ESTIMATION OF THE DIRECTION OF STROKES AND ARPEGGIOS

Isabel Barbancho[1], George Tzanetakis[2], Lorenzo J. Tardón[1], Peter F. Driessen[2], Ana M. Barbancho[1]
[1]Universidad de Málaga, ATIC Research Group, ETSI Telecomunicación,
Dpt. Ingeniería de Comunicaciones, 29071 Málaga, Spain
[2] University of Victoria, Department of Computer Science, Victoria, Canada
ibp@ic.uma.es, gtzan@cs.uvic.ca, lorenzo@ic.uma.es,
peter@ece.uvic.ca, abp@ic.uma.es

ABSTRACT

Whenever a chord is played in a musical instrument, the notes are not commonly played at the same time. Actually, in some instruments, it is impossible to trigger multiple notes simultaneously. In others, the player can consciously select the order of the sequence of notes to play to create a chord. In either case, the notes in the chord can be played very fast, and they can be played from the lowest to the highest pitch note (upstroke) or from the highest to the lowest pitch note (downstroke).

In this paper, we describe a system to automatically estimate the direction of strokes and arpeggios from audio recordings. The proposed system is based on the analysis of the spectrogram to identify meaningful changes. In addition to the estimation of the up or down stroke direction, the proposed method provides information about the number of notes that constitute the chord, as well as the chord playing speed. The system has been tested with four different instruments: guitar, piano, autoharp and organ.

1. INTRODUCTION

The design and development of music transcription systems has been an open research topic since the first attempts made by Moorer in 1977 [15]. Since then, many authors have worked in different aspects of the transcription problem [12], [17]. A common task in this context is automatic chord transcription [13], [1], [3], [7], [14], but also, other aspects beyond the mere detection of the notes played are nowadays considered, shifting the focus of the research to different pieces of information related to the way in which these notes are played, i.e. musical expressiveness [18], [4], [7], [11].

A chord can be defined as a specific set of notes that sound at the same time. Often, when a chord is played, not all the notes in the chord start at the same time. Because of the mechanics of actuation of some instruments like the guitar, the mandolin, and the autoharp [20], it is hard to excite different strings at the same time. Instead the performer typically actuates them sequentially in a stroke. A stroke is a single motion across the strings of the instrument. The stroke can have two different directions: UP, when the hand moves from the lowest to the highest note, and DOWN, when the hand moves from the highest to the lowest note. A strum is made up of several strokes combined in a rhythmic pattern. In other instruments like the piano or the organ, all the notes that belong to a certain chord can be played at the same time. However, the musician can still choose to play the chord in arpeggio mode, i.e., one note after another. Again, the arpeggio direction can be up or down.

In this paper, we propose a new chord related analysis task focused on the identification of the stroke or arpeggio direction (up or down) in chords. Because the movement can be fast it is not feasible to look for onsets [6] to detect each note individually. Therefore, a different approach will be proposed. In addition to the detection of the stroke direction, our proposed method also detects the speed with which the chord has been played as well as the number of notes. The estimation of the number of notes played in a chord is a problem that has not been typically addressed, although some references can be found related to the estimation of the number of instruments in polyphonic music [16], which constitutes a related but different problem. Regarding the chord playing speed, to the best our knowledge there are no published works to identify this parameter except when specific hardware is used for the task [19], [9]. The paper is organized as follows: in Section 2, the proposed system model is explained. Section 3 presents some experimental results and Section 4 draws some conclusions.

2. STROKE AND ARPEGGIO ANALYSIS

The main goal of this work is the analysis of audio excerpts to detect if a chord has been played from lower to higher notes (UP) or vice versa (DOWN). The movement to play a chord may be quite fast and all the information about the movement is contained at the very beginning of the chord waveform. After all the strings of the chord have been played, it is no longer possible to know whether the

© Isabel Barbancho[1], George Tzanetakis[2], Lorenzo J. Tardón[1], Peter F. Driessen[2], Ana M. Barbancho[1].
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Isabel Barbancho[1], George Tzanetakis[2], Lorenzo J. Tardón[1], Peter F. Driessen[2], Ana M. Barbancho[1]. "ESTIMATION OF THE DIRECTION OF STROKES AND ARPEGGIOS", 15th International Society for Music Information Retrieval Conference, 2014.

movement was up or down because the resulting sound contains all the component pitches. This means that any feature that may provide information about how the spectrum varies when the chord is being played has to be calculated at the very beginning of the chord. We will consider that the time needed to complete a stroke varies from 1 s (relatively slow) to less than 0.2 s, when the chord is played fast.

Let x denote the samples of the played chord under study. In order to calculate a spectrogram, the samples x are divided into segments $x_m = [x_m[1], ..., x_m[M]]^T$, where M is the selected window size for the spectrogram calculation. Let PSD_m denote the Power Spectral Density of each segment x_m and L_m the logarithm of the PSD_m i.e $L_m = 10\log_{10}(PSD_m)$. In Fig. 1, the log spectrogram of an 'F Major' guitar chord played from the lowest to the highest string is shown (up stroke). The exact fret position employed to play this chord is $frets = [2, 2, 3, 4, 4, 2]$ where the vector $frets$ represents the frets pressed to play the chord from string 1 (highest string) to string 6 (lowest string). This chord has been generated synthetically to control the exact delay between each note in the chord (in this case the delay is $\tau = 4ms$). The guitar samples have been extracted from the RWC database [10]. As it can be observed in Fig. 1, the information about the stroke direction is not directly visible in the spectrogram. Therefore, in order to detect the stroke direction, the spectrogram needs to be further analysed.

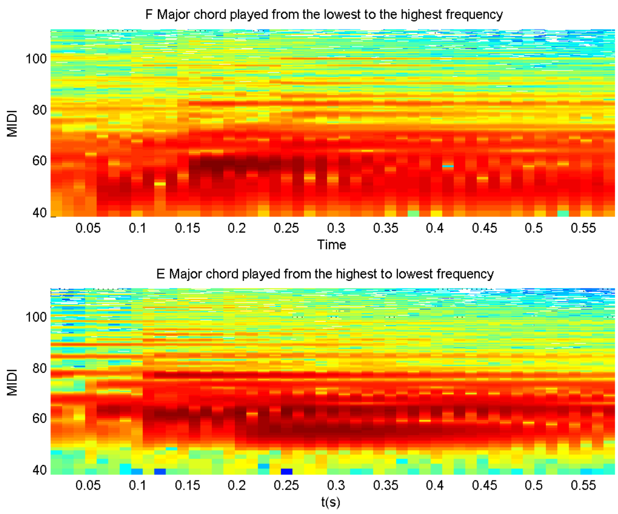

Figure 1. Spectrogram of an F Major chord UP stroke in a classic guitar (upper figure) and an E Major chord DOWN stroke. Audio file is sampled with $f_s = 44100$ Hz. For the spectrogram, the selected parameters are window size $M = 1024$, $overlapp = 512$ with a Hamming window. The DFT size is $K = 4096$. For convenience, the MIDI numbers are shown in the y-axis instead of the frequency bins: MIDI $= 69 + 12\log_2(f/440)$.

2.1 Detection of new spectral components

Whenever a new note is played, it is expected that new spectral components corresponding to the new note will be added to the existing components of the previous note (if any). In auditory scene analysis [8] this is termed the 'old+new heuristic'. The main idea is to take advantage of this heuristic by detecting whether the current spectrum contains new components or, conversely, whether it simply retains the components from the previous spectrum. As we are frequently dealing with sounds that decay quickly our model of sustained notes will also contain a decay component. In order to detect 'old+new' changes we minimize the following equation:

$$\epsilon[m] = \min_{\alpha[m]} \left[\sum_{k=1}^{K} |L_m[k] - \alpha[m] L_{m-1}[k]| \right] \quad (1)$$

The goal is to find a local $\alpha[m]$ (decay factor) that minimizes $\epsilon[m]$ for two consecutive windows m and $m-1$. The minimization is carried out by means of the unconstrained nonlinear minimization Nelder-Mead method [21]. The idea is to remove from window m all the spectral components that were also present in window $m-1$ with a gain adjustment so that any new spectral component becomes more clearly visible. Thus, if there are no new played notes in window m with respect to window $m-1$, $\epsilon[m]$ will be small, otherwise $\epsilon[m]$ will become larger because of the presence of the new note.

In Fig. 2 (a) and (b), the normalized evolutions of $\alpha[m]$ and $\epsilon[m]$ respectively are displayed for the F Major UP chord shown in Fig.1 (a). The vertical lines represent the instants when new notes appear in the chord. When a new note is played in the chord, $\alpha[m]$ attains a minimum and $\epsilon[m]$ a maximum. In order to automatically detect the instants when the new notes appear, the following variables are defined:

$$\epsilon'[m] = \begin{cases} \epsilon[m] - \epsilon[m-1] & \text{if } \epsilon[m] - \epsilon[m-1] > 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$\alpha'[m] = \alpha[m] - \alpha[m-1] \quad (3)$$

$$f_{\alpha,\epsilon}[m] = \frac{\epsilon'[m]}{max(\epsilon')} \cdot \left| \frac{\alpha'[m]}{max(\alpha')} \right| \quad (4)$$

Fig. 2 (c) shows the behaviour of $f_{\alpha,\epsilon}$, where if becomes easy to identify the presence of new notes. In addition, it is also possible to estimate the number of notes played in the chord (in this case 6), as well as the stroke speed.

2.2 Estimation of number of notes and stroke speed

After a measure that highlights the presence of new notes has been defined, the next step is to find the peaks of $f_{\alpha,\epsilon}$. Each sample of the function $f_{\alpha,\epsilon}(m)$ is compared against $f_{\alpha,\epsilon}(m-1)$ and $f_{\alpha,\epsilon}(m+1)$. If $f_{\alpha,\epsilon}(m)$ is larger than both neighbors (local maximum) and $f_{\alpha,\epsilon}(m) > 0.1$, then a candidate local peak is detected. Finally, if there are two

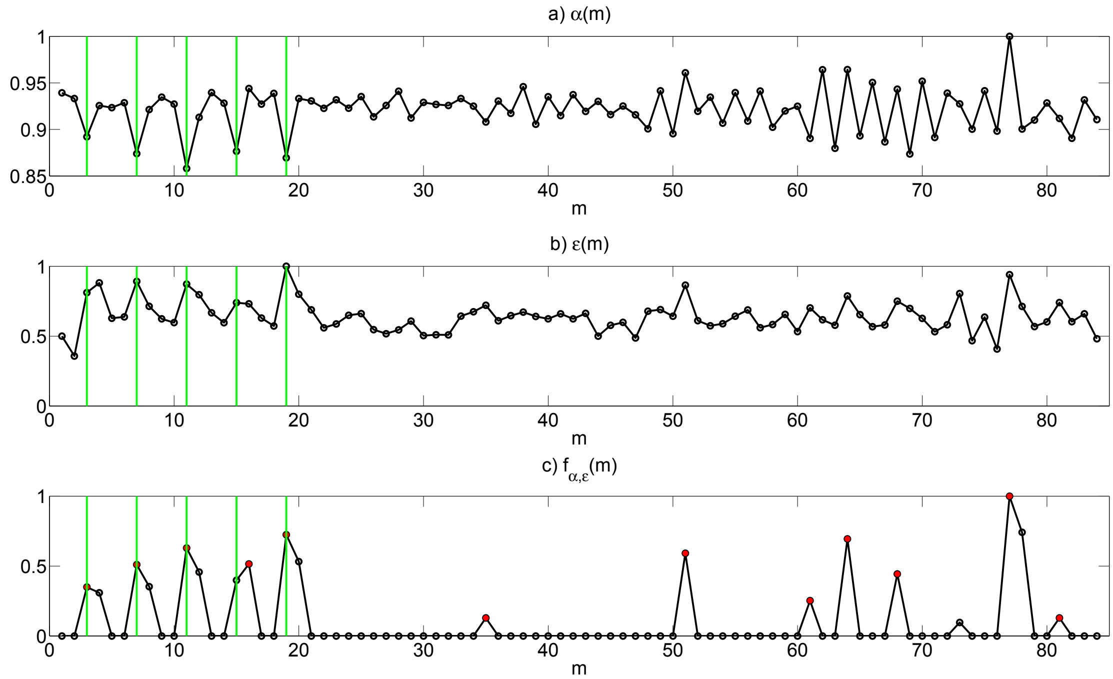

Figure 2. F Major chord UP stroke in classic guitar: (a) Evolution of $\alpha[m]$ that minimizes equation (1) , (b) Evolution of the error $\epsilon[m]$ as defined in equation (1) and (c) Evolution of $f_{\alpha,\epsilon}[m]$ (equation (4)) where the presence of new notes becomes apparent.

peaks less than two points apart, the smallest one is not considered. Once these selected peaks have been localized, the final step is to determine which ones belong to played notes so that the number of played notes can be estimated together with the speed of the stroke. The key observation is that the time difference between the note onsets that belong to the same stroke or arpeggio will be approximately constant. The reason is that, because of human physiology, in most cases the stroke is performed with fixed speed.

Let f_{locs} stand for a function that contains the positions where the selected peaks of $f_{\alpha,\epsilon}$ are located. The objective is to detect sets of approximately equispaced peaks which will correspond to the played notes in a chord or arpeggio. Then, the number of played notes NPN_e will be estimated as follows:

$$\mathrm{NPN}_e = n_{neq} + 2 \tag{5}$$

where n_{neq} represents the minimum value of n such that the distance between the positions of peaks contained in f_{locs} is no longer kept approximately constant. n_{neq} is defined as:

$$n_{neq} = \underset{n}{\mathrm{argmin}} \left(|f''_{locs}(n)| > 3 \right) \tag{6}$$

where $f''_{locs}(n)$ stands for the second order difference of $f_{locs}(n)$.

Finally, the stroke speed estimate in notes per second is given by:

$$V = \frac{f_{locs}(\mathrm{NPN}_e - 3) \cdot (windowsize - overlapp)}{f_s \cdot \mathrm{NPN}} \tag{7}$$

Once the location of every new note is estimated using the method described, the feature to detected the stroke direction is computed.

2.3 Feature to detect stroke direction

In Fig. 3, the details of the windows in which the spectral changes occur are depicted for the two guitar chords that are being analysed. The stroke direction can be guessed from those figures, but we still need to derive a meaningful computational feature that can be used for automatic classification.

In order to reduce the amount of information to be processed by the classifier that will decide the stroke direction, a meaningful feature must be considered. Thus, the spectral centroid in each of the windows in which the changes have been detected is calculated.

The spectral centroid is the centre of gravity of the spectrum itself [22], [24] and, in our case, it is estimated in each of the windows x_m where the change has been detected. This feature will be denoted SPC_m (Spectral Centroid of window m) and it is calculated as follows:

$$SPC_m = \left(\sum_{k=1}^{K} f_m(k) PSD_m(k) \middle/ \sum_{k=1}^{K} PSD_m(k) \right) \tag{8}$$

where PSD_m is the power spectral density of the window x_m and f_m is the corresponding frequency vector.

Note that we will use SPCs-KD when the SPCs are estimated with the delays known beforehand and SPCs-ED when the delays are estimated according to the procedure described in section 2.1.

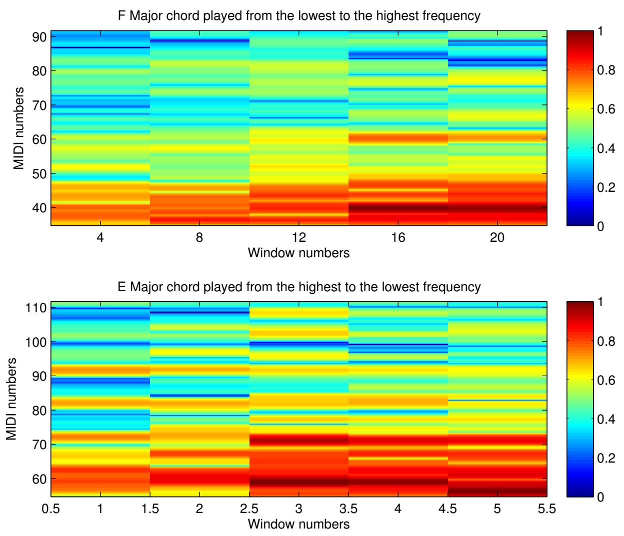

Figure 3. Windows of the spectrogram of the UP F Major chord and the DOWN E Major chord in which new notes appear.

Fig. 4 illustrates the behaviour of the SPC in the selected windows in which a change of the spectral content is detected for the UP F Major chord and the DOWN E Major chord shown in the previous illustrations.

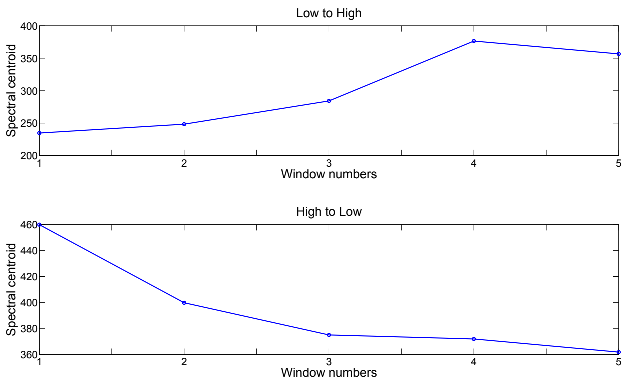

Figure 4. Spectral centroid evolution for the UP F Major chord and the DOWN E Major chord in the windows of the spectrogram in which the changes happen.

3. CLASSIFICATION RESULTS OF UP AND DOWN STROKES

The proposed scheme has been tested with four different instruments: guitar, piano, organ and autoharp. The guitar and organ samples have been extracted from the RWC database [10], the piano recordings have been extracted from [2] and the autoharp recordings have been specifically made by the research team.

A subset of the chords used in the experiment contains chords artificially assembled so that all the information regarding the number of notes played and the delay is available for assessing the proposed system performance. All audio file are sampled with $f_s = 44100$ Hz. The delay between consecutive notes in a chord ranges between 1000 samples (11 ms) and 5000 samples (55 ms).

With the guitar and the autoharp, the natural way of playing a chord is by adding the sound of one string after another. The guitar is a well known instrument [5], but the autoharp is not. The autoharp is an American instrument invented in 1881 by Charles F. Zimmerman. It was very popular in Canada and the USA for teaching music fundamentals because it is easy to play and introduces in a very intuitive way harmony concepts. Briefly, the instrument has 36 strings and the musician can select which ones can vibrate by pressing buttons corresponding to different chords. The buttons in the autoharp mute the strings corresponding to the notes that do not belong to the chord to be played. Then, the musician actuates the strings by strumming with the other hand. In the guitar, the decay of each string is exponential and very fast. In the case of the autoharp, due to the resonance box, the decay of the sound is slower. In the piano and in the organ, the musician can play the chords arpeggiated. In the piano the decay is also exponential but in the organ the sound of a note is sustained and decays slowly.

In Tables 1 and 1, the UP and DOWN classification results are summarized for the artificially assembled chords. In all the cases, 100 chords have been used for training (50 UP and 50 DOWN) and a total of 500 chords equally distributed among UP and DOWN have been used to evaluate the classification performance. The chord recordings used for training and testing purposes are separate and different.

The performance of the proposed feature is compared against a baseline that makes use of MFCCs (Mel Frequency Cepstral Coefficients) calculated as explained in [22]. More specifically, 15 coefficients are considered with the first one, corresponding to the DC component, removed.

A Fisher Linear Discriminant and a linear Support Vector Machine (SVM) classifier [23] have been evaluated.

Looking at Tables 1 and 2, we observe that the results of the proposed method and feature are satisfactory. In almost all the cases, the performance of the proposed scheme is better than the one achieved by the baseline based on MFCCs.

The error in the determination of the number of played notes is estimated as follows:

$$Error_{NPN} = A\left(\frac{|NPN_e - NPN_r|}{NPN_r}\right) \cdot 100 \quad (9)$$

where $A()$ is the averaging operator, NPN_e stands for the estimated Number of Played Notes in (5) and NPN$_r$ represents the the actual number of notes.

The error in the estimated delay between consecutive notes is evaluated as follows:

$$Error_{W} = A\left(\frac{|W_e - W_r|}{NPN_e \cdot W_d}\right) \cdot 100 \quad (10)$$

where W_e represents the windows in which a significant spectral change has been found, W_r stands for the windows

Instrument	stroke	Fisher		
		SPCs-KD	SPCs-ED	MFCCs
Guitar	up	93.88	78.88	72.22
	down	96.11	97.22	60.55
	overall	95.00	88.05	66.38
Piano	up	91.95	79.31	77.85
	down	97.81	84.36	81.42
	overall	94.88	81.83	79.64
Organ	up	90.00	89.16	78.33
	down	90.00	86.66	56.66
	overall	90.00	87.91	67.50
Autoharp	up	100	94.44	97.91
	down	100	86.80	79.86
	overall	100	90.62	88.88

Table 1. Success Rate (%) of UP and DOWN stroke classification using a Fisher linear classifier [23]. The features used by the classifier are: SPCs-KD (Spectral Centroid of selected Windows with known-delay), SPCs-ED (Spectral Centroid of selected Windows with estimated delay) and MFCCs (15 Mel Frequency Cepstral Coefficients).

Instrument	stroke	SVM		
		SPCs-KD	SPCs-ED	MFCCs
Guitar	up	91.57	87.77	58.44
	down	95.01	95.55	98.78
	overall	93.29	91.66	78.61
Piano	up	90.12	81.22	77.25
	down	96.45	82.84	83.63
	overall	93.28	82.03	80.44
Organ	up	89.16	90.52	90.83
	down	88.66	87.98	51.66
	overall	88.91	89.25	71.25
Autoharp	up	99.30	90.97	91.27
	down	97.91	95.14	90.89
	overall	98.61	93.05	91.08

Table 2. Success Rate (%) of UP and DOWN stroke classification using a linear SVM classifier [23]. The features used by the classifier are: SPCs-KD (Spectral Centroid of selected Windows with known-delay), SPCs-ED (Spectral Centroid of selected Windows with estimated delay) and MFCCs (15 Mel Frequency Cepstral Coefficients).

where the changes actually happen and W_d is number of windows between two consecutive W_r windows. Table 3 shows the obtained results.

The proposed method for the estimation of the number of notes and delays can be improved. This is a first approach to solve this problem. Our main goal has been to detect the up or down stroke direction which is useful to complete the transcription of the performance of certain instruments, specifically the autoharp. The performance attained in the detection of the stroke direction is satisfactory according to the results shown.

It is important to note, that even though $Error_W$ seems to be quite high, this error is in most of cases positive, i.e., the change is detected one or two windows after the first window that actually contains the change. This issue is not critical for the feature employed by the classifier because it is possible to observe the difference in the estimation of the SPC_m in (8).

Instrument	stroke	$Error_{NPN}$	$Error_W$
Guitar	up	37.65	10.49
	down	33.33	15.92
	overall	35.49	13.20
Piano	up	30.72	28.38
	down	33.65	18.10
	overall	32.18	23.24
Organ	up	24.54	29.72
	down	36.52	26.12
	overall	30.53	27.92
Autoharp	up	53.06	10.46
	down	42.88	13.96
	overall	47.97	12.21

Table 3. Error (%) in the estimation of the number of notes played and in the estimation of the delay between consecutive played notes in chords.

Instrument	stroke	Fisher	
		SPCs-ED	MFCCs
Autoharp	up	65.21	43.47
	down	86.44	94.91
	overall	75.10	69.19
		SVM	
		SPCs-ED	MFCCs
	up	73.77	62.84
	down	89.83	81.52
	overall	77.52	72.18

Table 4. Success Rate (%) of UP and DOWN stroke classification for real autoharp chords.

Finally, Table 4 presents the results obtained for real chords played in an autoharp. We have used 100 chords for training and 230 chords for testing. The 330 autoharp chords recorded are equaly distributed between UP and DOWN chords and in different tessituras. It can be observed that the proposed feature outperforms the baseline proposed based on the usage of MFCCs.

4. CONCLUSIONS

In this paper, a new feature to detect the up or down direction of strokes and arpeggios has been presented. The developed method also provides information about the number of played notes and the stroke speed.

The system have been tested with four different instruments: classic guitar, piano, autoharp and organ and it has been shown how the new proposed feature outperforms the baseline defined for this task. The baseline makes use of the well known MFCCs as classification features so that the baseline scheme can be easily replicated. The Matlab files used to generate the data-set for piano, guitar and organ, the audio files of the autoharp and the ground truth are available upon request for reproducible research.

5. ACKNOWLEDGMENTS

This work has been funded by the Ministerio de Economía y Competitividad of the Spanish Government under Project No. TIN2013-47276-C6-2-R, by the Junta de Andalucía under Project No. P11-TIC-7154 and by the Ministerio de Educación, Cultura y Deporte through the Programa Nacional de Movilidad de Recursos Humanos del Plan Nacional de I-D+i 2008- 2011. Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech.

6. REFERENCES

[1] A. M. Barbancho, I. Barbancho, B. Soto, and L.J. Tardón. Transcription of piano recordings. *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 377–380, 2011.

[2] A. M. Barbancho, I. Barbancho, L. J. Tardón, and E. Molina. *Database of Piano Chords. An Engineering View of Harmony*. SpringerBriefs in Electrical and Computer Engineering, 2013.

[3] A. M. Barbancho, A. Klapuri, L.J. Tardón, and I. Barbancho. Automatic transcription of guitar chords and fingering from audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 20:915–921, March 2012.

[4] I. Barbancho, C. de la Bandera, A. M. Barbancho, and L. J. Tardón. Transcription and expressiveness detection system for violin music. *Proceedings of the IEEE conference on Acoustics, Speech, and Signal Proc. (ICASSP)*, pages 189–192, March 2009.

[5] I. Barbancho, L.J Tardon, S. Sammartino, and A.M. Barbancho. Inharmonicity-based method for the automatic generation of guitar tablature. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(6):1857–1868, 2012.

[6] J.P. Bello, L. Daudet, and M.B. Sandler. A tutorial on onset detection in music signals. *IEEE Trans. on Audio, Speech and Language Processing*, 14:1035–1047, September 2005.

[7] E. Benetos, S. Dixon, D. Giannoulis, and H. Kirchhoff. Automatic music transcription: Breaking the glass ceiling. *ISMIR*, 2012.

[8] Albert S Bregman. *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994.

[9] D. Chadefaux, J.-L. Le Carrou, B. Fabre, and L. Daudet. Experimentally based description of harp plucking. *The Journal of the Acoustical Society of America*, 131:844, 2012.

[10] M. Goto. Development of the RWC music database. In *18th Int. Con. on Acoustics*, volume I, pages 553–556, 2004.

[11] A. Kirke and E. R. Miranda. An overview of computer systems for expressive music performance. In *Guide to Computing for Expressive Music Performance*, pages 1–47. Springer, 2013.

[12] A. Klapuri and T. Virtanen. Automatic music transcription. In *Handbook of Signal Processing in Acoustics*, pages 277–303. Springer, 2009.

[13] A.P. Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Trans. on Speech and Audio Processing*, 11:804–816, Nov. 2003.

[14] K. Lee and M. Slaney. Acoustic chord transcription and key extraction from audio using key-dependent hmms trained on synthesized audio. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):291–301, 2008.

[15] J. A. Moorer. On the transcription of musical sound by computer. *Computer Music Journal*, pages 32–38, 1977.

[16] M.Schoeffler, F.R. Stoter, H.Bayerlein, B.Edler, and J.Herre. An experiment about estimating the number of instruments in polyphonic music: a comparison between internet and laboratory results. *ISMIR*, 2013.

[17] M. Müller, D. P. W. Ellis, A. Klapuri, and G. Richard. Signal processing for music analysis. *IEEE Journal of Selected Topics in Signal Processing*, 5:1088–1110, October 2011.

[18] T. H. Ozaslan, E Guaus, E. Palacios, and J. L. Arcos. Identifying attack articulations in classical guitar. In *Computer Music Modeling and Retrieval. Exploring Music Contents. Lecture Notes in Computer Science*, pages 219–241. Springer-Verlag, 2011.

[19] J.A. Paradiso, L.S Pardue, K.-Y. Hsiao, and A.Y. Benbasat. Electromagnetic tagging for electronic music interfaces. *Journal of New Music Research*, 32(4):395–409, 2003.

[20] M. Peterson. *Mel Bays Complete Method for Autoharp or Chromaharp*. Mel Bay Publications, 1979.

[21] W.H. Press, S. A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes: The Art of Scientific Computing. Third Edition*. Cambridge University Press, 2007.

[22] L.J. Tardón, S.Sammartino, and I.Barbancho. Design of an efficient music-speech discriminator. *Journal of the Acoustical Society of America*, 1:271–279, January 2010.

[23] S. Theodoridis and K. Koutroumbas. *Pattern Recognition, 4th Edition*. Academic Press, 2008.

[24] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Trans. on Audio, Speech and Language Processing*, 10:293–302, 2002.