RUMBATOR: A FLAMENCO RUMBA COVER VERSION GENERATOR BASED ON AUDIO PROCESSING AT NOTE-LEVEL

Carles Roig, Isabel Barbancho, Emilio Molina,
*Lorenzo J. Tardón and Ana María Barbancho**

Dept. Ingeniería de Comunicaciones, E.T.S. Ingeniería de Telecomunicación,
Universidad de Málaga, Campus Universitario de Teatinos s/n, 29071, Málaga, Spain

carles@ic.uma.es, ibp@ic.uma.es, emm@ic.uma.es,
lorenzo@ic.uma.es, abp@ic.uma.es

ABSTRACT

In this article, a scheme to automatically generate polyphonic flamenco rumba versions from monophonic melodies is presented. Firstly, we provide an analysis about the parameters that defines the flamenco rumba, and then, we propose a method for transforming a generic monophonic audio signal into such a style. Our method firstly transcribes the monophonic audio signal into a symbolic representation, and then a set of note-level audio transformations based on music theory is applied to the monophonic audio signal in order to transform it to the polyphonic flamenco rumba style. Some audio examples of this transformation software are also provided.

1. INTRODUCTION

A lot of research has been done by the audio signal processing community in the field of audio transformation [1][2]. In this context, an innovative approach for automatic music style transformation is presented in this paper. The objective of this work is to implement an unattended style transformation process from an undetermined style to flamenco rumba. A similar objective is performed by Songify, [3]. The pitch adaptation performed by Songify implements a vocoder synthesizer [4]. Rumbator is based on a different approach: The output synthesis uses a transformation of the original signal, thus achieving better audio quality, compared to the robotic effect of the phase vocoder. Furthermore, whereas Songify's target style is electronic music, the system presented in this contribution, Rumbator, aims at a transformation into flamenco rumba style.

Another goal, apart from the entertaining purpose, is an appealing illustration of the main characteristics of a particular flamenco form (namely *palo*), that differentiate it from other kinds of flamenco forms and styles. The emphasized features are the particular harmony progression with a special cadence extensively used in most of the flamenco songs and a specific rhythmic structure. A secondary objective of this work is to promote and make known this particular music style that belongs to the Spanish cultural heritage. In addition, the transformation will generate a score sheet with the transformed melody, so the user can observe the changes performed and sing or play with an instrument the flamenco rumba version automatically generated by Rumbator.

The aspects that constitute targets of the transformation are rhythm and harmony. Flamenco rumba compositions are based on 4/4 measures. The accompaniment is composed repeating a group of two *tresillo rhythms* with eight beats (3+3+2) each. In Figure 1, the basic rhythm of the flamenco rumba is shown. The harmony progression is a repeated loop of four chords. The applied chord wheel corresponds to the Andalusian cadence (I-VII-VI-V), which is very commonly used in flamenco music [5]. Tempo is typically slower than in other flamenco styles, approximately 100-120 bpm. Nevertheless, in order to respect the original meter, the accompaniment pulse will be adapted to the input sound. Flamenco rumba may be composed in any key, major and minor keys. In this paper, however, the key is fixed to A minor, so the chord wheel will be Am-G-F-E [5].

Figure 1: *Basic rhythm of flamenco rumba accompaniment.*

The rest of the paper is organized as follows. In Section 2, a reference to similar projects is made, and the contribution of this paper is outlined. In Section 3, the proposed system is presented and described. In Section 4, the transcription process is explained. Next, in Section 5, the analysis process for the tempo estimation is presented. In Section 6, the set of algorithms that will form a part of the flamenco rumba style transformation, such as rhythm organization, harmony adaptation and tempo adjustment, are introduced. Finally, in Section 7, the conclusions and a discussion of this work are presented.

2. RELATED WORK

The proposed system resorts to transformations adapted to the input [6], focusing on the harmony progression and rhythm. Since the melody contour is crucial for the identification of melodies [7], it must be maintained in order to generate a melody similar to the

* This work has been funded by the Ministerio de Economía y Competitividad of the Spanish Government under Project No. TIN2010-21089-C03-02 and Project No. IPT-2011-0885-430000 and by the Junta de Andalucía under Project No. P11-TIC-7154. This work has been done at Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech.

original one. Global pitch transposition, as performed in [8], will not work, as the input sound might not follow the proper harmonic progression. Thus, since pitch contour modification has to be done separately, a note-level processing scheme is applied. This means that the system requires a melody transcription stage. Note that this idea is similar to Melodyne's [9] approach, which allows individual note transformations to performed. In our case, the transcription method is based on the approach presented in [10]. This method is a pitch-based segmentation with a hysteresis cycle. It was selected because of its simplicity compared to methods like HMM [11], and its robustness against false positives [12].

In order to extract the original rhythm structure and pitch evolution, in particular for the proper identification of the rhythm figures, an analysis of the input tempo is required. Uhle and Herre [13] presented an approach based on the spectral analysis and a subsequent study of the repetitions that estimates the length of the bar. On the other hand, the algorithm by Gouyon et al. [14] is based on the extraction of a histogram in order to obtain the most repeated inter-onset value, and then calculate the tempo, avoiding the spectral analysis. The novel algorithm proposed in this work is based on the framework shown in [14]. However, since our system deals with acapella audio excerpts recorded by the user, the tempo extractor has to deal with the absence of percussive tracks, that represent a fundamental feature in [14]. Thus, some modifications have been done related to the onset detection. More precisely, the onsets are obtained from the note timestamps (as if it was a MIDI file) that are obtained from the segmentation. In Section 5, the algorithm is presented with some evaluation results.

The algorithms for the analysis and harmony adaptation process are novel computational implementations of harmony adaptation methods based on musical concepts and music theory, avoiding complex machine learning systems. They rely on the simple concept of harmony based on the accented tones [15], applied to harmony adaptation by pitch level change method [16], whereas other approaches use musical concepts for the melody adaptation. In [17], for example, the harmonic dissonances on the melody are corrected by the interpolation of new consonant tones using counterpoint rules.

The main novelty of this project is based on the development of an automatic process that is able to generate a cover version of a certain input. Furthermore, both analysis and adaptation algorithms are novel implementations for temporal estimation and harmony adaptation, respectively. The temporal estimation framework is similar to [14], but adapted to MIDI files and to the absence of percussive references that helps the tempo estimation. By this, necessities of correlation are eliminated, since the algorithm deals with onsets. Concerning the melody contour adapter, the novelty consists in the dynamic adjustment of the harmony while keeping the melodic contour intact. The algorithm has been designed to perform beyond a dissonance corrector [17] by adding the constraint of maintaining the melodic envelope. The output cover will thus resemble more strongly to the original excerpt. The use of musical concepts instead of performing a machine learning process is also innovative.

3. PROPOSED SYSTEM

The scheme of the system is presented in Figure 2. The output will be composed of the properly transformed (harmony and rhythm) input sound mixed with a guitar rumba accompaniment. Since the accompaniment is pre-recorded, temporal estimations are also required for the tempo adjustment of the accompaniment.

As illustrated in Figure 2, as a first step, the original signal is transcribed into separated notes. Then the estimation and transformations can be carried out at note level. Next, the duration information (focused on the inter onset interval) is used for the estimation of the tempo. This set of data is useful for the proper establishment of the tempo, for both adapting the tempo of the accompaniment to the input signal, and the rhythm transcription, enabling the separation of the input sound into measures.

Whence, a chord of the Andalusian cadence is assigned as a target chord to each bar in a cyclic way. Once the chord progression has been related to all the measures that compose the melody, the harmony adaptation is performed. This process is based on music concepts (chord tones and non-chord tones position within the measure [15]) for changing the harmony of the measures, and adapting them to the accompaniment [16].

Finally, after the input sound has been rhythmically organized and harmonically adapted, the transformed input and the accompaniment (with the adjusted tempo) are mixed.

In the following sections, each of the subsystem is described in detail.

4. TRANSCRIPTION

The method used for audio transcription is based on the approach presented in [10]. This approach is a pitch-based segmentation with a hysteresis cycle to avoid spurious note detections. The method converts monophonic audio signals into a symbolic representation based on individual notes, with information of their pitch and duration. The procedure can be divided into three steps: (1) detection of voiced/unvoiced regions, (2) note segmentation, and (3) note pitch estimation.

The first step is the estimation of the $f0$ curve [10] using the YIN algorithm [18]. This is a simple and low cost approach for this purpose.

Once the $f0$ curve is obtained, the algorithm will detect the voiced/unvoiced regions. A region will be labelled voiced if $f0$ is stable. Furthermore, in order to increase the accuracy of the process, other descriptors are taken into account for the voiced/unvoiced detection. The mean power of all the previous segments, the aperiodicity [18], and the frequency stability [19] are the three descriptors analysed for the voiced/unvoiced detection.

Once the stable regions of pitch are detected, a second segmentation is required for splitting legato notes. According to [12] and [11], the method used for the analysis of the stability of the voiced region and the segmentation of notes leads to the idea of pitch centres. When the pitch deviation around the estimated pitch centre is sustained or very abrupt, a note change is considered, and the computation of a new pitch centre starts.

The estimation of the pitch centre is performed dynamically averaging the growing segment (α-trimmed weighted mean of the pitch curve [20]):

$$X_\alpha = \frac{1}{N - 2\lceil \alpha N \rceil} \sum_{i=\lceil \alpha N \rceil + 1}^{N - \lceil \alpha N \rceil} X_i \qquad (1)$$

where $\lceil \cdot \rceil$ is the ceiling function, $\alpha \in [0, 0.5]$ is the amount of values *trimmed* and X_i represents the i-th element in the sorted vector ($X_1 \leq X_2 \leq \cdots \leq X_N$). Note that this estimation becomes more stable as the duration of the notes increases.

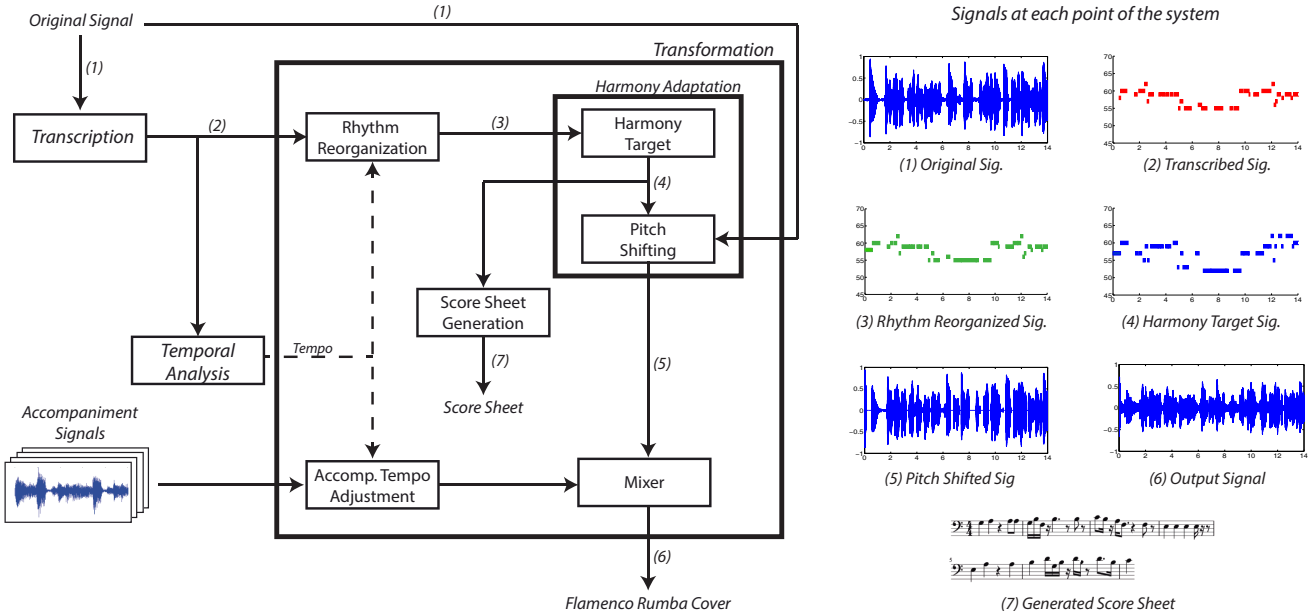

Figure 2: *Block diagram of the Rumbator system. Emphasized blocks (in bold) correspond to transformation processes explained in Section 6*

5. TEMPORAL ANALYSIS

Once the input data is segmented, the tempo has to be estimated for a proper bar separation. The starting point will be the extraction of the inter onset interval (IOI) of the segmented input signal [21]. As commented above, the main challenge in this scenario is the onset detection dealing with a monophonic acapella singing signal. In this case, the starting point of the voiced segments obtained by the transcription subsystem will be considered as the onset used for the tempo estimation. Considering that the system does not perform a perfect transcription, the accuracy of the onset position will be affected, and the tempo estimation will be less precise. However, since the objective of this system is not the extraction of an accurate tempo, but the extraction of the rhythmic structure and its adaptation to a proper flamenco accompaniment, an approximate tempo is sufficient. Thus, the IOIs, together with their histogram, are computed using the onsets provided by the segmentation process.

After computing the histogram of the IOIs obtained from the segmented sound, the most often repeated value will be considered a beat candidate or, more technically, the tactus [22] candidate of the input melody. Figure 3 shows the IOI histogram and the extraction of the tactus candidate.

Due to the discrete nature of the histogram, the tactus candidate has to be finely adjusted, in order to find the proper tempo with the lowest error between the estimated one (established from the tactus candidate) and the actual onsets of the segmented audio. Thus, the temporal estimation error will be as small as possible.

The procedure consists of the definition of a set of n equally distributed values between three quarters and five quarters of the estimated tactus. These will be considered as candidates, evaluating the global tempo error for each of these values, to find the optimal tactus. The idea is to find the tactus that will cause the fewest note shifting in the rhythm reorganization subsystem.

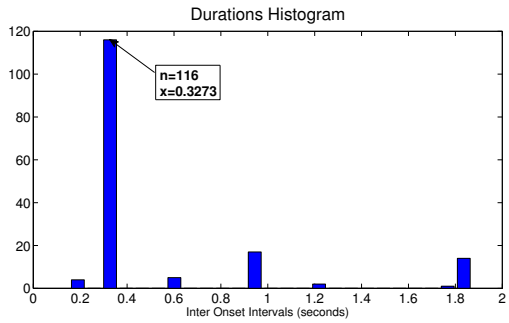

Figure 3: *The histogram of the inter onset intervals (IOI) of the transcribed melody. The maximum of the histogram set the candidate of the tactus to* 0.3409 *seconds.*

The algorithm will generate one rhythmic grid for each tactus candidate (a vector of multiples of the tactus, equivalent to the pulse positions for a particular tempo related to the tactus candidate). The grid global error is the accumulation of the distance from the onset time of each note of the transcribed melody to the closest point of the grid (the closest pulse position). It is computed for each candidate by using the error expression, Eq. 2:

$$e_j = \sum_{i=1}^{TN} \min_k (g_{jk} - IOI_i) \qquad (2)$$

where e_j is the global error of the j-th candidate, TN is the number of notes of the transcribed melody, g_{jk} is the k-th entry of the grid vector corresponding to the j-th candidate (a vector with the pulse positions related to the j-th tactus candidate), and IOI_i is the onset position of i-th note of the transcribed melody.

The candidate with the lowest error will be considered the optimal tactus. Figure 4 presents an example of tactus optimization. The optimized tactus attains the lowest error with respect to the original onsets.

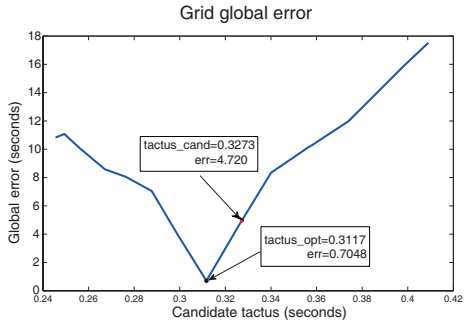

Figure 4: *Global error for the set of tactus candidates. The optimized tactus attains the lowest error (0.7048 seconds).*

As discussed earlier, the segmentation process adds an error to the measure. So in order to evaluate the algorithm, two tests have been done. The first experiment was designed to evaluate the accuracy of the tempo extractor implementation without the influence of the transcriptor. The input dataset was a group of MIDI files, with a duration of 30 seconds each, and a known tempo. These MIDI files were created manually, such that the duration of each of the rhythmic elements has the proper duration according to the tempo (e.g. if the tempo is 100 bpm, the quarter note duration is 0.6 seconds). Since the complete system receives a kind of MIDI file (with more information) from the segmentation, the system has not been modified for the experiment. The results of the first experiment are shown in the Table 1.

Table 1: *Experiment 1 (ideal case): Using a MIDI input file with quantized durations. The table presents the real tempo, the estimation done by the algorithm, and the number of inter onset intervals used for generating the histogram (equivalent to the number of transcribed notes).*

Sample	Actual Tempo	Estimated Tempo	IOI used
#1	125	124.75	89
#2	125	124.85	87
#3	125	124.83	90
#4	125	124.21	117
#5	100	100.15	92
#6	100	99.27	120
#7	90	90.02	91
#8	90	89.84	101
#9	90	89.99	65
#10	85	84.98	54
#11	85	84.87	61
#12	85	84.99	62

The second experiment was aimed at evaluating the complete temporal analysis, taking into account the effect of the transcriptor in the measure. It was based on the same MIDI files used in the first experiment, but in this case the melody was sung by a human. Thus the input in this case were 12 wave files, with a duration of 30 seconds each. By this, the conditions of the usual application of the system are simulated, i.e. the instability of the pitch and the inexactitude (and variability) of the rhythm performed by a real person. The purpose of this experiment is to measure the robustness of the algorithm against deviations caused by the human. The performer had not a rhythmic reference as the usual application of the system. This experiment also allows the observation of the error added by the processing chain of the segmentation process. The results of the second experiment are shown in Table 2.

Table 2: *Experiment 2 (real case): Using a WAV input file with an actual human performance of the previous MIDI. The table presents the real tempo and the estimation done by the algorithm.*

Sample	Actual Tempo	Estimated Tempo
#1	125	135.05
#2	125	137.05
#3	125	117.42
#4	125	135.01
#5	100	101.75
#6	100	109.62
#7	90	99.10
#8	90	100.20
#9	90	98.63
#10	85	97.65
#11	85	85.98
#12	85	93.01

There is a noticeable deviation of the estimated tempo of about 10 bpm (on average) above the original tempo. However, it is a good estimation, considering that the input signal is an acapella signal (lacking rhythmic references compared to polyphonic signals and percussive tracks), and that the user has no reference sound, possibly making the tempo unstable. Furthermore, regarding our goals, the observed approximation suffices to recover the original rhythmic structure of the input signal, so we can split the sound in measures and adapt to the accompaniment.

Since the time signature of the rumba is 4/4, the input sound should be modified to fulfil this constraint. In order not to restrict the time signature of the input sound, its rhythmic structure is adapted that of the rumba. The disadvantage of this approach is that the original accents of the input sound will be passed over. However, this choice enables the creation of unexpected and interesting music content. A rhythmic reorganization is required for the proper placement of the original segments on the beat onsets of the accompaniment generating a coherent structural rhythm in the final composition. This process will be explained in detail in the following section.

6. TRANSFORMATION

As shown in Figure 2, there are three transformation processes in the scheme: the rhythm reorganization and the harmony adaptation, applied to the segmented signal of the input sound, and a temporal adjustment of the accompaniment signals. In what follows, these methods will be described in detail.

6.1. Rhythm Reorganization

As previously indicated, the time signature of the input waveform will not be considered, so that the segments will not be organized by their accent, but by their duration and position. Since the objective now is matching the segments with the accompaniment downbeats, the beat obtained from the tempo estimation is used to construct a rhythmic grid in which each point of the grid corresponds to a quarter note related to the estimated tempo. The aim is to apply rhythmic modifications that are as small as possible. Since the tempo is estimated from the original signal, the rhythmic grid will be well adapted to the segment positions. Also, the rhythm reorganization process will reallocate the segments in coherence with the flamenco rumba structure, even changing the original accent scheme.

The segments obtained from the segmentation, and the resting periods (unvoiced regions), will be arranged orderly according to this grid, so that the segments will be placed on the measure downbeat, creating a new rhythmic structure. However, placing segments orderly at each point of the grid can cause overlapping between segments. To deal with this issue, the proposed method will first determine the segments that belong to each beat. The beat assigned will be the one that owns more part of the duration of the segment. Then, all segments are time stretched to fit in the note period assigned. The time stretching factor applied to segments in a beat is the same, so the ratio between them is kept. Within each beat, all segments that belong to this beat are stretched by the same factor. If the duration of the segments that belong to a certain beat is less than the duration of a quarter note, the notes will be placed side by side at the beginning of the beat without applying any time stretching process. Furthermore, if a resting period is longer than a beat, it will be treated as a common note, so the important pauses are maintained.

Figure 5 illustrates the result of a rhythmic reorganization example. The time scaling process applied to the segments in this stage is the same as performed to the accompaniment for the tempo adaptation. This procedure will be explained later, in Subsection 6.3.

Once the rhythm is fully reorganized, it is easy to split the excerpt into measures by grouping the elements every four beats. Each of the measures will then be assigned to a certain harmony, i.e. each measure will be harmonically adapted to a certain target chord. This process is explained in the next section.

6.2. Harmony Adaptation

Harmony perception is strongly related to the chords placed on the downbeats of the measure [15]. The notes on the downbeats are responsible for the harmony definition, and the chords with many tones in common with the accented tones will be good candidates for harmonizing the measure with natural sound [15]. The task of the harmony adapter, however, will be the opposite: given a particular chord in the accompaniment, the accented notes will be moved to the closest tone that fulfils the harmony condition while maintaining the melodic contour. This method was coined in the music theory as *harmony adaptation by pitch level change* [16].

Thus, the first task of the harmony adapter is to identify the accented notes. As the rumba measure is 4/4, the downbeats correspond to the onset of the four quarter notes in the measure [5]. The downbeats of the 4/4 measure therefore match with the grid position of the previously organized measure. In other words, each quarter note position in a 4/4 will be considered considered chord tones [1] [15], and will be adapted to fulfil the harmony established by the harmony structure. On the other hand, the segments placed in weak parts of the measure will not be forced to belong to the given chord and will be adapted to keep the melodic contour of the input melody as a passing note. Note that in some particular cases, the real chord notes are not placed in the measure downbeats [15]. In our setting, however, where the most important feature is the pitch contour, the model is simplified to the most common case, where the chord notes are placed in measure downbeats.

As mentioned before, the process for the harmony adaptation depends on the note type:

- Accented notes (or real notes [15]) must belong to the established chord
  - First chord note in the melody: It is assigned to the closest pitch that belongs to the chord, as a starting tone.
  - Other chord notes: In order to keep the original pitch contour, secondary accented notes are moved to the closest pitch in the contour direction.
- Unaccented notes do not necessarily belong to the chord. In this case, the original interval between the previous note and the current note is added or subtracted. In order words, if the unaccented note in the original pitch contour was two tones higher than the previous note, the final progression will keep the interval, resulting in a note that is again two tones higher than the previous note (which has already been adapted to the harmony).

Figure 6 depicts an example of the harmony adapter process.

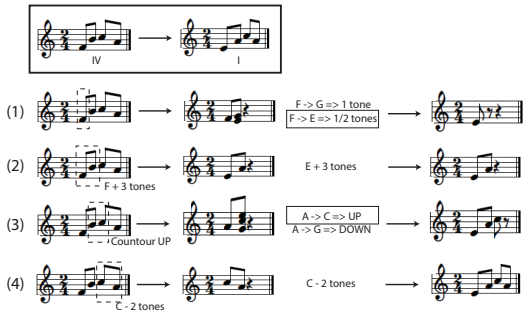

Figure 6: *Example of harmony adaptation from fourth chord to first one.*

The steps performed in the example in Figure 6 are the following: (1) The first note is moved to the closest tone belonging to the chord (C Major first chord is C-E-G). It is closer to move from F to E (half tone), than to G (one tone). (2) Since B is placed on an upbeat, it is considered a passing note, and the interval to the original previous note has to be restored. The original interval was 3 tones, so starting from the previously adapted note, E plus three tones is A. (Actually from E to A there are 2 tones and a half, but since A# does not belong to the tonality, the final note is set as A.) (3) The third note is placed on the downbeat, so it has to be moved to a chord note. In this case the chord notes closest to A (previously

[1] the ones placed in second and fourth division will be weighted and will be given more importance since these positions are the strongest in the measure

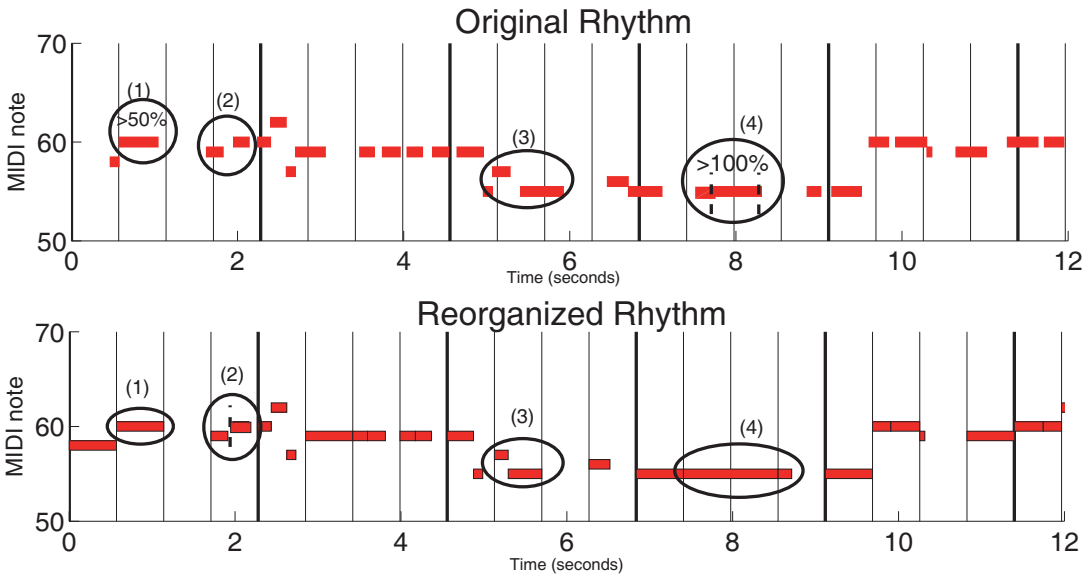

Figure 5: *In the upper plot, the rhythm obtained as an output of the segmentation. In the lower plot, the final rhythm structure after the reorganization. Some particular examples are emphasized. (1) The note duration is longer than 50% of a quarter note, so it is fit to the complete duration. (2) Two short notes (less than 50% of the quarter note) are fit together at the beginning of the beat. (3) The notes are fit in one beat since the complete duration is bigger than one beat and both have most part of their duration in the same beat. (4) The note duration is bigger than one beat and its position occupies more than three subdivisions, so it is fit in two beats.*

adapted) are G or C. As the original contour direction is upwards, the selected note will be C, in order to keep the contour. (4) Similar to the second step, as A is placed on an upbeat, the original interval has to be restored. This is two tones below the previously adapted note: C minus two tones is Ab, but as in the previous case, Ab does not belong to C Major, so it will be left natural.

This process generates a MIDI file that indicates the pitch evolution target which the segmented audio samples has to adapt to. An example of such a pitch target is shown in Figure 7.

Once the algorithm finds the proper pitch progression that fulfils the harmony fixed by the Andalusian loop, a pitch shifting process has to be performed over the segments. The pitch shifting algorithm is based on a Sinusoidal plus Residual model [23] of the sound and includes a timbre preservation algorithm. This method can be applied, because the input is a singing voice signal, and this kind of signal can be modelled as a summation of sinusoids plus a non-harmonic part named residual [23]. Formally,

$$s(t) = \sum_{r=1}^{R} A_r(t)\cos[\theta_r(t)] + e(t) \tag{3}$$

where $s(t)$ is the input signal, R is the number of sinusoids that model the signal, $A_r(t)$ and θ_r are the instantaneous amplitude and phase of the r-th sinusoid, respectively, and $e(t)$ is the residual component at time t.

A harmony sinusoidal analysis is performed, which extracts the harmonic spectral peaks from the spectrum at multiples of the estimated fundamental frequency. The residual spectrum is obtained by subtracting the sine spectrum from the original spectrum. In order to preserve the original timbre, information is extracted from the spectral peaks, in order to estimate the spectral envelope. Then, the peaks are properly shifted to change the original pitch to the desired one (according to the harmony adaptation). Finally, the original envelope is applied to the shifted spectral peaks. The new residual spectrum is modelled by a shaped stochastic signal with the same spectral envelope as the original residual signal subtracted from the original spectrum. In this case, the residual can be described as filtered white noise, 4.

$$e(t) = \int_0^t h(t,\tau)u(\tau)d\tau \tag{4}$$

where $e(t)$ is the modelled residual, $h(t,\tau)$ is the response of a time varying filter to an impulse at time t, and $u(\tau)$ is white noise. In other words, the residual is modelled by the time-domain convolution of white noise with a time-varying frequency-shaping filter equivalent to the spectral shape of the non-harmonic part of the input signal.

The sinusoidal spectrum is re-synthesized from the shifted peaks information and added to the modelled residual spectrum. The transformed signal is then reconstructed using the IFFT, windowing the resulting signal by a triangular window and finally using the usual overlap-add method [23].

After the rhythm organization and harmony adaptation process, the input signal is already converted into a flamenco signal. In order to complete the cover generation, a guitar accompaniment is added to the track. The tempo of this accompaniment has to be adapted.

As mentioned above, the input melody is adapted to fulfil the chord progression Am-G-F-E (Andalusian cadence). In contrast, the guitar accompaniment does not require any harmony processing since it was already recorded according to this progression.

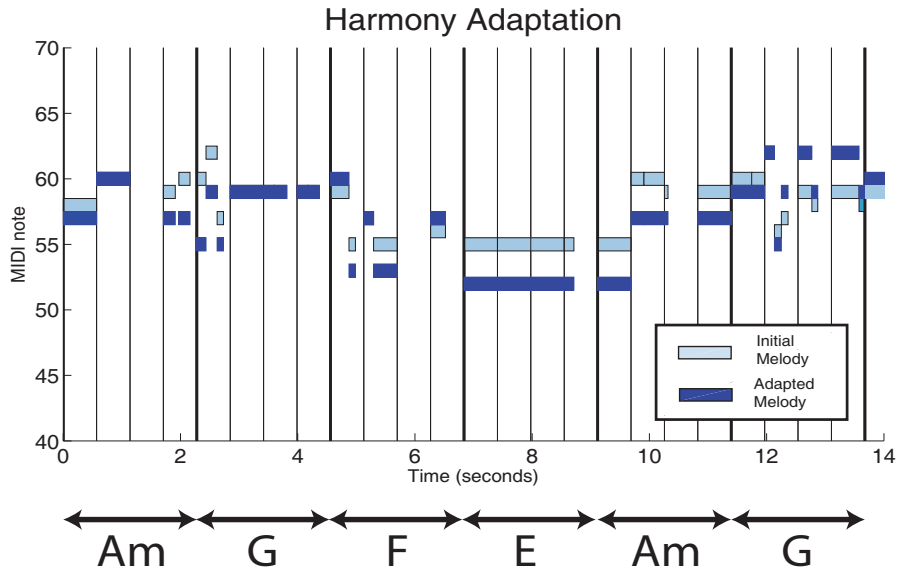

Figure 7: *Harmony adaptation process. In fair blue the original melody, in dark blue the adapted melody keeping the original contour.*

6.3. Accompaniment Tempo Adjustment

The objective of this process is to adjust the tempo of the accompaniment to that of the input signal. To this end, a time stretching process has to be applied to the accompaniment signals. The algorithm selected for time scaling [23] is a frame based frequency domain technique based on the Spectral Modelling Synthesis (SMS model [24]). The output spectral frames are computed by the interpolation of both sinusoidal and residual components separately frame-by-frame.

The synthesis hop size is kept constant, such that if the stretching factor is set to slow down the sound, new frames will be generated. Hence, the synthesis time reference will advance at a constant rate defined by the hop size, while the pointer to the analysis time will advance according to the stretching factor. The procedure is the following:

1. Advance the analysis time pointer according to stretching factor.
2. Perform a weighted interpolation using the previous and the next frame, according to the temporal distance from the analysis pointer to the central time of each of these two frames.
3. Add the interpolated frames to the synthesized signal using the synthesis time as its centre time.
4. Add hop size samples to update the synthesis time pointer.

By computing all frames in this way, the accompaniment tempo is adjusted to the input data rumba version.

Now, the last step is mixing the accompaniment and the vocals. In order to avoid undesired masking, the gain of accompaniment channel is automatically adjusted to ensure that the vocal channel is 3 dB over the accompaniment.

6.4. Score Sheet Generation

The final step consist of the creation of the score sheet. In order to achieve this, an automatic process based on Lilypond software [25] which can create a document containing the score in a programmatic way. The data used for the creation of the score is the pitch target file obtained in the Harmony Adaptation module (Section 6.2), since it contains the final version of the rhythm and pitch.

Considering that the pitch target file does not necessary contain non-quantized figures (i.e. incomplete durations such as quarter notes, eighth notes, ...), the graphic version of the flamenco rumba generated will try to transcribe the transformed rhythmic structure by quantizing the notes.

In this way, in addition to an audio sample of the flamenco rumba cover version of what the user has sung, he/she also obtains a score sheet with the transformed melody. The user is thus enabled to re-sing the rumba transformation or play it with an instrument.

7. CONCLUSIONS AND FURTHER DEVELOPMENT

A scheme for the automatic generation of rumba cover versions has been presented. A novel way to generate new audio material, based on a set of basic transformation applied at the note level, has been proposed. Furthermore, with this paper we also want to promote this music style that belongs to the Spanish heritage, by the presenting its unique characteristics in an appealing way.

In order to automatize the style transformation, some innovative algorithms, adapted to the restrictions of the input, have been implemented. Concerning the tempo estimation, since the input signal is expected to be an acapella signal without a strictly stable tempo (i.e. with short deviations in the tempo), an algorithm designed for the analysis of acapella songs is implemented. The main idea was to estimate the tempo by means of the study of the inter onset interval durations. The information about the IOI is provided by a transcriptor that turns the input audio signal into segments

with time and pitch information. Considering that the transcriptor is not an ideal process, some onset deviations can cause a variation in the tempo estimation.

The symbolic representation of the input melody is also used for obtaining the rhythm structure and pitch contour information from the input signal. This information is modified in order to fulfil the rhythmic and harmonic constraints of the flamenco rumba style. The rhythmic adaptation is based on the position and the duration of each note, so they are arranged to coincide in the downbeats of the rumba rhythm structure (Figure 1) and also fall together with the downbeats of the accompaniment. After this process, the duration of the segments could be affected, so a time stretching process is applied.

The harmonic adaptation of the melody is done by a computational implementation of the music theory method called harmonic adaptation by pitch level change [16].

The estimated tempo is also used to adapt the tempo of prerecorded guitar accompaniment to the transformed input melody.

Alternative implementations of the transformation algorithms could be considered to improve the system performance or to generate different types of audio content. Although the system performs properly with any type of voiced input signals, the best results (from a subjective and musical point of view) are obtained when these signals correspond to singing voice[2].

8. REFERENCES

[1] O. Mayor, J. Bonada, and J. Janer, "Kaleivoicecope: Voice transformation from interactive installations to videogames," *in Proceedings of AES 35th International Conference: Audio for Games*, February 2009.

[2] O. Mayor, J. Bonada, and J. Janer, "Audio transformation technologies applied to video games," *in Proceedings of AES 41st International Conference: Audio for Games*, February 2011.

[3] Smule, "Songify: Turn speech into music," *Website, http://www.smule.com/songify/index*, 2012.

[4] J.L. Flanagan, D.I.S. Minhart, R.M. Golden, and M.M. Sondhi, "Phase vocoder," *Journal of Acoustic Society of America*, vol. 38, no. 5, pp. 939–940, 1965.

[5] L. Fernandez, *Flamenco music theory: rhythm, harmony, melody and form*, Mel Bay Publications, 2005.

[6] E. Gómez, G. Peterschmitt, X. Amatriain, and P. Herrera, "Content-based melodic transformations of audio meterial of a music processing application," *Proc. of the 6th Int. Conference on Digital Audio effects (DAFX-03)*, September 2003.

[7] J.W. Downling, *Blackwell Handbook of Perception*, chapter Music Perception, Blackwell, 2001.

[8] B. Lawlor, "A novel efficient algorithm for music transposition," *in Proceddings of 25th EUROMICRO Conference*, vol. 2, pp. 45–54, 1999.

[9] Celemony, "Melodyne editor," *Website, http://www.celemony.com*, 2001.

[10] E. Molina, "Automatic scoring of singing voice based on melodic similarity measures," M.S. thesis, Universitat Pompeu Fabra, 2012.

[11] M.P. Ryynänen, A.P. Klapuri, P.O. Box, and F. Tampere, "Modelling of note events for singing transcription," *in Proceedings of ISCA Tutorial and Research Workshop on Statistical and Perceptual*, 2004.

[12] R.J. McNab, L.A. Smith, and I.H. Witten, "Signal processing for melody transcription," *in Proceedings of the 19th Australasian Computer Science Conference*, vol. 4, no. 18, pp. 301–307, 1996.

[13] C. Uhle and J. Herre, "Estimation of tempo, micro time and time signature from percussive music," in *Proceedings of Digital Audio Effects Workshop 2003 (DAFx 2003)*, 2003.

[14] F. Gouyon, P. Herrera, and P. Cano, "Pulse-dependent analyses of percussive music," in *Proceedings of ICASSP 2002*, 2002, vol. 4.

[15] B. Benward, *Music: In Theory and Practice*, vol. 1, McGraw-Hill Companies, 7th edition, 2003.

[16] D. Roca and E. Molina, *Vademecum musical*, Instituto de Eduación Musical, 2006.

[17] R. Groves, "Melody-to-harmony correction based on simplified counterpoint," *in Proceedings of ISMIR 2011*, 2011.

[18] A. Cheveigne and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *Journal Acoustical Society of America*, vol. 4, no. 111, pp. 1917–1930, 2002.

[19] W. J. Riley, *Handbook of Frequency Stability Analysis*, National Institute of Standars and Technology, 2007.

[20] J. Bednar and T. Watt, "Alpha-trimmed means and their relationship to median filters," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 32, no. 1, pp. 145–153, 1984.

[21] J. London, *Hearing in Time: Psychological Aspects of Musical Meter*, Oxford University Press, 2004.

[22] F. Lerdahl and R. Jackendoff, *A generative theory of tonal music*, MIT Press, Cambridge, MA, 1983.

[23] X. Amatriain, J. Bonada, A. Loscos, and X. Serra, *Spectral Processing*, chapter DAFX: Digital Audio Effects, John Wiley & Sons Plublishers, 2008.

[24] X. Serra and J. Smith, "Spectral modeling synthesis: A sound analysis/synthesis based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, pp. 12–24, 1990.

[25] H. W. Nienhuys and J. Nieuwenhuizen, "Lilypond, a system for automated music engraving," *in Proceedings of the XIV Colloquium on Musical Informatics (CIM 2003)*, 2003.

[2] Audio samples of the performance of the presented system can be found at http://www.atic.uma.es/Rumbator