AUTOMATIC TRANSCRIPTION OF FLAMENCO GUITAR FALSETAS

Sonia Rodríguez, Emilia Gómez, Helena Cuesta
Music Technology Group, Universitat Pompeu Fabra
{sonia.rodriguez,emilia.gomez,helena.cuesta}@upf.edu

ABSTRACT

This work deals with the automatic transcription and characterization of flamenco guitar, with a focus on short melodic interludes improvised between sung verses. These are called *falsetas* in the flamenco argot and are very challenging for manual and automatic transcription due to their fast and highly ornamented nature. However, they are a key resource for guitar players to practice. We adapted a state of the art singing transcription algorithm to process an audio signal containing one or several guitar *falsetas* and extract their symbolic representation. The algorithms first perform a segmentation to locate the guitar fragments and then a symbolic transcription of these segments into symbolic representation. In order to evaluate it, we collected the first (to our knowledge) annotated *falseta* datasets. Our results confirm the difficulty of the task, and a detailed study of two transcriptions revealed that combining the algorithm with specific musical knowledge about the scale used by the song, improves the performance of the system. Our approach follows the principles of research reproducibility, and the system is integrated in a computer-assisted paradigm, where the user complements the automatic annotation with a priory knowledge to generate a final transcription.

1. INTRODUCTION

Flamenco is a musical genre that includes three basic elements: *cante* (singing), *toque* (guitar), *baile* (dancing), and has its own rules and traditions. This sociocultural movement has extended internationally beyond its geographical origin, becoming an Intangible Cultural Heritage of Humanity by UNESCO [1] in 2010.

Unlike other musical genres, flamenco guitar performance is orally transmitted; both songs and terminology have passed down across generations without a standard writing system. Flamenco *falsetas* are defined as short improvised melodies played between sung verses.

1.1 Related work

The COFLA [2] project deals with how computational models can support the analysis and synthesis of flamenco music to provide an adaptation of the general Music Information Retrieval (MIR) methodologies. Some of these aspects are linked to standard MIR tasks such as melodic similarity (Kroher et al., 2014) or genre classification (Salamon, Rocha & Gómez, Salamon et al.) which have been evaluated and adapted to flamenco music. Previous research has addressed the automatic transcription of flamenco singing, which has revealed to be challenging compared to other singing styles (Gómez & Bonada, 2013) (Kroher & Gómez, 2016). The present study focuses on the flamenco guitar.

[1] http://www.unesco.org/culture/ich/en/RL/flamenco-00363
[2] COFLA: COmputational analysis of FLAmenco music. www.cofla-project.com.

1.2 The flamenco guitar

When analyzing pieces of traditional flamenco music, we observe a dialog between instruments that appears throughout most of the songs. In particular, the most important dialogue is found between singing (*cante*) and guitar (*toque*), as they alternate on the roles of soloist and accompanying instrument. This is a key factor in our research because after a singing section, the lead is taken by the guitar player during the *falseta*. The detection and segmentation of the *falsetas* are the first steps of the proposed system, which are then followed by the transcription into a symbolic representation using the MIDI format as depicted in Figure 1. Regarding the transcription stage, we study relevant sound characteristics of flamenco guitar and typical playing techniques in order to build an optimal transcription method.

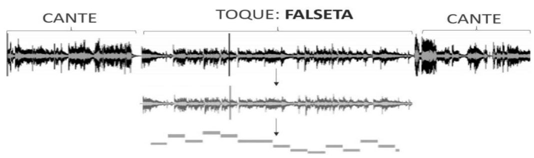

Figure 1: *Falseta transcription given the traditional dialogue between cante and toque*

1.3 Goals and contributions

We aim at providing a computer-aided system as a first step to the manual transcription, which requires advanced music and flamenco knowledge. This project is motivated by the lack of scores of flamenco guitar pieces and the high difficulty of creating them. These manual transcriptions provide complementary information to the note's representation such as fingering or dynamics. However, this information is very hard to find in an automatic way with existing techniques.

Our system provides a first step to reduce the cost of a complete transcription. The main purpose of this work is to develop an algorithm capable of automatically segmenting and transcribing flamenco guitar sections, also called *falsetas*, which would be a useful tool for learning and studying guitar. To evaluate the system, we created two manually annotated datasets: one for the segmentation and one for the MIDI transcription.

2. DATASET BUILDING

It is hard to find flamenco guitar datasets due to the absence of related research, mainly in the transcription field. Since we want to evaluate two different parts of our system, we built two datasets: one for the segmentation stage and another one for the transcription.

2.1 Segmentation dataset

Falsetas are musical segments delimited in time and in the segmentation stage we look for their boundaries, i.e. start and end times. We come upon a big controversy regarding the required length of a guitar section to be considered as a *falseta*, as well as identifying start and end points, because there is not a clear agreement on this topic. For this reason we define the minimum length as an input parameter of our system, which is set to 15 seconds by default.

The segmentation dataset contains twenty songs, including 43 falsetas (19.5 minutes of audio) from Camarón de la Isla and Paco de Lucia (1969-1977). For each of them we manually annotated the start and end times as ground truth for the segmentation step. All of them are recognized as an exemplary repertoire for classical flamenco dialog between *cante* and *toque*.

2.2 Transcription dataset

The dataset created for automatic transcription, called *ToqueFlamenco* contains the manual annotations of ten *falsetas* including onset, offset and pitch for each note. To create this data, we obtained [3] and edited the score of each piece and then converted them into MIDI files. Finally, we manually aligned the MIDI and the original audio to increase the accuracy. The dataset and details of the annotation procedure are provided in our web page [4] .

3. PROPOSED METHOD

In this section, we detail the steps of the algorithm: given an audio file, the system automatically finds and transcribes the guitar sections taking into account some classical flamenco features. Our algorithm is based on a state of the art method for flamenco singing transcription (Kroher & Gómez, 2016) which is published as a python library *PyCante* [5] . We adapt it to flamenco guitar in a similar python library *PyToque* [6] . The method consists of three main stages: *falsetas* segmentation, transcription and post-processing, each of them explained in the following subsections.

3.1 Falsetas segmentation

The aim of this part is to detect and segment *falsetas* from a song that also includes vocal parts (*cante*). *PyToque*, as a user's choice, allows to skip this step if the input contains only guitar sections.

[3] www.canteytoque.es, www.tabsflamenco.com
[4] https://doi.org/10.5281/zenodo.804050
[5] https://github.com/NadineKroher/PyCante
[6] https://github.com/SoniaLuque/PyToque

3.1.1 Channel selection

In flamenco stereo recordings, the vocals are usually more predominant in one of the channels: in order to create an artificial panorama that simulates a live performance, the guitar is strongly separated from the vocals. To make the *falsetas* segmentation easier, one of the channels is automatically selected to reduce noise and irrelevant information. To carry out this task, both channels are analyzed in terms of the distribution of their spectral energy. As showed in (Kroher & Gómez, 2016), the energy density increases between 500 Hz and 6 kHz when vocals are present. The channel selection strategy proposed in *PyToque* is based on picking out the one with the lowest average density in that range. Alongside the guitar section, the singer commonly says short sentences, also known as *jaleos*. Because of this, by choosing the channel where the vocals are not predominant, the delimitation is more precise since it avoids having a *falseta* divided by a *jaleo* by mistake.

To analyze the spectral content we compute the Short Time Fourier Transform (STFT) using a Hanning window of size N=2048 samples. According to the spectral vocal features that we mentioned above, we define the suitable frequency range both for vocals and guitar. Then, the spectral band ratio (SBR) is computed frame-wise dividing the sum of normalized magnitudes spectrum $|\dot{X}|$ of the vocal frequency band by the one corresponding to the guitar (see Eq.1), being k the bin corresponding to frequency f. After computing the average along the entire signal for each channel, the system selects the one with lowest vocal presence i.e. with lowest SBR average (unlike *PyCante*)

$$SBR[n] = 20 \cdot log10 \left(\frac{\sum\limits_{k(500Hz)<k<k(6KHz)} \left|\dot{X}[k,n]\right|}{\sum\limits_{k(80Hz)<k<k(400Hz)} \left|\dot{X}[k,n]\right|} \right) \tag{1}$$

3.1.2 Melody extraction

As we mentioned before, flamenco songs contain a dialogue between the voice (*cante*) and the guitar (*toque*). The proposed method exploits this feature to locate the *falsetas*. We use the melody extraction algorithm MELODIA (Salamon & Gomez, 2012) to extract the predominant pitch of the whole piece, which will correspond to the singing voice part, as the fundamental frequency range is adapted to singing, as well as rules for selecting pitch contours with fluctuations which are characteristic of singing. The result is an array, $f_0[n]$, that contains a pitch value per frame. As shown in Figure 2, we assume the segments detected as unvoiced by MELODIA to be *falsetas*.

As a parameter of the algorithm, we set the frequency range between 120 Hz and 720 Hz to track the vocals by using an analysis window of 4096 samples, as suggested by the paper authors (Gómez et al., 2012).

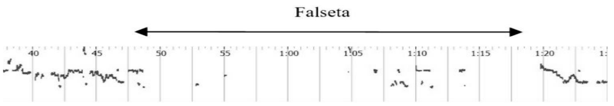

Figure 2: *Visualization of the vocal melodic line extracted using MELODIA. The segment with no melody is likely to be a falseta.*

3.1.3 Contour Classification

In this section, we classify each frame as voiced/unvoiced if it corresponds to vocal content or not. We will focus on unvoiced segments as candidates for guitar *falsetas*. This process adapts the one proposed in (Kroher & Gómez, 2016) to get rid of the guitar sections based on spectral features differences as exemplified in Figure 3. We first extract the energy in the lower twelve bark bands (see Eq. 2) computed frame-wise to carry out a preliminary discrimination.

$$B[n,m] = \sum_{k(f_{1,m})<k<k(f_{2,m})} |X[k,n]|^2 \qquad (2)$$

The result for each band m, delimited by f_1 (lower frequency limit) and f_2 (upper frequency limit) and where $k(f)$ is the frequency bin corresponding to the frequency f, is stored in a a 12-length vector $\vec{x}$ for each analyzed frame n. By using predominant melody information (i.e. the output of MELODIA), an initial label is assigned to each vector $\vec{x}$: the melody frames are marked as voiced and the non-melody frames as unvoiced. We then compute the mean and the covariance for both groups and fit a single multivariate Gaussian distribution to both sets separately. The fitting process is done for each recording and no training is needed beforehand. We obtain a probability p for each element to be voiced or unvoiced and we perform a binary classification taking into account the highest probability.

In order to avoid fast fluctuations in this prediction, a binary moving average filter of length 1 second is applied to make the vocal detection smoother. Finally, we search for segments of consecutive melody frames in f_0 and evaluate the result of the prediction for each of them. Those segments where all the prediction values are equal to zero, i.e. non-melodic according to MELODIA, are removed from the f_0 list because following our hypothesis, they will correspond to *falsetas*, and their boundaries are then used for segmentation. Instead of directly removing the vocal sections, we observed that if we first eliminate the guitar parts, as in *PyCante*, the *falsetas* delimitation is more accurate.

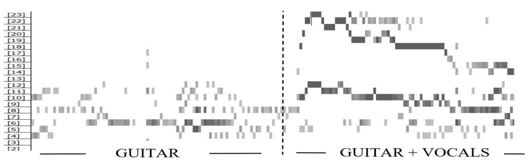

Figure 3: *Bark coefficients representation for vocal and guitar sections*

3.1.4 Falsetas identification and segmentation

In the last step of the segmentation, we locate the null segments in f_0 (see Figure 4) and extract their boundaries. These segments correspond to the non-vocal parts of the recording and thus we assume they are guitar sections. Finally, we compute the duration of each segment: if it is longer than the minimum duration specified by the user (15 seconds by default), it is classified as a *falseta*. Otherwise, we eliminate the segment.

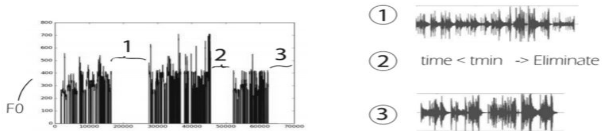

Figure 4: *Identification and evaluation of the falsetas candidates*

The output of the segmentation stage consists of an audio wave file that contains a concatenation of all the detected *falsetas*. Since we can recover the original audio file, if it is a stereo recording we obtain one audio file per channel. In addition, the system also generates a text file with the boundaries (i.e. start and end points) of each **falseta** in seconds. This information is further used for evaluation.

3.2 Transcription

We use the output of the previous stage to obtain a symbolic representation (i.e. a MIDI file) of the *falsetas*. First, we analyze the guitar melody with a pitch tracking algorithm and then we find the onsets and offsets of the notes to define their boundaries. We finally label each note with its corresponding pitch value. All the steps are detailed in the following subsections.

3.2.1 Guitar melody extraction

If the input is a stereo audio file, we compute the mean of both channels because although the guitar is more predominant in one of them, there is still some guitar in the other one that we also need for a complete analysis. At this point, we extract the guitar melody using an algorithm proposed by (Klapuri, 2006) and implemented in the *Essentia* library (Bogdanov et al., 2013). This method estimates multiple pitch values per frame, which correspond to the melodic lines present in a polyphonic music signal. By default, the transcription is monophonic but it can also be polyphonic as a user's choice:

- For monophonic *falsetas*, the algorithm only selects the first frequency value for each frame. We restrict the frequency range to 80-750 Hz, which corresponds to the guitar range; however, this parameter that can be modified.

- If we have a polyphonic guitar line, as a limitation of our algorithm, we take a maximum of two values within the mentioned frequency guitar range for each sample. Due to restrictions of the multi-pitch tracking algorithm, in the polyphonic case we do not limit the frequency range; instead, we remove the values which are out of the range afterwards.

3.2.2 Onset and offset detection

In this step we present the methodology used for note segmentation. We consider typical flamenco guitar techniques such as contiguous notes that can be tied together, as well as very fast *staccato* notes, called *picado*, played with the index or middle finger, or *alzapúa* played with the thumb.

For onset detection we use an algorithm based on diverse novelty functions (Dixon, 2006), which is implemented in *Essentia*. In our case, even though the guitar strings have a percussive component, we consider spectral features, specifically the spectral flux. This novelty function is obtained by computing the euclidean distance between two consecutive and normalized spectra. This method provides the best results for instruments like guitar, defined as pitched and percussive by (Dixon, 2006).

To determine the offsets, we computed the average duration of all notes within the transcription dataset which was found to be $d_{avg} = 0.16s$. If we consider subsegments as sections between onsets:

- If the subsegment is shorter than d_{avg}, the offset is set as the previous sample of the next onset. In this case, we consider that the subsegment is too short and does not need to be analyzed in depth because the note is muted by immediately playing another one on the same string.

- Otherwise, we analyze the energy in each subsegment as showed in Figure 5. We compute the RMS (root mean square) using a window of size $N = 256$ and define E_{max} as the maximum energy value within the subsegment. We also define $thr = 0.1 \cdot E_{max}$. We define the offset as the first value that fulfills the condition: $E_m < thr$.

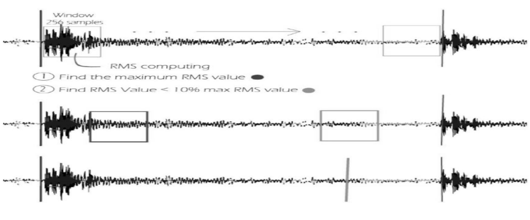

Figure 5: *Offset detection process*

3.2.3 Pitch estimation and labeling

After delimiting each note in time, we analyze its pitch content as depicted in Figure 6. We first convert pitch values from Hz to cents, using $f_T = 440$ Hz as the reference tone. Then, we compute a local pitch histogram for each note, $H[f_{cent}]$, and choose the most common pitch value - as long as it belongs to the guitar frequency range defined in previous sections. For polyphonic cases, we repeat this process for the second melodic line. Finally, we convert the obtained pitch values in cents into MIDI notes using:

$$MIDI_{note} = \left\lceil 12 \cdot log_2\left(\frac{f}{f_T}\right) + MIDI_{ref} \right\rceil \quad (3)$$

Given the output of the onset detection function, we label each note by aligning each onset value (in frames) with the original signal to obtain the actual points in time (seconds) using Eq. (4). Afterwards, we use the computed offsets to find the duration of each note.

$$oTime = onset \cdot \frac{HopSize}{f_s} \quad (4)$$

Together with the MIDI note, the onset and the duration, we also add an energy value which is closely related to the volume for each note to provide better perceptual results when listening to the MIDI file. To this end, we use the energy function included in the *Essentia* library. Instead of the regular MIDI range (0-127), the output is bounded between 40 and 100 in order to avoid abrupt volume changes between notes.

With this information we create the resulting MIDI file using the MIDIUtil Python library [7], for which we need to set a *tempo* value in *bpm* (beats per minute). We use the algorithm proposed in (Percival & Tzanetakis, 2014) to estimate the tempo of the input recording and use it as a default value. This *tempo* can be later modified by the user using any sequencer or MIDI editor. Finally, each note is defined by an onset time, duration, pitch and energy value, and all this information is stored in a MIDI file as well as in a CSV (comma-separated values) file.

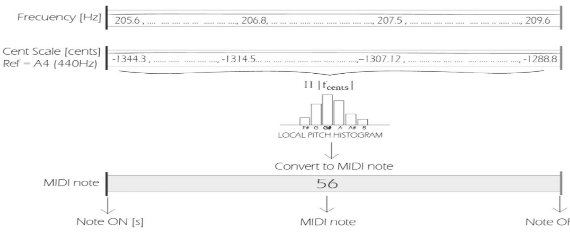

Figure 6: *Pitch labeling process*

3.3 Post-processing

In the last stage of the proposed system we aim to adjust some potential errors using pitch re-scaling: this method can only be applied in cases where the user knows the scale of the piece. It consists of scanning the whole set of notes and re-scaling the pitch values using tonal features and typical scales of flamenco music (Fernandez, 2004). Our system includes the following scales:

- From modern Western modes, the *Phrygian* mode on E and *Phrygian* dominant scale produced by raising the third scale degree when ascending as we displayed in Figure 6.

- A *Flamenco* mode which arises from the previous scale but using a transposition of two tones and a half as shown in Figure 8

[7] https://pypi.python.org/pypi/MIDIUtil/

Figure 7: *Phrygian dominant scale also called Spanish Gypsy Scale*

Figure 8: *Phrygian mode transposed to A*

- Major scale based on E shown in Figure 9

Figure 9: *E Major scale*

In flamenco, it is common to adapt the pitch range of the *falseta* to the pitch range of the singer by means of a *capo*. In order to allow for that we implement a transposition function where the user can specify a number of semitones.

4. EVALUATION METHODOLOGY

We evaluate the segmentation and transcription stages separately using the two datasets (see Section 2) and two different methods, both included in the *mir_eval* library (Raffel et al., 2014).

Regarding the segmentation stage, we evaluate two different aspects: first, the number of *falsetas* longer than fifteen seconds found by the algorithm, and second, the precision of their boundaries. In flamenco pieces, we usually hear short guitar resources used to open and close *falsetas* called *llamadas*, *remates* or *cierres*. Since it is not clear whether to consider them as part of the *falseta* or not, we use a tolerance window of size $N = 4$ seconds to evaluate the boundaries of each *falseta*.

For the transcription stage, we evaluate the onset, offset, and pitch value for each note in the *falseta* using the *transcription* method in *mir_eval*. According to MIREX 2015 [8], this method assumes an estimated note to be correct if its pitch value is within ± quarter tone of the corresponding reference note. Regarding the onset and offset rules, we increase the tolerance from ±50 ms to ± 100 ms because of the relative inaccuracy of the manual annotations in the transcription dataset.

4.1 Standard Metrics

To evaluate the performance of our system we use the three standard information retrieval evaluation metrics: Precision (P), Recall (R), F-measure (F):

$$P = \frac{c}{c + f^{+}}, R = \frac{c}{c + f^{-}}, F = \frac{2 \cdot P \cdot R}{P + R}, \qquad (5)$$

[8] http://www.music-ir.org/mirex/wiki/2015:Main_Page

Where c is the number of correct detections, and f^{+} and f^{-} represent the number of false positives and false negatives respectively. For the transcription stage, we also obtain the average overlap ratio (AOR) as the mean overlap ratio computed over all matching reference and estimated notes.

5. RESULTS

We first examine the number of *falsetas* identified for each song without taking into account the segmentation accuracy, i.e. we count the number of *falsetas* that the algorithm finds even though their boundaries are not exact, and find that our approach is capable of locating as many *falsetas* as there are in the ground truth. This accuracy confirms that this method is able to discern the guitar sections in a flamenco piece.

The results of the evaluation of the second part of the segmentation stage (the delimitation of the *falsetas*) and transcription stage are summarized in Table 1, averaged for all the *falsetas*. Notice that even though we obtained an accuracy of 100% in the first part of the segmentation stage (i.e. the number of *falsetas*), the average precision of their boundaries falls to 75%. We observe that our system is better at segmenting *falsetas* than at transcribing them, which suggests that there is room for improvement, especially in this second step.

Stage	P	R	F	AOR
Segmentation (boundaries)	0.75	0.77	0.76	-
Transcription	0.61	0.62	0.615	0.618

Table 1: Results for both stages (independently) according to the methodology detailed in Section 4.

To understand the limitations of our system, we measure how pitch re-scaling (PRS), detailed in Section 3.3, affects the performance of the algorithm. We evaluate two excerpts of our dataset using the PRS procedure: the first one (*Soleá* 1) is a monophonic *falseta* which has a duration of 6 seconds and is re-scaled using the *Phrygian* dominant scale. The second one (*Alegrias*) corresponds to a 30 seconds polyphonic piece and is re-scaled using the E major scale. Table 2 shows these results, and illustrates an increase in precision and recall. This suggests that adding specific musical knowledge, such as the scale of the song, to the algorihm has a positive impact on the performance.

Data	P	R	F	AOR
Soleá 1	0.79	0.81	0.80	0.71
Soleá 1 (PRS)	0.823	0.848	0.83	0.718
Alegrias	0.735	0.59	0.654	0.412
Alegrias (PRS)	0.756	0.61	0.674	0.415

Table 2: Impact of the PRS applied on two specific cases

Since we are not able to compare the transcription results with any existing system, we manually inspect the

MIDI files together with their corresponding audio input. We observe that the onset detection is quite unstable and provides significantly different results throughout the dataset because of the wide range of different techniques used in flamenco guitar. To measure the performance of onset detection both perceptually and using standard metrics, we evaluate the system using two methods based on different novelty functions as shown in Table 3: the complex domain spectral difference function (Complex) and the high frequency content (HFC) detection. We compare these results with the spectral flux function used by default. Since the guitar is a pitched percussive instrument, we also include the evaluation removing the offset rule.

Method	P	R	F	AOR
Spectral Flux	0.61	0.62	0.615	0.618
Complex	0.608	0.607	0.60	0.61
HFC	0.618	0.496	0.54	0.59
Spectral Flux (no offset)	0.65	0.661	0.65	0.629
Complex (no offset)	0.658	0.656	0.652	0.614
HFC (no offset)	0.68	0.553	0.607	0.563

Table 3: Results for both stages according to the methodology detailed in Section 4

By manual inspection of the results, we observe that the HFC method is useful for regions with transients, but can be misleading when hand-clapping appears. The *complex* method works properly for pitch-changing notes (*legato* or *glissando*) but not for fast notes. As mentioned in Section 3.2.2, the spectral flux method performs well for pitched and percussive notes, although it still provides unstable results for techniques such as *glissando*, flamenco *tremolo*[9] or *alzapúa*.

6. CONCLUSIONS AND FUTURE WORK

We have addressed the problem of *falseta* detection and transcription and we consider that the obtained results are satisfactory. The dataset collection has been one of the most challenging tasks in our project, given the lack of scores and related research. Due to this fact, our dataset still contains few samples and needs to be expanded to obtain more representative results. In spite of that, we think that this work provides a good starting point for further research in this problem.

We consider that the segmentation provides reliable results but the system would sometimes need to disambiguate what is considered as a *falseta* or not. Our segmentation method is limited to pieces that contain dialogs between *cante* and *toque*. If a new instrument is present and has its own sections, the system will probably classify it as a guitar *falseta*. As a future work, we suggest to use spectral features to create timbre spaces allowing the discrimination between a varied set of instruments. Regarding onset detection, a multimodal fusion technique could be used to stabilize the results by improving the precision also for those techniques in which weak transients can be included.

[9] https://www.atrafana.com/flamenco-guitar-techniques–tremolo.html

7. ACKNOWLEDGMENTS

This work is partially supported by the Spanish Ministry of Economy and Competitiveness under the CASAS project (TIN2015-70816-R).

8. REFERENCES

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J. R., & Serra, X. (2013). Essentia: an audio analysis library for music information retrieval. In *ISMIR'13*, (pp. 493–498).

Dixon, S. (2006). Onset Detection Revisited. *Proc. of the 9th Int. Conference on Digital Audio Effects (DAFx-06)*, 1–6.

Fernandez, L. (2004). *Teoria musical del flamenco*. Acordes Concert.

Gómez, E. & Bonada, J. (2013). Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a cappella singing. *Computer Music Journal*, *37*, 73–90.

Gómez, E., Cañadas, F., Salamon, J., Bonada, J., Vera, P., & Cabañas, P. (2012). Predominant fundamental frequency estimation vs singing voice separation for the automatic transcription of accompanied flamenco singing. In *13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, Porto.

Klapuri, A. (2006). Multiple Fundamental Frequency Estimation by Summing Harmonic Amplitudes. *Proceedings of the International Symposium/Conference on Music Information Retrieval (ISMIR)*, 216–221.

Kroher, N. & Gómez, E. (2016). Automatic transcription of flamenco singing from polyphonic music recordings. In *IEEE/ACM Transactions on Audio Speech and Language Processing*, volume 24, (pp. 901–913).

Kroher, N., Gómez, E., Guastavino, C., Gómez-Martín, F., & Bonada, J. (2014). Computational models for perceived melodic similarity in a cappella flamenco cantes. In *15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan.

Percival, G. & Tzanetakis, G. (2014). Streamlined Tempo Estimation Based on Autocorrelation and Cross-correlation With Pulses. *IEEE/ACM TRANSACTIONS ON AUDIO*, *22*(12).

Raffel, C., Mcfee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., & Ellis, D. P. W. (2014). mir_eval: A TRANSPARENT IMPLEMENTATION OF COMMON MIR METRICS. In *15th International Society for Music Information Retrieval Conference*.

Salamon, J. & Gomez, E. (2012). Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech and Language Processing*, *20*(6), 1759–1770.

Salamon, J., Rocha, B., & Gómez, E. Musical Genre Classification using Melody Features Extracted from Polyphonic Music Signals.