audio*, Jan 2014.

[4] F. Gómez, JM Díaz-Bánez, E. Gómez, and J. Mora, "Flamenco music and its computational study," in *Bridges: Mathematical Connections in Art, Music, and Science*, 2014, pp. 119–126.

[5] E. Gómez and J. Bonada, "Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a cappella singing," *Computer Music Journal*, vol. 37, no. 2, pp. 73–90, 2013.

[6] J.M. Díaz-Báñez and J.C. Rizo, "An efficient dtw-based approach for melodic similarity in flamenco singing," in *Similarity Search and Applications*, pp. 289–300. Springer, 2014.

[7] A. Pikrakis et al., "Tracking melodic patterns in flamenco singing by analyzing polyphonic music recordings," in *Proc. ISMIR*, 2012, pp. 421–426.

[8] G. Peeters, "Sequence representation of music structure using higher-order similarity matrix and maximum-likelihood approach.," in *Proc. ISMIR*, 2007, pp. 35–40.

[9] M. Müller and F. Kurth, "Towards structural analysis of audio recordings in the presence of musical variations," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 163–163, 2007.

[10] R. B. Dannenberg and N. Hu, "Discovering musical structure in audio recordings," in *Music and Artificial Intelligence*, pp. 43–57. Springer, 2002.

[11] M. Levy and M. Sandler, "Structural segmentation of musical audio by constrained clustering," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 2, pp. 318–326, 2008.

[12] O. Nieto and M. M. Farbood, "Identifying polyphonic patterns from audio recordings using music segmentation techniques," in *Proc. ISMIR*, Taipei, Taiwan, 2014.

[13] G. Sankalp et al., "Mining melodic patterns in large audio collections of indian art music," in *International Conference on Signal Image Technology & Internet Based Systems - Multimedia Information Retrieval and Applications*, Marrakesh, Morocco, 2014.

[14] J.M. Díaz-Báñez J.M. Marqués, I., "El cante por alboreá en utrera: desde el rito nupcial al procesional.," in *Investigación y Flamenco, J.M. Díaz-Báñez y F. Escobar (eds.).*, pp. 193–204. Signatura Ediciones, 2010.

[15] V. Rao, C. Gupta, and P. Rao, "Context-aware features for singing voice detection in polyphonic music," in *Proc. of the Adaptive Multimedia Retrieval Conf.*, 2011.

[16] M. Ramona, G. Richard, and B. David, "Vocal detection in music with support vector machines," in *Proc. of the IEEE ICASSP*, 2008, pp. 1885–1888.

[17] M. A. Bartsch and G. H. Wakefield, "Audio thumbnailing of popular music using chroma-based representations," *Multimedia, IEEE Transactions on*, vol. 7, no. 1, pp. 96–104, 2005.