Automatic Transcription of Polyphonic Piano Music Using Genetic Algorithms, Adaptive Spectral Envelope Modeling, and Dynamic Noise Level Estimation

Gustavo Reis, *Member, IEEE*, Francisco Fernandéz de Vega, *Senior Member, IEEE*, and Aníbal Ferreira, *Member, IEEE*

***Abstract*—This paper presents a new method for multiple fundamental frequency (F0) estimation on piano recordings. We propose a framework based on a genetic algorithm in order to analyze the overlapping overtones and search for the most likely F0 combination. The search process is aided by adaptive spectral envelope modeling and dynamic noise level estimation: while the noise is dynamically estimated, the spectral envelope of previously recorded piano samples (internal database) is adapted in order to best match the piano played on the input signals and aid the search process for the most likely combination of F0s. For comparison, several state-of-the-art algorithms were run across various musical pieces played by different pianos and then compared using three different metrics. The proposed algorithm ranked first place on Hybrid Decay/Sustain Score metric, which has better correlation with the human hearing perception and ranked second place on both onset-only and onset–offset metrics. A previous genetic algorithm approach is also included in the comparison to show how the proposed system brings significant improvements on both quality of the results and computing time.**

***Index Terms*—Acoustic signal analysis, automatic music transcription, fundamental frequency (F0) estimation, music information retrieval, pitch perception.**

I. Introduction

MULTIPLE fundamental frequency (F0) estimation was introduced by Shields [1] in his work on separating co-channel speech signals. Afterwards, the research of multiple-F0 estimation was extended to polyphonic pitch estimation in the context of automatic music transcription for polyphonic music signals by Moorer [2] and Piszcalski and Galler [3]. In general, multi-pitch estimation algorithms assume that there can be more than one harmonic source in the same short-time signal. As mentioned by Yeh *et al.* [4], that signal can also be expressed as a sum of harmonic sources plus a residual[1]:

$$y[n] = \sum_{m=1}^{M} y_m[n] + z[n], M > 0 \text{ with } y_m[n] \approx y_m[n + N_m] \tag{1}$$

where n is the discrete time index, M is the number of harmonic sources, $y_m[n]$ is the quasi-periodic part of the mth source, N_m represents the period of the mth source, and $z[n]$ is the residual. Using the Fourier Series, this model can be represented as follows:

$$\begin{aligned} y[n] &= \sum_{m=1}^{M} \left\{ \sum_{h=1}^{\infty} A_{m,h} \cos(h\omega_m n + \phi_{m,h}) \right\} + z[n] \\ &\approx \sum_{m=1}^{M} \sum_{h=1}^{H_m} A_{m,h} \cos(h\omega_m n + \phi_{m,h}) + z[n]. \end{aligned} \tag{2}$$

The approximation on the last step of this equation is for practical usage: a finite and small number of sinusoids H is commonly used to approximate a quasi-periodic signal.

The main problem behind multiple-F0 estimation is dealing with the modeling of $y[n]$, a task which implies estimating the number of sources and the related F0s. The decomposition of the observed signal into an unknown number of model sources is actually not only a problem of pattern matching but also a search problem, i.e., finding the most likely combination of F0s for the modeling of $y[n]$. Genetic algorithms are very successful in solving both pattern matching and search problems [5]. This is our main motivation in proposing a new multiple-F0 estimation algorithm based on genetic algorithms.

Since the first works in polyphonic music transcription by Moorer [2] and Piszcalski and Galler [3], polyphonic music transcription systems almost always rely on the analysis of information present in the frequency domain. Klapuri [6] uses iterative calculation of predominant fundamental frequencies in separate frequency bands and Martin [7] uses blackboard systems. There are also techniques that use the principles of human auditory organization for pitch analysis, as implemented

Manuscript received June 30, 2011; revised October 17, 2011, January 25, 2012, and April 26, 2012; accepted May 03, 2012. Date of publication May 25, 2012; date of current version August 13, 2012. This work was supported in part by the Spanish Ministry of Science and Innovation under Project ANYSELF (TIN2011-28627-C04), in part by Gobierno de Extremadura, under Projects GRU09105, GR10029, and in part by the Municipality of Almendralejo. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Engin Erzin.

G. Reis is with the Department of Computer Science, Polytechnic Institute of Leiria, 2401-951 Leiria, Portugal (e-mail: gustavo.reis@estg.ipleiria.pt).

F. Fernandéz de Vega is with the University of Extremadura, 06800 Badajoz, Spain (e-mail: fcofdez@unex.es).

A. Ferreira is with the Faculty of Engineering, FEUP-DEEC , University of Porto, 4200-465 Porto, Portugal (e-mail: ajf@fe.up.pt).

Digital Object Identifier 10.1109/TASL.2012.2201475

[1]The residual—$z(t)$—comes from components that are not explained by the sinusoids, for instance, the background noise, spurious components or non-harmonic partials.

by Kashino *et al.* [8] by means of a Bayesian probability network, where bottom-up signal analysis can be integrated with temporal and musical predictions, and by Wamsley *et al.* [9], [10], who use the Bayesian probabilistic framework to estimate the harmonic model parameters jointly for a certain number of frames. The usage of a hidden Markov model and spectral feature vectors was proposed by Raphael [11] to describe chord sequences in piano music signals. Neural Networks were used by Carreras *et al.* [12] for spectral-based harmonic decompositions of signals. Marolt [13] uses networks of adaptive oscillators to track partials over time. A physical model of the piano was used by Ortiz *et al.* [14] to generate spectral patterns in order to compare them to the incoming spectral data.

A. Genetic Algorithms and Multiple-F0 Estimation

Traditional approaches to multiple-F0 estimation rely on the analysis or decomposition of the source signal. However, genetic algorithm based approaches rely on the opposite: they focus on the construction of a second audio signal that best resembles the original audio.

In literature, the first paper using genetic algorithms for multiple-F0 appeared in 2001 by Garcia [15]. Garcia proposes that polyphonic pitch detection can be considered as a search space problem where the goal is to find the pitches that compose the polyphonic acoustic signal. Although Garcia's algorithm is able to detect several pitches in a short-time signal, it does not take in consideration the following aspects: onset time; offset time; and also the dynamics of each detected F0. On the other hand, this approach is capable of working with almost any frequency resolution. In 2007, Lu [16] proposed an automatic music transcription system based on genetic algorithms: this approach assumes that a polyphonic audio signal can only be produced by the 128 possible pitches (from the low C, frequency 8.18 Hz to a high G, 12543.88 Hz) defined in the MIDI specification [17]. This approach is limited to signals made by simple mathematical models like the sine, sawtooth and triangle waves. Also, it does not take into account the dynamics of each note. Although the author of this approach claims that he is addressing music transcription using genetic algorithms, his approach does not use recombination, which is the main pillar of genetic algorithms [5]. The approach relies exclusively on mutations.

In 2007, Reis and Fernandez [18] proposed a new genetic algorithm approach to automatic music transcription using synthesized instruments. However, this approach is not able to deal with multiple instruments as does Lu's [16] algorithm. Also, it cannot detect note dynamics. Later in 2007, Reis *et al.* [19] proposed the first genetic algorithm approach to music transcription using real audio recordings. The authors explored the influence of the "harmonic overfitting" phenomena, which is related to differences in the spectral envelope between different pianos. In 2008 Reis *et al.* [20] proposed a new algorithm with adaptive spectral envelope modeling to reduce the impact of the harmonic overfitting, and later, in 2009 [21], the latter approach was extended to multi-timbre.

Although the previously mentioned approaches already try to apply genetic algorithms to automatic music transcription, the proposed system described in this paper is a new and written from scratch genetic algorithm based on the knowledge of the authors from previous work on applying evolutionary algorithms to multi-pitch estimation [18]–[21]. The innovation in this system relies on dynamic noise level estimation and its subsequent combination with spectral envelope modeling in order to perform the transcription task. Also, to the best of our knowledge, this is the first time an evolutionary computing approach achieves competitive results when compared to other state-of-the-art algorithms. Moreover, computing time required to make the transcription is significantly reduced when compared to previous approaches.

The rest of this document is structured as follows. Section II overviews the proposed system. Section III describes the implemented onset detection algorithm. Section IV describes the proposed genetic algorithm. Section V describes the implemented hill-climber algorithm. Section VI shows our experiments and results. Finally, Section VII presents our conclusions and discusses future work.

II. System Overview

During the audio segmentation, an onset detector is applied on the input signal to extract onset information. Afterwards, the audio signal is divided into several audio segments according to the detected onsets. Each interval between two consecutive onsets is considered a segment. Then, for each segment, a *thread* is launched running a 50-generation genetic algorithm to perform the corresponding transcription. The search for the most-likely combination of F0s to model $y[n]$ is aided by an internal database consisting of previously recorded piano samples. The genetic algorithm also adapts the spectral envelope of the used piano samples in order to best match the power spectrum of the corresponding audio segment. During this process, spectral envelope of the residual noise is also dynamically estimated to favor the search process towards the desired solution (see Fig. 1). The results obtained on each audio segment are then merged into one whole transcription. Finally, a hill-climber algorithm [22] is applied on the global transcription to adjust note duration or to merge notes that transverse several segments. The output of the system is the hill-climber's final result.

Since the main focus of our work is the actual transcription problem and not the onset detection (and also because the lack of accuracy of the onset detector could compromise the performance of the algorithm) the user is able to choose other onset information as an input to the algorithm. The audio will then be segmented in accordance to the supplied information. This way, other onset detection systems that might be more robust can be used and, when dealing with labeled data, usage of the real onsets as input is also possible.

III. Onset Detection Algorithm

The onset detection algorithm is based on the onset detection algorithm used by Martins [23], with slight modifications. The approach used is based on the Spectral Flux as the onset detection function, defined as

$$SF(n) = \sum_{k=0}^{\frac{N}{2}} H\left(|X(n,k)| - |X(n-1,k)|\right) \qquad (3)$$

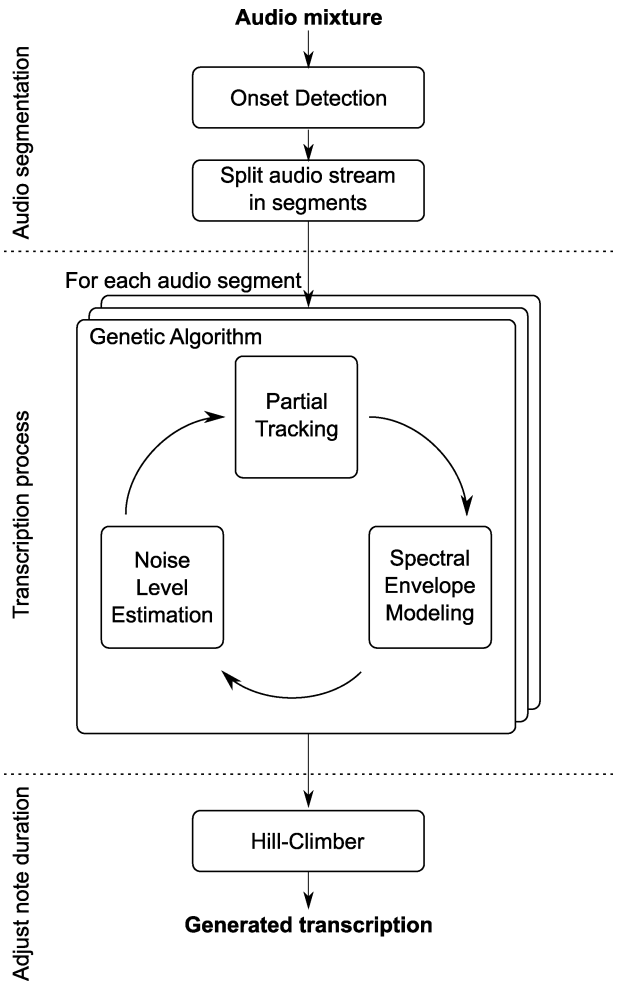


Fig. 1. Block diagram of the transcription algorithm.

where $H(x) = (x + |x|)/2$ is the half wave rectifier function, and $X(n,k)$ represents the kth bin of the nth frame of the short-time Fourier transform (STFT) of the input $x[n]$. Linear magnitude is used instead of logarithmic as in [24]. N is the Hamming window size. Experiments performed use a 46-ms frame size (i.e., $N = 2048$, with sampling rate $f_s = 44\,100$ Hz) and a 10-ms hop size (i.e., 441 samples with $f_s = 44\,100$ Hz).

As in [23], in order to reduce the false positive rate, the onset detection function $SF(n)$ is smoothed using a Butterworth filter defined by $H(z) = (0.1173 + 0.2347z^{-1} + 0.1174z^{-2})/(1 - 0.8252z^{-1} + 0.2946^{-2})$. To avoid the phase distortion (which would deviate the time of the detected onsets) the input data is filtered in both forward and backward directions. The result has a precisely zero phase distortion, being the magnitude the square magnitude of the filter response, and the order of the filter the double of the order specified by $H(z)$.

The onsets are detected using a peak-picking algorithm to find a local maximum. In fact, a peak at instant $t = nH/f_s$ is chosen as an onset if it meets the following criteria:

1) $SF(n) \geq SF(k)\ \forall\, k : n - w \leq k \leq n + m$;
2) $SF(n) > (\sum_{k=n-mw}^{n+w} SF(k)/(mw+w+1)) \times thres + \delta$.

where $w = 6$ is the window size to achieve the local maxima; $m = 4$ is a multiplier so that the average should be calculated in a broader area before the peak; $thres = 2.0$ is a threshold value relative to the local average that a peak must reach in order to be sufficient prominent to be selected as an onset; and $\delta = 10^{-20}$ a residual value to avoid false positive detection in silence regions of the signal. All these parameters were adjusted empirically on previously performed tests using a collection of several piano compositions played by different pianos.

Since the onsets are used as segmentation boundaries of the input signal, the occurrence of false negatives might have a considerable impact in the final results of the event segregation. The impact of the implemented onset detection system is studied on Section VII.

IV. Proposed Genetic Algorithm

It is important to emphasize that the main idea behind a genetic algorithm [5] is to have a set of candidate solutions (individuals) to a problem evolving towards the desired solution. In each iteration (generation) those candidate solutions are evaluated according to their quality (fitness). The worst solutions are then discarded and the best will generate new candidate solutions resulting from the combination of their parent's characteristics (genes) and minor variations (mutation). This way, candidate solutions with better quality tend to live longer and to generate better solutions, thus improving the robustness of the algorithm. Also, when addressing a genetic algorithm to a problem there are several aspects that must be taken into account:

Genotype	How to encode each individual or candidate solution to the problem.
Fitness Function	How to evaluate the quality of each candidate solution.
Selection	How individuals are selected from the population to breed.
Recombination	How to employ recombination: given two individuals, how to exchange genetic material between them to breed two new individuals (offspring).
Mutation	What kind of mutations we should take into account, according to the problem being solved.
Initialization	How the first population is generated.
Survivor Selection	How survivors are selected from one generation to the next.

A. Genotype

Since the problem being solved is the automatic transcription of an audio segment, a candidate solution must be a candidate transcription. We consider a transcription as a set of musical notes where each note has four attributes: start time, duration, MIDI note and also MIDI velocity. Therefore, an individual is encoded as a chromosome with a set of genes where each gene corresponds to a musical note [see Fig. 2(a)].

Despite the onset being fixed to its segment boundaries, the onset information needs to be included into the chromosome so the Hill-Climber can operate properly: after the transcription of each onset-synchronous segment, those transcriptions

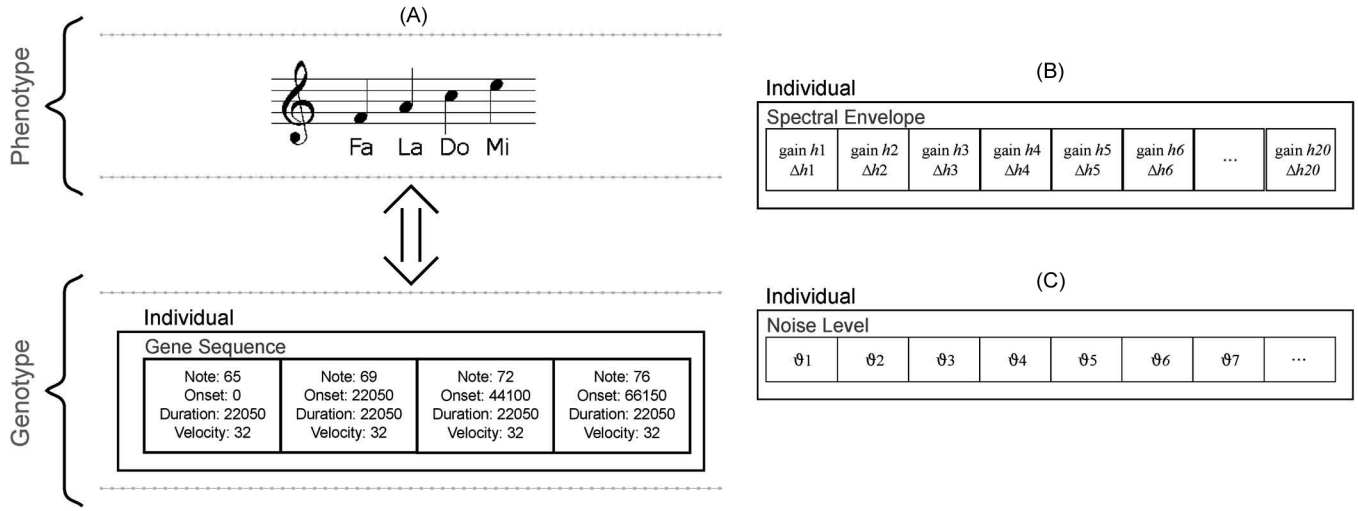


Fig. 2. (a) Genotype of an individual and corresponding phenotype. (b) Spectral envelope encoding. Each gene corresponds to a pair of values: the gain of the respective harmonic (expressed in dB) and its inharmonicity deviation. (c) The noise is encoded as an adaptive threshold below a maximum peak of the current time frame. Each gene corresponds to the noise threshold (expressed in dB) of the corresponding frequency bin.

are merged into one whole transcription (new individual). At this stage, the onset time differs from each gene. Moreover, the purpose of the Hill-Climber algorithm is to adjust the duration of notes and merge notes that overlap (transverse several audio segments). This process requires both duration and onset information on each gene.

In order to deal with dynamic noise level estimation and also spectral envelope modeling, additional chromosomes were included inside the genotype: one additional chromosome to encode the adaptive threshold used for the dynamic noise level estimation and one additional chromosome for each internal piano to adapt its spectral envelope to best match the piano played in the original audio.

1) Spectral Envelope Modeling: The search process for the most likely F0 combination matching the original signal is aided by previously recorded piano samples—internal synthesizer. It is almost certain that the piano used to record the original audio is not the same piano that is inside the genetic algorithm to aid the search process. Thus, when rendering a candidate solution using the internal synthesizer there will be several differences when compared to the original audio, specially on harmonic locations due to timbre differences between both instruments [see Fig. 3 (bottom left plot)]. In order to overcome those differences between the spectral envelope of both instruments, each individual or candidate solution has a second chromosome to encode the spectral envelope to be applied to the internal synthesizer so it can adapt to the original piano. This way, the internal piano can match the piano where the original audio was played.

The spectral envelope is encoded on the individual as a new chromosome [see Fig. 2(b)], where each gene corresponds to a pair of values: the gain for its harmonic—gain h_i—expressed in dB and its inharmonicity deviation—Δh_i—for each partial. Although the frequency of each partial could be calculated using the equation proposed by Fletcher *et al.* [25], such as in the works of Emiya *et al.* [26], [27]: $f_h = hf_0\sqrt{1+\beta h^2}$, where β is the inharmonicity coefficient of the piano tone [28], the encoding of the harmonic deviation of each partial pertaining to the genome of each individual genome was adopted so the system could be general enough to work with other kinds of pitched instruments.

2) Noise Level Estimation: During the transcription process, the algorithm compares the magnitude spectrum of each generated transcription with the original audio. This comparison should rely only on the spectral peaks of both sounds. Otherwise, spectral differences on spurious locations might lead the algorithm to an erroneous transcription. Thus, spectral data which does not belong to the spectral peaks should be discarded. This requires a way to somehow ignore spectral differences of spurious components.

As in [4], the noise is understood as generated from white noise filtered by a frequency-dependent spectral envelope. This way, the noise level is defined as the expected magnitude level of noise peaks and encoded in an additional chromosome. This chromosome has the noise level for each frequency bin [see Fig. 2(c)]. The noise level is encoded as an adaptive threshold below the maximum peak of the current time frame n, such that

$$z[k] = \max\left(X(n,k)\right) + ind.noise[k] \tag{4}$$

where $ind.noise[k]$ corresponds to the noise value encoded in the kth gene [see Fig. 2(c)]. Therefore, synthesized peaks below the $z[k]$ threshold are considered as noise peaks

$$\widehat{X}(n,k) = \begin{cases} \min\left(z[k], X(n,k)\right), & \text{if } \widehat{X}(n,k) < z[k] \\ \widehat{X}(n,k), & \text{if } \widehat{X}(n,k) \geq z[k] \end{cases} \tag{5}$$

where $X(n,k)$ is the magnitude of the kth bin from the nth frame of the original spectrum, and $\widehat{X}(n,k)$ represents the magnitude of the kth bin of the nth frame of the model spectrum (individual). This way, spectral data below the noise threshold will be considered as $\min(z[k], X(n,k))$. Thus, below the threshold, the spectrum of each generated transcription will be equal to the spectrum of the original audio: their difference below the noise threshold will always be zero (see top right plot of Fig. 3). Moreover, the spectral peaks (or at least most of them) will be above the threshold and, thus, have impact on the comparison. If it happens, for some reason, to have a spectral peak below the

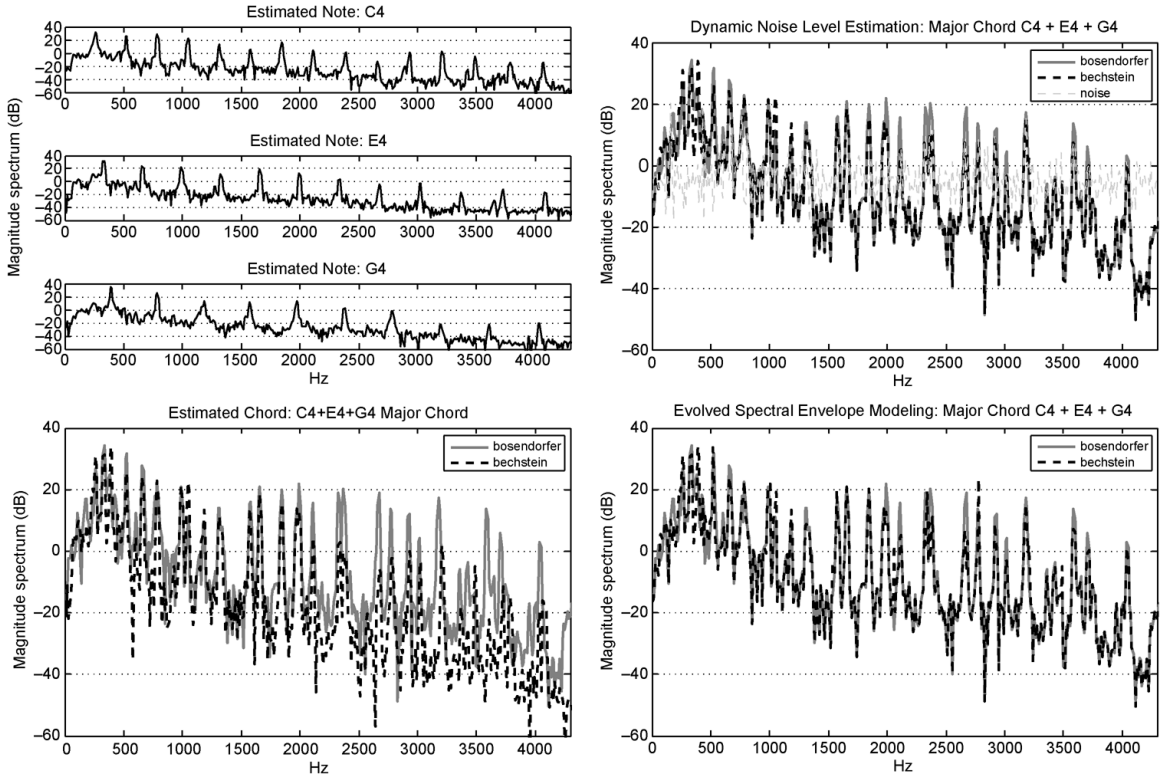


Fig. 3. Bottom left plot represents a major chord (C4-261.6256 Hz; E4-329.6276 Hz; and G4-391.9954 Hz) played by a Bosendorfer piano (original audio) and its generated transcription (Bechstein). The spectrum of the generated transcription consists of the sum of each estimated component (top three plots on the left of the same figure). The top right plot represents the same spectrum after applying the noise level estimation (light gray) and, finally, the bottom right plot represents the latter spectrum with the evolved spectral envelope modeling.

threshold there are two hypotheses: 1) if the corresponding spectral peak on the original audio is also below the threshold: they will be equal (see top right plot of Fig. 3, below 3500 Hz); 2) if the corresponding spectral peak on the original audio is above the threshold: the spectral peak of the candidate transcription will be equal to the threshold (see top right plot of Fig. 3, below 3500 Hz)—this way, it will still have impact on the comparison, but since its difference between the corresponding spectral peak on the original audio is diminished, it is easier for the adaptive spectral envelope modeling to compensate their differences and to do the rest.

The bottom left plot on Fig. 3 shows a major chord (C4-261. 6256 Hz; E4-329.6276 Hz; and G4-391.9954 Hz) played by a Bosendorfer piano (original audio) and its generated transcription (Bechstein). The spectrum of the generated transcription consists of the sum of each estimated component (top three plots on the left of the same figure). The top right plot shows how the algorithm sees the generated mixture played by the internal synthesizer (Bechstein) after applying the noise level estimation, and the bottom right plot shows how the algorithms sees the generated mixture after applying both noise model estimation and spectral envelope. If we compare the bottom left plot, which represents the original audio versus the generated mixture with the bottom right plot, where both dynamic noise level estimation and adaptive spectral envelope modeling were applied, we can see that the two initially different spectra became almost identical. This way, the algorithm will consider the generated transcription as correct despite their spectral differences (bottom left plot).

B. Fitness Evaluation

A good evaluation of each candidate's quality leads to a better selection of candidate solutions to form the next generation, speeding up the convergence of the algorithm towards a possible maximum. On the other hand, a less efficient quality (fitness) evaluation of each candidate (individual) can drastically reduce the evolution of the genetic algorithm. The fitness function is the key in the evolution/convergence of the genetic algorithms when solving different kinds of problems.

To evaluate candidate transcriptions, first we need to render them to an audio signal and then compare the corresponding audio signals to the input audio segment. Transcriptions whose audio is similar to the audio input are closer to the desired solution and, thus, have fewer errors. The comparison between the candidate transcriptions and the input audio segment is done in the frequency domain.

For the rendering process, we considered a dynamic range of 16 dB, that is: a note can vary its dynamics between 1 and 127. In particular, 127 MIDI velocity value corresponds to $+8$ dB gain and 1 MIDI velocity corresponds to -8 dB gain and 64 MIDI velocity corresponds to 0 dB gain. The gain, according to each note dynamic, is given by $10^{(vel-64)/80}$. After each note offset, the following release equation is applied: $release(t) = (2000-(t/36))/(2000+t)$, where t varies from $t = 0 \ldots 72\,000$.

The current fitness function is based on the log spectral distance or log spectral distortion and was chosen empirically

among several other spectral distances [29]. The fitness function is defined by the equation

$$f(i)=\sum_{n=1}^{nMax}\sqrt{\sum_{k=2}^{\frac{N}{2}}\left(\left[10\log_{10}\frac{|X(n,k)|}{\left|\widehat{X}(n,k)\right|}\right]^2\times\log_2\left(1+\frac{1}{k}\right)\right)} \tag{6}$$

where N is the size of the Hamming window, which is 93 ms (i.e., $N = 4096$ with 44 100-Hz sampling rate). k starts in 2 because it is the bin corresponding to the frequency of the first piano note ($\mathrm{A}_0 = 27,5$ Hz). The multiplication by $\log_2(1+(1/k))$ normalizes the weight of the bins of each octave so that, when summed, all the octaves have the same weighted sum equal to 1. As in (5), $X(n,k)$ is the magnitude of the kth bin from the nth frame of the original spectrum, $\widehat{X}(n,k)$ represents the magnitude of the kth bin of the nth frame of the model spectrum (candidate solution being evaluated).

C. Selection

Selection also plays a major role in the convergence of genetic algorithms. In each generation all individuals are evaluated according to their fitness. The fittest individuals have higher probability of being selected for reproduction, which results on the creation of fitter offspring, which ultimately leads to the algorithm convergence. Among the available selection operators, we have chosen "tournament" [5]. The tournament is a selection operator employed to choose each parent for the recombination: n individuals, where n is the size of the tournament, are randomly selected from the population, and then, from those n individuals, the fittest (winner of the tournament) is selected as a parent for breeding.

D. Recombination

Recombination is the main pillar of genetic algorithms: the building-block hypothesis [30] states that a genetic algorithm performs well when short, low-order, highly-fit schemata (often called building blocks) recombine to form even more highly-fit, higher-order schemata. In other words: the ability to produce fitter partial solutions by combining building blocks (recombination) is believed to be the primary source of the GA's search power.

In this specific problem, individuals might differ in the number of genes (detected F0s) so the classic one-point crossover [5] had to be adopted to recombine individuals with different number of genes by choosing different points-of-cut on each parent to generate the offspring. This one-point *cut-and-splice* recombination operator is described in Fig. 4(a).

The chromosomes encoding the spectral envelope of each instrument or the noise level estimation have always the same size: the number of genes is fixed. Thus, the classic one-point crossover operator [5] [see Fig. 4(b)] can be easily applied without any restriction.

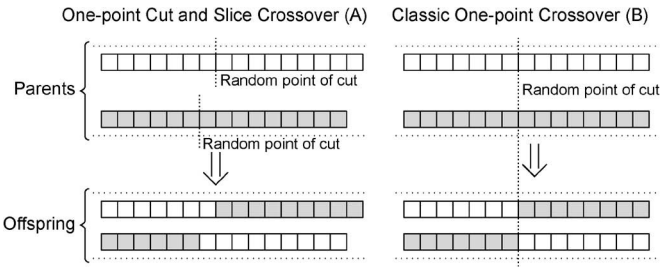


Fig. 4. Recombination operators: one-point *cut and slice* crossover (a) and classic one-point crossover (b).

E. Mutation

Mutation is also a very important genetic operator because it keeps biodiversity. The major feature of mutation is to avoid the genetic algorithm to be stuck in a local maximum by applying random changes in the individual's chromosome.

The mutation operator consists on the following steps described in Algorithm 1, where P is the mutation probability.

Algorithm 1 Mutation

1: **for** each gene **do**
2: $r \leftarrow$ random()
3: **if** $r \leq P_m$ **then**
4: Choose one of the following mutations:
note change
duration change by [−50 ms, 50 ms]
harmonic change
add a new note
velocity change by [−8, 8]
timbre change
remove the current gene
5: **end if**
6: **end for**

During the *note change* and *add a new note* mutations, the note value is chosen from a list containing all allowed notes. This list is previously calculated according to the most prominent spectral peaks in each frame (see Appendix A). The *harmonic change* mutation changes the current note to a harmonic location: minus 12 semitones (half of the frequency of the note—one octave below); minus 19 semitones; minus 24 semitones (one fourth of the frequency of the note—two octaves below); minus 28 semitones; minus 31; minus 34 and, minus 36. This mutation has the purpose of solving harmonic errors that may occur in the detection of the possible notes, since the detection only selects notes from the most prominent spectral peaks. *Timbre change* mutation happens to change the instrument that plays a given note. This mutation exists to improve the support of other kind of pitched instruments.

1) Spectral Envelope Modeling: The chosen mutations for this chromosome are changing the gain of a harmonic by a random value in the range [−12, 12] dB and changing the inharmonicity deviation using a bin value in the range [−3, 3]. The gain for F0 and its inharmonicity deviation are always 0 since they are not coded in the chromosome.

2) Noise Level Estimation: The mutation may occur in each gene and changes the power magnitude of the corresponding bin by a value in the range [−3, 3] dB.

F. Initialization

According to our previous results [19], the initial population has two major contributions to the end result. First, if the initial population is created nearer (or even half way) to the final result than a randomly generated initial population, the genetic algorithm will need a much smaller number of generations to achieve the target result. Nevertheless, it is also important to have a very heterogeneous initial population to allow a better exploration of different areas of our search space.

The first step to get good results is to find a way to create an initial population somehow based on the original audio signal. Although the authors are aware that the initial population could have been based on cepstrum [31], they chose to base it on the major peaks of the spectrogram. This happens to ensure that the genetic algorithm has enough biodiversity (genetic material) to perform the search process. The main power behind genetic algorithms relies on the capability of selecting the best parts of each candidate solution and recombining them into fitter and fitter solutions, something which requires a more heterogeneous population [32]. Thus, for the starting population, each individual is created with a random number of notes selected from the previously generated list of possible notes (see Appendix A). After its creation, each individual suffers ten forced mutations.

Although the initialization stage depends on the spectral peak picking to determine the list of possible notes, the lack of frequency resolution does not hinder the accurate peak picking at the low frequencies, and thus does not affect the estimation of those notes. During the selection of possible notes stage, the α most prominent peaks are selected from the original spectrum and then, for each peak, the corresponding musical notes are added to the list of possible notes (see Appendix A). This process takes into account that each bin (specially on lower frequencies) might correspond to several musical notes and, if it is the case, all those notes are added to the possible notes list. This way, even with low frequency resolution, we assure that at least the correct note is added to the list of possible notes. Therefore, the algorithm is able to choose the correct note from the set of candidates. Moreover, since we are selecting the α most prominent peaks, there is a high probability of selecting harmonically related notes of the correct ones. If this is the case, the harmonic change mutation will fix those notes.

Tests were also performed using the cepstrum for this process, but the experiments have shown that despite having several individuals that are harmonically related, the genetic algorithm performs better when the initial population is based on the spectrogram.

G. Survivor Selection

During recombination, each pair of individuals generates two offsprings. This leads to an overcrowded population. The survivor selection operator chooses which individuals should or should not pass to the next generation. The chosen selection method consists in determining the best individuals for survival. Also, 5% of the new generation consists on copies of the best individual of the previous generation, each with one forced mutation. This extends the robustness of the genetic algorithm, improving the global search by using local search on the vicinity of the best achieved solution [33].

V. Hill-Climber

Algorithm 2 Hill-Climber

1: $best \leftarrow$ individual returned by the genetic algorithm
2: $bestFitness \leftarrow best.evaluateFitness()$
3: $i \leftarrow 1$
4: **while** $i \leq best.numberOfNotes$ **do**
5: $ind \leftarrow best$
6: change $ind.notes[i]$ duration by +50 ms
7: **if** $ind.notes[i]$ now overlaps with another note **then**
8: merge both notes
9: **end if**
10: $fit \leftarrow ind.evaluateFitness()$
11: **if** $fit \geq bestFitness$ **then**
12: $best \leftarrow ind$
13: $bestFitness \leftarrow fit$
14: **else**
15: $i \leftarrow i + 1$
16: **end if**
17: **end while**
18: **return** MIDI file

The algorithm of the Hill-Climber (see Algorithm 2) consists of the following steps: transversing all musical notes and, for each note, increasing its duration by 50 ms. If this note now overlaps with another note both notes are merged. Also, if the quality of the individual improves, this process is then repeated on the same musical note; otherwise, the last change is discarded and the algorithm goes for the next musical note.

The output of the system is the result achieved by the hill-climber. The impact of the Hill-Climber on the overall results is studied in Section VI-D.

VI. Experiments and Results

A. Implementation and Tuning

The proposed approach is implemented using the C programming language. The transcription of each audio segment is run in parallel (one thread per segment). The computational time of the approach is 60 × real time. Several tests were carried across a selection of audio files, including a development database of 2700 mixtures from seven different pianos, with polyphony levels from 2 to 7. Those same tests employed various frame lengths, window functions and hop sizes. A 93-ms frame length with Hamming window function and 75% hop size were chosen empirically. $\alpha = 10$ is used for generating the possible notes list (see Appendix A) and the number of harmonics for the spectral envelope modeling was also empirically set to 20. The algorithm was also set to a 5 limit polyphony since polyphony levels in musical recordings have a 4.5 average polyphony and a

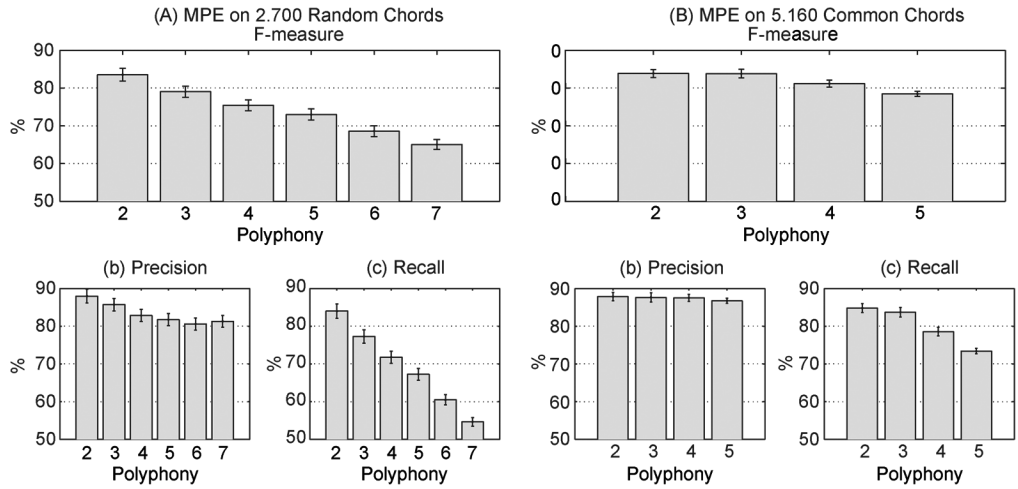


Fig. 5. (a) Multi-pitch estimation results for each polyphony on 2.700 random-pitched chords. (b) Multi-pitch estimation results for each polyphony on 5.160 common chords.

TABLE I
GENETIC ALGORITHM PARAMETERS

Genetic Algorithm parameters	
population size	200
maximum number of generations	50
probability of recombination	0.8
parent Selection	tournament (tournament size = 2)
probability of mutation	0.2 (transcription) 0.01 (spectral envelope) 0.02 (noise)

3.1 standard deviation reported for a number of classical music pieces [34, p. 114]. The probabilities and other parameters specific to the Genetic Algorithm are shown in Table I.

Although the resolution of 10.76 Hz might not appear enough for discriminating notes starting at MIDI note A0 (25.500 Hz), it suits their spectrum comparison. Recall that the algorithm searches for the best combination of notes by performing an evaluation based on their spectrum comparison with the original audio. Thus, same musical notes have similar harmonic locations, even if played by different pianos. Therefore, and that is also the case for lower notes, the algorithm tends to choose the correct piano samples because they correlate best with those played in the original audio. In fact, the algorithm is capable of identifying and discriminating notes from the 21 MIDI note (A0-27.500 Hz) to the 108 MIDI note (C8-4186.0 Hz). Also, the comparison between the original audio and the individual's audio is made on the frequency domain, frame by frame. A frame overlap ratio of 25% means less STFT frames and, therefore, fewer comparisons, which in turn means having fewer mathematical calculations that consequently computationally accelerate the algorithm. Moreover, less STFT frame comparisons mean fewer errors in the signal comparison that might occur due to spurious components or harmonic cancellation, which ultimately leads to a faster conversion of the genetic algorithm, hence better results. Even though it might occur an amplitude modulation during overlap-add, it will happen on both signals being compared therefore, so it will not affect the comparison.

B. Evaluation

The proposed algorithm has been tested on a database called MAPS [27], [34] consisting of around 10,000 piano sounds either recorded by using an upright Disklavier piano or generated by several virtual piano software products based on sample sounds. The development set and the test set are disjoint. A set of 2,700 random chords between A0 (25.500 Hz) and C8 (4186 Hz) with polyphony levels ranging from 2 to 7 were used in the former while the latter comprises 2700 random chords and 5160 common chords from western music (major, minor, etc.) from C2 (65.406 Hz) to B6 (1975.5 Hz). For each sound, a single 93ms-frame located after the last onset time is extracted and analyzed. In total, 10 560 audio files were used from two upright pianos and five grand pianos. The authors considered only the frame after the last onset for both training and test tasks because both tasks consisted in transcribing simple chords and also due to time restrictions since there were used a total of 10 560 different chords. During all other tests, the algorithm performs a frame by frame analysis.

General results are presented in Fig. 5. Relevant items are defined correct notes after rounding each F0 to the nearest halftone. Typical metrics are used: the recall is the ratio between the number of relevant items and of original items; the precision is the ratio between the number of relevant items and of detected items; and the F-measure is the harmonic mean [35] between the precision and the recall. In this context, our system returns F-measures of 84%, 79%, 74%, 73%, 69%, and 65% for polyphony 2, 3, 4, 5, 6, and 7 on random chords and 84%, 84%, 81%, and 78% for polyphony 2, 3, 4, and 5 on common chords. Moreover, the precision is high for all polyphony levels whereas the recall is decreasing whenever polyphony increases.

The ability of the system to infer the polyphony levels (independently of the pitches) is presented on Fig. 6. For polyphony levels from 2 to 5 the system only fails on detecting the correct polyphony level on polyphony 4 on the random chords data set. Polyphony levels of 6 and 7 obviously fail because the system is set to a maximum 5 polyphony. On the other hand, the

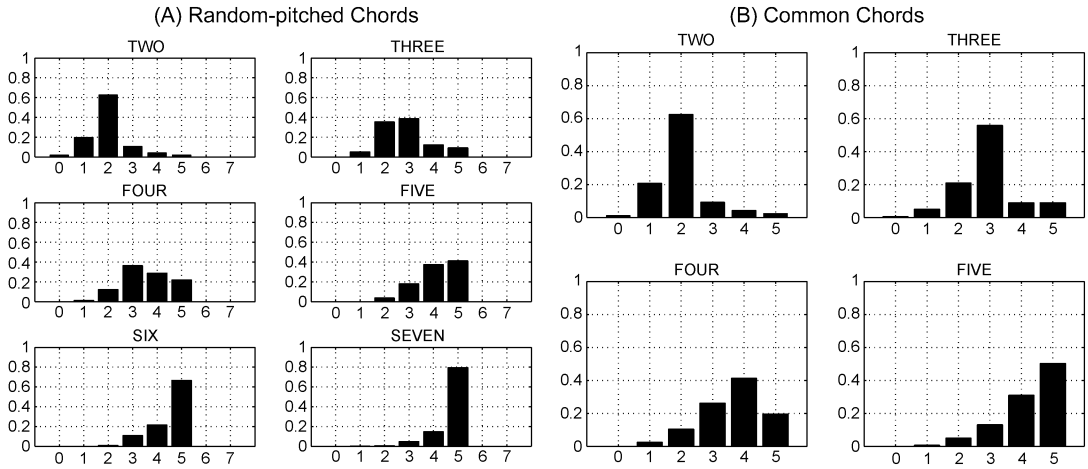


Fig. 6. Distribution of estimated polyphony for the polyphony from 2 to 7 on (a) random-pitched chords and from 2 to 5 on (b) common chords. The title of each subfigure indicates the correct polyphony; the x-axis represents the estimated polyphony; the y-axis represents the percentage of the estimated polyphony among all instances. The peaking at the correct polyphony is observed for polyphony below six, except for four.

system successfully detects the correct polyphony for all polyphonies in the common chords data set. Also, F-measure values are between 6% and 10% better for common chords than for random-pitched chords. This suggests that, while the algorithm faces more harmonically related notes in common chords (spectral overlap), widely-spread F0s in random chords are a bigger difficulty. By limiting the algorithm polyphony to 5, the proposed system underestimates the polyphony level since the parameter tuning consists in optimizing the F-measure on the development set. This objective function could have been changed to take the polyphony level balance into account. This would result in reducing the polyphony underestimation trend. However, the overall F-measure would decrease. Moreover, from a different perspective, it has been shown that a missing note is generally less annoying than an added note when listening to a resynthesized transcription [36]. Thus, underestimating the polyphony may be preferable to overestimating it. Still, this trend turns out to be the main shortcoming of the proposed method, and should be fixed in the future so it can efficiently address sounds with polyphony higher than 5 notes.

While the test database has seven different pianos, the internal synthesizer of the algorithm consists only of three pianos, which means that the spectral envelope modeling plays a major role in the achieved results by adapting the internal piano samples to the seven pianos on the database. Moreover, the results are comparable from one piano to another, with only a small % deviation. This means that the results do not significantly depend on the upright/grand piano differences.

C. Comparison With Other State-of-the-Art Algorithms

For a deeper analysis, we decided to extend a previous study performed by Emiya [26]. This study compares several state-of-the-art music transcription algorithms: [26], [37], [38], and [39]. We included four new algorithms: [21], [40], [41][2] and the proposed method. In total, our algorithm is compared to seven other state-of-the-art transcription algorithms:

- Vincent'10 [40];
- Reis'09 [21]
- Emiya EUSIPCO'08 [26];
- Vincent B'07—The baseline method presented in [37];
- Vincent H'07—The harmonic method presented in [37];
- Bertin'07 [38];
- Marolt'04 [39];
- Ryynänen'05 [41]—The authors thought that the inclusion of this algorithm was very important for the study performed since this algorithm won the last MIREX [42], [43] competitions.

Reis'09 [21] was included for comparison because we believe it is the best algorithm of all previous genetic algorithm approaches: Garcia algorithm [15] does not take into account onset time and offset time, Lu's approach [16] and Reis and Fernandez's algorithm [18] are limited to simple mathematical models (e.g., sawtooth wave, rectangle wave, sine wave) and for Reis *et al.* [19], although it applies the first genetic algorithm approach to music transcription using real audio recordings, it lacks a means to deal with harmonic overfitting. Also, Reis *et al.* [20] and [21] are essentially the same algorithm, except that the latter approach was extended to multi-timbre. This way, we can see how our proposal performs when compared to the previous best genetic algorithm approaches to automatic music transcription.

These algorithms were run on nine randomly chosen pieces of music used on Emiya benchmark [26][3] All the results that will be presented on this paper are available at the author's website[4]. Along with these results we also provide the audio files and their MIDI representation in visual form.

Fig. 7(a) shows the results obtained by our algorithm in comparison to the other seven state-of-the-art algorithms and (B) shows the Friedman Mean Ranks with regard to F-measure

[2]The authors asked to several researchers in the field for their algorithms so that they could also be included in this study. These were the algorithms provided.

[3]see: http://www.irisa.fr/metiss/vemiya/EUSIPCO08/bench0.html

[4]http://www.estg.ipleiria.pt/~gustavo.reis/benchmark

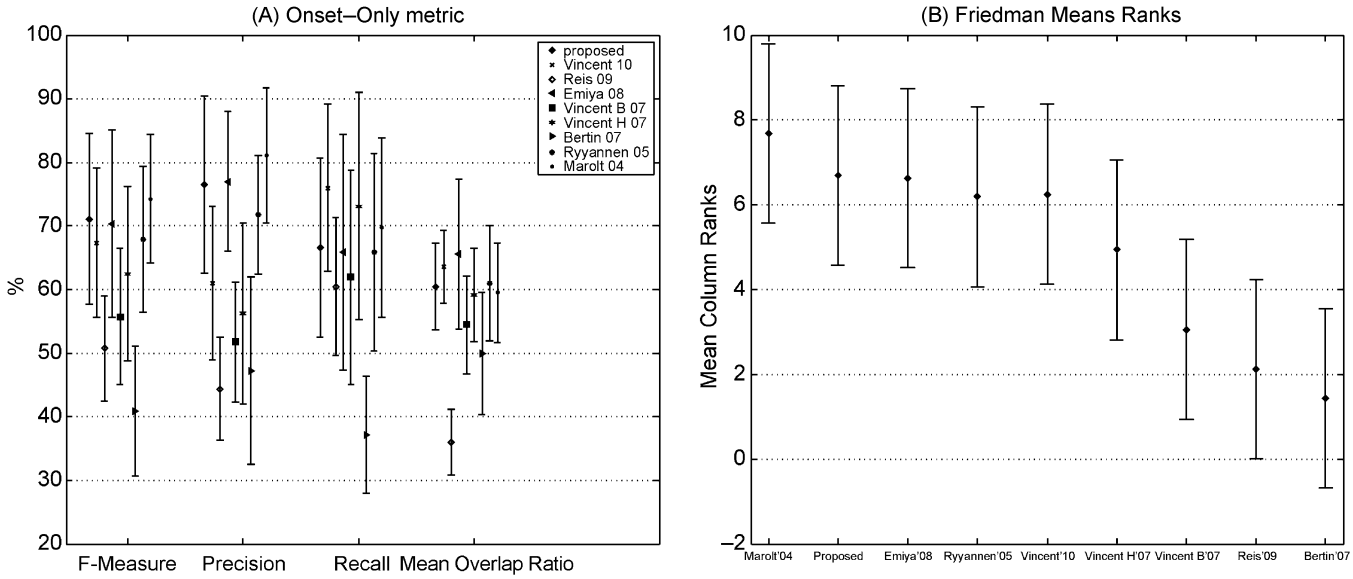


Fig. 7. (a) Onset-only F-measure, Precision, Recall, and Mean Overlap Ratio, respectively. (b) Friedman Mean Ranks with regard to F-measure on individual files.

on individual files. Results in Fig. 7(a) are presented using the onset-only metric. In this metric a correct note implies a correct onset with a deviation up to 50 ms. Results are presented on the onset-only metric as Recall, Precision, F-measure, and Mean Overlap Ratio (MOR) [41]. Mean Overlap Ratio is an averaged ratio between the length of the intersection of the temporal supports of an original note and its transcription, and of the length of their union. This measure acts more like a guideline for researchers to know how the correctly transcribed notes intersect with the original notes in terms of note duration. It is also used to measure the phrasing similarity with the original piece.

Performance rates on Fig. 7(a) have been averaged with respect to all tested musical pieces, and the respective standard deviation is also represented. While we have achieved low F-measure deviation on the evaluation of audio chords (multi pitch estimation), on the automatic transcription benchmark we have a deviation around 15%, no matter the method. This happens because in the chord evaluation we used isolated frames composed of one chord, while here the evaluation implies several other difficulties, like having asynchronous notes overlapping in time, detecting onsets, estimating the end of damping notes, dealing with reverberation queues and so on. Thus, large standard deviation values are due to the dependency of musical excerpts. For instance: F-measure greater than 85% is reached on musical pieces with slow tempo or low polyphony, while fast pieces are generally difficult to transcribe. Since the results dramatically depend on the database, this was the main reason why we decided to extend a previous study. This way we can have a more realistic comparison among all tested algorithms.

In this context [Fig. 7(a)] our system is comparable to the state-of-the-art: it ranked the 2nd place, below Marolt'04 and above Emiya'08. It should be noted that Emiya'08 is the algorithm with higher Mean Overlap Ratio, which suggests that the used HMM framework for note tracking is efficient in both selecting pitches among candidates, and also in detecting their possible endings.

Regarding the proposed approach and Reis'09, both algorithms have large differences in F-Measure (20%), Precision (32%) and Mean Overlap Ratio (25%) and a small difference on Recall (6%). The large F-Measure difference (20%) shows that the proposed system has much better performance than the previous genetic algorithm approaches. It also features a smaller percentage of both false positive rate and false negative rate (Precision and Recall). Note that there is also a considerable difference on Mean Overlap Ratio (25%). This means that our system results in a more efficient transcription, enhancing the phrasing similarity with the original pieces and thus improving the subjective quality when hearing the correctly transcribed notes. The computation time of Reis'09 algorithm is 540 times real-time which is much higher than the proposed method: 60 times real-time. This also represents a significant improvement. Also, the most significant difference between both algorithms relies on Precision (32%). This shows that the adaptive spectral envelope modeling, along with the dynamic noise level estimation, play a major role in reducing the false positive rate: precision is the percentage of the transcribed notes that are correctly transcribed. Thus, the proposed algorithm is much more effective on reducing the harmonic overfitting. On the other hand, a lower difference on Recall tells us that both systems have a similar false negative rate, which again emphasizes that the main difference on both systems is on how the dynamic noise level estimation, along with the adaptive spectral envelope modeling, have a significant impact on reducing the false positive rate: this system is much more effective in dealing with the harmonic overfitting.

Fig. 7(b) shows that the proposed approach achieves the second best the mean rank, which means that, on average, our system ranked second place in each individual file. Table II shows the Tukey–Kramer Honestly Significant Difference

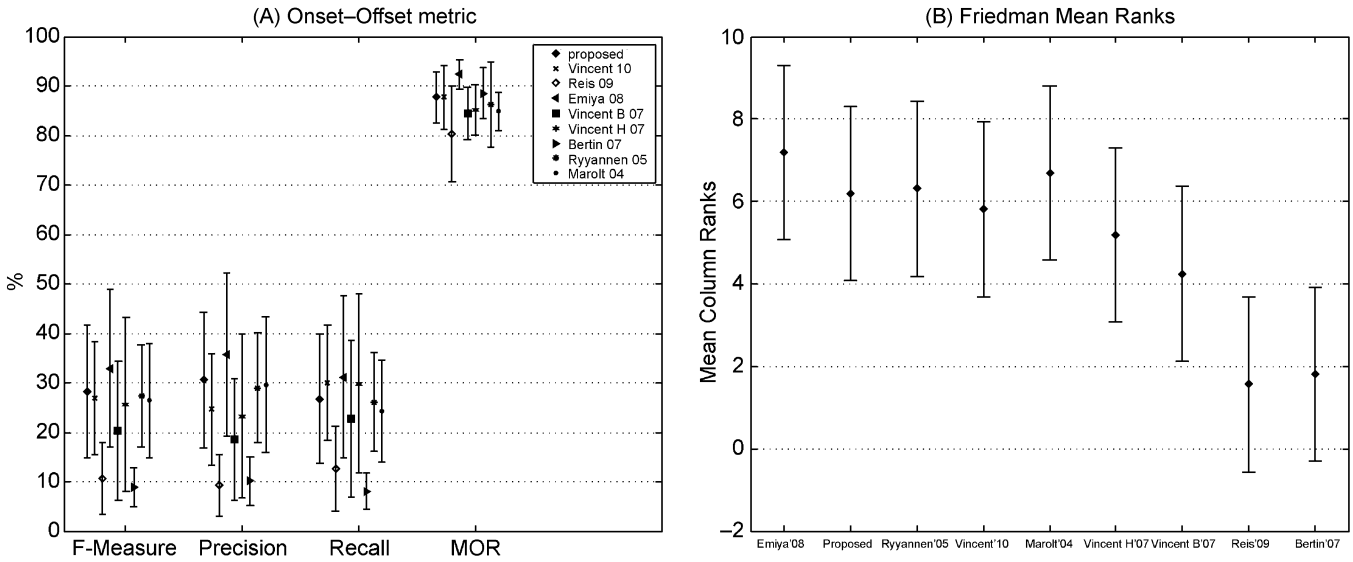


Fig. 8. (a) Onset–offset F-measure, precision, recall and mean overlap ratio, respectively. (b) Friedman mean ranks with regard to F-measure on individual files.

TABLE II
TUKEY–KRAMER HSD (HONESTLY SIGNIFICANT DIFFERENCE) MULTI-COMPARISON ONSET-ONLY METRIC

Algorithm	Algorithm	Lower Bound	Mean	Upper Bound	Significance
Marolt'04	Proposed	-3.2317	1.0000	5.2317	FALSE
Marolt'04	Emiya'08	-3.1692	1.0625	5.2942	FALSE
Marolt'04	Ryyannen'05	-2.7317	1.5000	5.7317	FALSE
Marolt'04	Vincent'10	-2.7942	1.4375	5.6692	FALSE
Marolt'04	Vincent H'07	-1.4817	2.7500	6.9817	FALSE
Marolt'04	Vincent B'07	0.3933	4.6250	8.8567	TRUE
Marolt'04	Reis'09	1.3308	5.5625	9.7942	TRUE
Marolt'04	Bertin'07	2.0183	6.2500	10.4817	TRUE
Proposed	Emiya'08	-4.1692	0.0625	4.2942	FALSE
Proposed	Ryyannen'05	-3.7317	0.5000	4.7317	FALSE
Proposed	Vincent'10	-3.7942	0.4375	4.6692	FALSE
Proposed	Vincent H'07	-2.4817	1.7500	5.9817	FALSE
Proposed	Vincent B'07	-0.6067	3.6250	7.8567	FALSE
Proposed	Reis'09	0.3308	4.5625	8.7942	TRUE
Proposed	Bertin'07	1.0183	5.2500	9.4817	TRUE
Emiya'08	Ryyannen'05	-3.7942	0.4375	4.6692	FALSE
Emiya'08	Vincent'10	-3.8567	0.3750	4.6067	FALSE
Emiya'08	Vincent H'07	-2.5442	1.6875	5.9192	FALSE
Emiya'08	Vincent B'07	-0.6692	3.5625	7.7942	FALSE
Emiya'08	Reis'09	0.2683	4.5000	8.7317	TRUE
Emiya'08	Bertin'07	0.9558	5.1875	9.4192	TRUE
Ryyannen'05	Vincent'10	-4.2942	-0.0625	4.1692	FALSE
Ryyannen'05	Vincent H'07	-2.9817	1.2500	5.4817	FALSE
Ryyannen'05	Vincent B'07	-1.1067	3.1250	7.3567	FALSE
Ryyannen'05	Reis'09	-0.1692	4.0625	8.2942	FALSE
Ryyannen'05	Bertin'07	0.5183	4.7500	8.9817	TRUE
Vincent'10	Vincent H'07	-2.9192	1.3125	5.5442	FALSE
Vincent'10	Vincent B'07	-1.0442	3.1875	7.4192	FALSE
Vincent'10	Reis'09	-0.1067	4.1250	8.3567	FALSE
Vincent'10	Bertin'07	0.5808	4.8125	9.0442	TRUE
Vincent H'07	Vincent B'07	-2.3567	1.8750	6.1067	FALSE
Vincent H'07	Reis'09	-1.4192	2.8125	7.0442	FALSE
Vincent H'07	Bertin'07	-0.7317	3.5000	7.7317	FALSE
Vincent B'07	Reis'09	-3.2942	0.9375	5.1692	FALSE
Vincent B'07	Bertin'07	-2.6067	1.6250	5.8567	FALSE
Reis'09	Bertin'07	-3.5442	0.6875	4.9192	FALSE

(HSD) multi-comparison of the Friedman Mean Ranks calculated on Fig. 7(b). This table shows that the difference between the proposed system and the algorithm that ranked first is not statistically significant. Moreover, on this metric, our proposal is significantly better than Reis'09 (best algorithm of all previous genetic algorithm approaches) and Bertin'07.

Fig. 8(a) shows the same benchmark using the Onset-Offset metric. This metric also presents the results as Recall, Precision, F-measure, and MOR. In the onset–offset metric a correct note implies a correct onset with a deviation up to 50 ms and a correct offset with a deviation of up to 20% of the note length or 50 ms. Performance rates have also been averaged with respect to all tested musical pieces, and the respective standard deviation is also represented.

According to this metric, our system is also comparable to the state-of-the-art: it ranked the 2nd place, below Emiya'08 and above Vincent'10. It should be noted that, in this context, Emiya'08 is the algorithm with higher F-measure, Precision, Recall, and Mean Overlap Ratio. This happens because it is the most effective algorithm estimating the offset time, which reinforces what was mentioned before: the HMM framework used by Emiya'08 for note tracking is efficient in both selecting pitches among candidates, and also in detecting their possible endings. However, Emiya'08 algorithm has a computing time of 200 times real-time while our system's computing time is 60 times real-time. Fig. 8(b) shows that the proposed approach achieves the fourth best mean rank, which means that, despite having the second best overall mean on F-Measure, on average, our system ranked fourth place in each individual file. Table III shows the Tukey–Kramer HSD multi-comparison of the Friedman Mean Ranks calculated on Fig. 8(b). This table shows that the difference between the proposed system and the algorithm that ranked first is not statistically significant. Moreover, on this metric, our proposal is significantly better than Reis'09 and Bertin'07.

Finally, one last metric was employed for algorithm comparison: the Hybrid Decay/Sustain Score [44]. This metric was employed because it is the one that best correlates with the human hearing perception [44]. Fig. 9(a) shows the results obtained using this metric. Results are presented as Decay Score, Sustain Score, and Final Score: Decay Score is used for percussive pitched instruments and employs a note oriented approach considering only pitches and onsets, generating a score ([0–100]%) for each note; Sustain Score is used for sustain musical instruments (e.g., woodwind) and employs a time oriented approach measuring the overlap between the original and transcribed notes; the Final Score is the average value between Sustain Score and Decay Score.

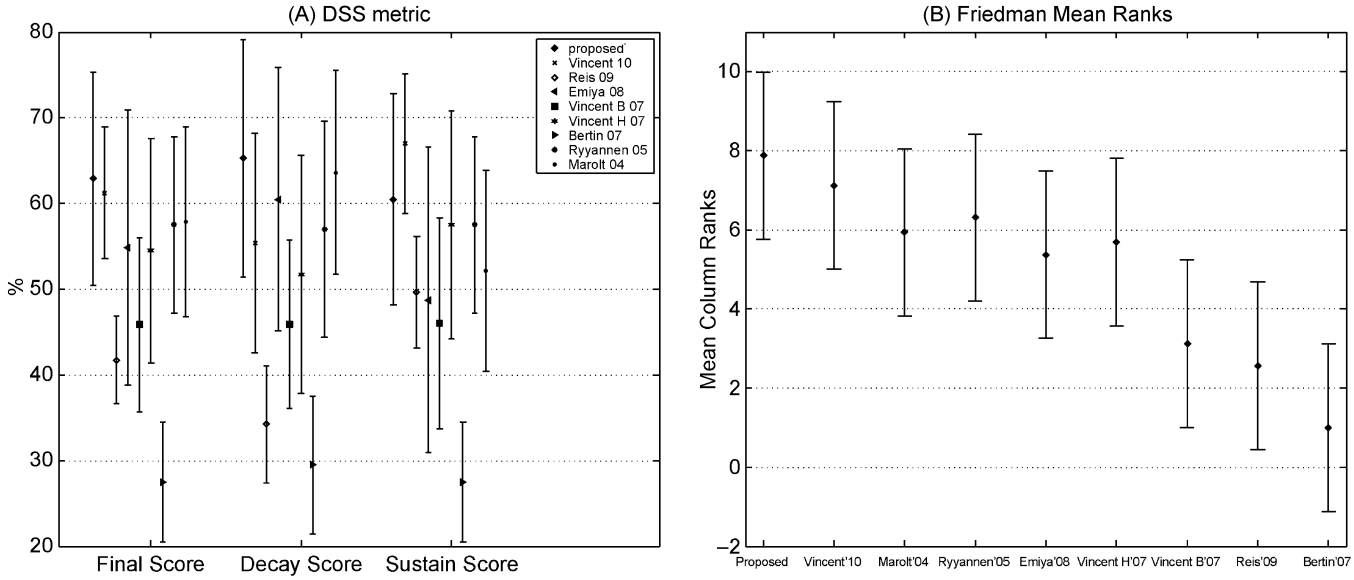


Fig. 9. Final Score, Decay Score, and Sustain Score, respectively. (a) DSS metric. (b) Friedman Mean Ranks.

TABLE III
TUKEY–KRAMER HSD (HONESTLY SIGNIFICANT DIFFERENCE) MULTI-COMPARISON ONSET-OFFSET METRIC

Algorithm	Algorithm	Lower Bound	Mean	Upper Bound	Significance
Emiya'08	Proposed	-3.2339	1.0000	5.2339	FALSE
Emiya'08	Ryyannen'05	-3.3589	0.8750	5.1089	FALSE
Emiya'08	Vincent'10	-2.8589	1.3750	5.6089	FALSE
Emiya'08	Marolt'04	-3.7339	0.5000	4.7339	FALSE
Emiya'08	Vincent H'07	-2.2339	2.0000	6.2339	FALSE
Emiya'08	Vincent B'07	-1.2964	2.9375	7.1714	FALSE
Emiya'08	Reis'09	1.3911	5.6250	9.8589	TRUE
Emiya'08	Bertin'07	1.1411	5.3750	9.6089	TRUE
Proposed	Ryyannen'05	-4.3589	-0.1250	4.1089	FALSE
Proposed	Vincent'10	-3.8589	0.3750	4.6089	FALSE
Proposed	Marolt'04	-4.7339	-0.5000	3.7339	FALSE
Proposed	Vincent H'07	-3.2339	1.0000	5.2339	FALSE
Proposed	Vincent B'07	-2.2964	1.9375	6.1714	FALSE
Proposed	Reis'09	0.3911	4.6250	8.8589	TRUE
Proposed	Bertin'07	0.1411	4.3750	8.6089	TRUE
Ryyannen'05	Vincent'10	-3.7339	0.5000	4.7339	FALSE
Ryyannen'05	Marolt'04	-4.6089	-0.3750	3.8589	FALSE
Ryyannen'05	Vincent H'07	-3.1089	1.1250	5.3589	FALSE
Ryyannen'05	Vincent B'07	-2.1714	2.0625	6.2964	FALSE
Ryyannen'05	Reis'09	0.5161	4.7500	8.9839	TRUE
Ryyannen'05	Bertin'07	0.2661	4.5000	8.7339	TRUE
Vincent'10	Marolt'04	-5.1089	-0.8750	3.3589	FALSE
Vincent'10	Vincent H'07	-3.6089	0.6250	4.8589	FALSE
Vincent'10	Vincent B'07	-2.6714	1.5625	5.7964	FALSE
Vincent'10	Reis'09	0.0161	4.2500	8.4839	TRUE
Vincent'10	Bertin'07	-0.2339	4.0000	8.2339	FALSE
Marolt'04	Vincent H'07	-2.7339	1.5000	5.7339	FALSE
Marolt'04	Vincent B'07	-1.7964	2.4375	6.6714	FALSE
Marolt'04	Reis'09	0.8911	5.1250	9.3589	TRUE
Marolt'04	Bertin'07	0.6411	4.8750	9.1089	TRUE
Vincent H'07	Vincent B'07	-3.2964	0.9375	5.1714	FALSE
Vincent H'07	Reis'09	-0.6089	3.6250	7.8589	FALSE
Vincent H'07	Bertin'07	-0.8589	3.3750	7.6089	FALSE
Vincent B'07	Reis'09	-1.5464	2.6875	6.9214	FALSE
Vincent B'07	Bertin'07	-1.7964	2.4375	6.6714	FALSE
Reis'09	Bertin'07	-4.4839	-0.2500	3.9839	FALSE

TABLE IV
TUKEY–KRAMER HSD (HONESTLY SIGNIFICANT DIFFERENCE) MULTI-COMPARISON HYBRID DECAY/SUSTAIN METRIC

Algorithm	Algorithm	Lower Bound	Mean	Upper Bound	Significance
Proposed	Vincent'10	-3.4817	0.7500	4.9817	FALSE
Proposed	Marolt'04	-2.2942	1.9375	6.1692	FALSE
Proposed	Ryyannen'05	-2.6692	1.5625	5.7942	FALSE
Proposed	Emiya'08	-1.7317	2.5000	6.7317	FALSE
Proposed	Vincent H'07	-2.0442	2.1875	6.4192	FALSE
Proposed	Vincent B'07	0.5183	4.7500	8.9817	TRUE
Proposed	Reis'09	1.0808	5.3125	9.5442	TRUE
Proposed	Bertin'07	2.6433	6.8750	11.1067	TRUE
Vincent'10	Marolt'04	-3.0442	1.1875	5.4192	FALSE
Vincent'10	Ryyannen'05	-3.4192	0.8125	5.0442	FALSE
Vincent'10	Emiya'08	-2.4817	1.7500	5.9817	FALSE
Vincent'10	Vincent H'07	-2.7942	1.4375	5.6692	FALSE
Vincent'10	Vincent B'07	-0.2317	4.0000	8.2317	FALSE
Vincent'10	Reis'09	0.3308	4.5625	8.7942	TRUE
Vincent'10	Bertin'07	1.8933	6.1250	10.3567	TRUE
Marolt'04	Ryyannen'05	-4.6067	-0.3750	3.8567	FALSE
Marolt'04	Emiya'08	-3.6692	0.5625	4.7942	FALSE
Marolt'04	Vincent H'07	-3.9817	0.2500	4.4817	FALSE
Marolt'04	Vincent B'07	-1.4192	2.8125	7.0442	FALSE
Marolt'04	Reis'09	-0.8567	3.3750	7.6067	FALSE
Marolt'04	Bertin'07	0.7058	4.9375	9.1692	TRUE
Ryyannen'05	Emiya'08	-3.2942	0.9375	5.1692	FALSE
Ryyannen'05	Vincent H'07	-3.6067	0.6250	4.8567	FALSE
Ryyannen'05	Vincent B'07	-1.0442	3.1875	7.4192	FALSE
Ryyannen'05	Reis'09	-0.4817	3.7500	7.9817	FALSE
Ryyannen'05	Bertin'07	1.0808	5.3125	9.5442	TRUE
Emiya'08	Vincent H'07	-4.5442	-0.3125	3.9192	FALSE
Emiya'08	Vincent B'07	-1.9817	2.2500	6.4817	FALSE
Emiya'08	Reis'09	-1.4192	2.8125	7.0442	FALSE
Emiya'08	Bertin'07	0.1433	4.3750	8.6067	TRUE
Vincent H'07	Vincent B'07	-1.6692	2.5625	6.7942	FALSE
Vincent H'07	Reis'09	-1.1067	3.1250	7.3567	FALSE
Vincent H'07	Bertin'07	0.4558	4.6875	8.9192	TRUE
Vincent B'07	Reis'09	-3.6692	0.5625	4.7942	FALSE
Vincent B'07	Bertin'07	-2.1067	2.1250	6.3567	FALSE
Reis'09	Bertin'07	-2.6692	1.5625	5.7942	FALSE

According to this metric, the proposed system ranks the 1st place, above Vincent'10. This means that our system results in an efficient transcription, enhancing the phrasing similarity with the original piece and thus improving the subjective quality when hearing the correctly transcribed notes. Note that, since we are dealing with piano transcriptions, we can consider the value Decay Score instead of the Final score. In this case all algorithms rank the same places (not the same results), except Vincent'10. This happens because the latter algorithm has a relatively high Sustain Score.

Fig. 9(b) shows that the proposed approach achieves the best mean rank, which means that, on average, our system ranked first place in each individual file. Moreover, Fig. 9(b) along with Table IV shows that our proposal is significantly better than Vincent B'07, Reis'09 and Bertin'07.

We believe that our approach has perceptually better results because, among all the other state-of-the-art algorithms, our system is the only one that tries to mimic the way how professional musicians learn to play a new song by ear: the algorithm "listens" to an audio file and then, during the transcription

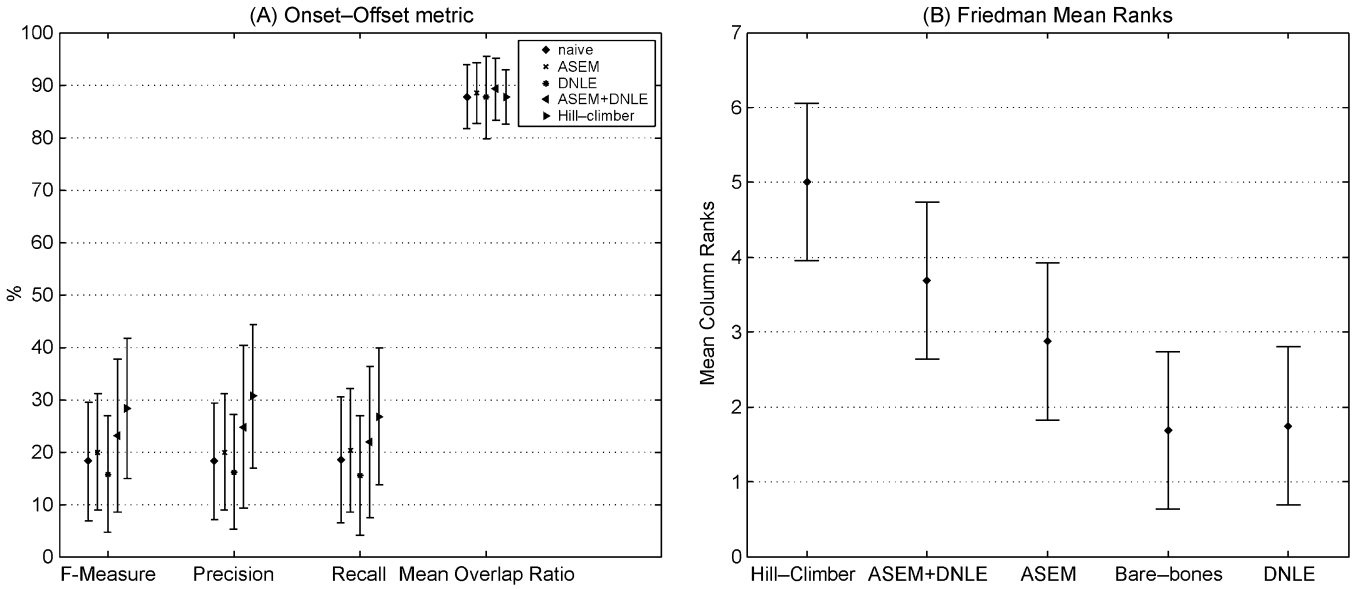


Fig. 10. (a) Contribution of each module to the global results of the proposed system—Onset–offset. (b) Friedman Mean Ranks with regard to F-measure on individual files.

process, is always comparing the several candidates, to figure out which one is closer to the original song. In the end, it returns the transcription that most resembled the original audio.

D. Contribution of Each Module to the Overall Results

Fig. 10 shows the contribution of each module to the global results of the proposed system: naïve, adaptive spectral envelope modeling (ASEM), dynamic noise level estimation (DNLE), both spectral envelope modeling and dynamic noise level estimation (ASEM+DNLE) and Hill-climber. Naïve corresponds to the genetic algorithm without the spectral envelope modeling and noise level estimation; ASEM is the naïve version with the spectral envelope modeling; DNLE is the naïve version with the dynamic noise level estimation; ASEM+DNLE corresponds to naïve version with both spectral envelope modeling and dynamic noise level estimation; and, finally, Hill-climber is the ASEM+DNLE with the Hill-climber applied, i.e., the whole system.

By analyzing Fig. 10 we can tell that the naïve (genetic algorithm without hill-climber, spectral envelope modeling and dynamic noise level estimation) has a performance of: 18.25% F-measure, 18.25% Precision, and 18.5% Recall. By enabling the adaptive spectral envelope modeling (ASEM), Precision, F-Measure and Recall have an improvement around 1.75%: ASEM by itself does not bring significant improvements on the quality of the results since differences on spurious components will bias the comparison (candidate evaluation), and thus, the convergence of the genetic algorithm. On the other hand, if we only enable the dynamic noise level estimation (DNLE), F-Measure decreases 2.5%, Precision decreases 1.3% and Recall decreases 3.0%. This happens because the noise estimation discards all the spurious information when evaluating the transcriptions: only the spectral peaks above the noise threshold are considered. This leads the algorithm on adding several notes in harmonic locations to compensate the timbre differences, decreasing both precision and recall.

TABLE V
TUKEY–KRAMER HSD (HONESTLY SIGNIFICANT DIFFERENCE) MULTI-COMPARISON ONSET-OFFSET METRIC

Algorithm	Algorithm	Lower Bound	Mean	Upper Bound	Significance
Hill-Climber	ASEM+DNLE	-0.7894	1.3125	3.4144	FALSE
Hill-Climber	ASEM	0.0231	2.1250	4.2269	TRUE
Hill-Climber	Bare-bones	1.2106	3.3125	5.4144	TRUE
Hill-Climber	DNLE	1.1481	3.2500	5.3519	TRUE
ASEM+DNLE	ASEM	-1.2894	0.8125	2.9144	FALSE
ASEM+DNLE	Bare-bones	-0.1019	2.0000	4.1019	FALSE
ASEM+DNLE	DNLE	-0.1644	1.9375	4.0394	FALSE
ASEM	Bare-bones	-0.9144	1.1875	3.2894	FALSE
ASEM	DNLE	-0.9769	1.1250	3.2269	FALSE
Bare-bones	DNLE	-2.1644	-0.0625	2.0394	FALSE

Adaptive spectral envelope modeling along with dynamic noise level estimation (ASEM+DNLE), improve the system's performance: F-measure increases 4.88% and Precision has a boost of 6.5%. This tells that adaptive spectral envelope modeling together with dynamic noise level estimation have a great impact on reducing the harmonic overfitting: the percentage of correctly transcribed notes increases around 6.5%, which means that the system significantly reduced the false positive rate. Recall improves 3.38%. The algorithm performs well because both ASEM and DNLE were designed to work together so that they can compensate each-other.

Hill-Climber gives the major improvement to the proposed system. It raises the performance of ASEM+DNLE by: 5.13% F-Measure, 5.88% Precision and 4.88% Recall.

Fig. 10(b), along with Table V show that the contributions made by Hill-Climber along with ASEM+DNLE are statistically significant: Hill-Climber with ASEM+DNLE are statistically better than the bare-bones GA, ASEM and DNLE.

The computing time of the proposed system is 60 times real-time. However, the proposed system has several parallelization capabilities since a separate genetic algorithm is run on each audio segment. This way, the transcription task could be run on

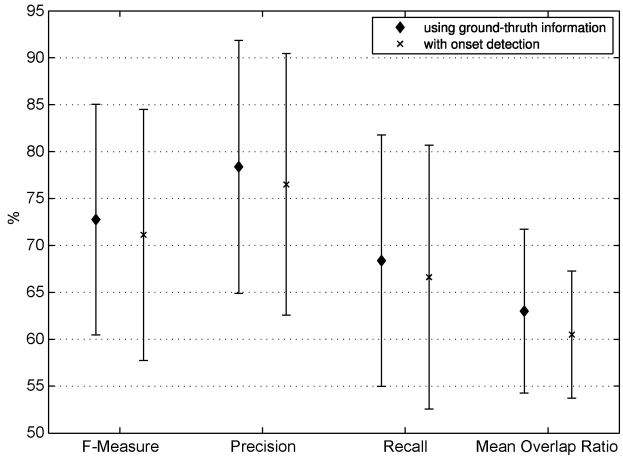


Fig. 11. F-measure, Precision, Recall, and Mean Overlap Ratio, using onset detection and ground-truth information.

a separate CPU for each audio segment. Thus, we could have one master CPU to apply onset detection and distribute the resulting audio segments for several CPUs and wait for their results. Afterwards, the master CPU merges their results into one whole transcription and applies the hill-climber.

E. Impact of the Onset Detector on the Overall Results

By taking into account that onsets are used as guidelines for the segmentation of the input signal, the occurrence of false negatives might have a considerable impact in the final results of the event segregation. To evaluate the impact of the chosen onset detection algorithm, the authors ran the same experiments using the ground-truth onset information as the output of the onset detector. Fig. 11 shows the comparison between the proposed system using the implemented onset detector and using the ideal onset detector.

The F-measure difference between using the ground-truth information rather than the onset detector is around 1%, which means that the implemented onset detection has good performance and does not have a significant impact in the proposed system: the performance of the algorithm does not drop significantly.

VII. Conclusion and Future Work

We have proposed a new approach for automatic transcription of polyphonic piano music using genetic algorithms for multi-pitch estimation and also for spectral envelope modeling and dynamic noise level estimation. The transcription process happens in three stages: first an onset detector is applied, so that the audio can be separated in several audio segments; then, for each segment, the genetic algorithm is applied to perform the transcription of the corresponding audio segment; and, finally, a hill-climber is applied to adjust note durations that cross multiple audio segments. The performance of the algorithm does not drop significantly when compared to the usage of the ideal onset detector. Nevertheless, since there is some decay in the quality of the results, the user is able to use any other onset detector and use its data as system input. Due to the fact that the audio segmentation is based on onset information, the duration of each segment is very short, which reduces the search space of the genetic algorithm. Thus, 50 generations are more than enough for the algorithm to find the appropriate solution. Each candidate solution is encoded as a set of discrete note events associated with a timbre and noise model. The evolution of these two models aids the transcription process because it mitigates the spectral differences between different instruments. The Hill-Climber, by adjusting the duration of the transcribed notes, gives a major contribution to the quality of the results from the perception point of view.

The performance of the method was measured using 7860 audio files and was also compared to the state-of-the-art. The comparison was made using three different metrics: onset-only (to measure the ability of detecting the F0s), onset-offset (to measure the overlap between the original score and the transcribed score) and, finally, Hybrid Decay/Sustain score for an evaluation from the human hearing perception point of view. The proposed method achieved satisfying results when compared to other algorithms on all metrics: it ranked as the best algorithm on the metric that best correlates with the human hearing perception—Hybrid Decay/Sustain Score—and it ranked as second best on both Onset-only and Onset-Offset metrics. Also, when compared to previous genetic algorithm approaches, the proposed system brings significant improvements on both computation time and quality of the results.

Future work will focus on demonstrating the capabilities of the proposed system on other kinds of pitched instruments (e.g., string ensemble, woodwind, etc.). Also, we intend to exploit the parallelization capabilities of the proposed system to reduce its computing time.

Appendix A
Get Possible Notes

The function GetPossibleNotes generates a list of possible notes and is described on Algorithm 3. This function analyzes the power spectra of the acoustic signal and then returns the list of possible notes: for each frame n the biggest α peaks are selected, and then, for each peak, the corresponding MIDI notes are added to the possible notes list. α was empirically set to 10. The MIDI notes corresponding to the frequency bin bin are those which verify the following equation:

$$bin = \text{Round}\left(\frac{freq_{note}}{resolution}\right) \tag{7}$$

where the $freq_{note}$ is the frequency of a MIDI note:

$$freq_{note} = 6.875 \times 2^{\frac{3+note}{12}} \tag{8}$$

and where the $resolution$ is the frequency bin resolution:

$$resolution = \frac{Fs}{N}. \tag{9}$$

Algorithm 3 Get Possible Notes algorithm.

Require spectrum X, hop size R
for each frame $X(n)$ **do**
  eliminate all non-peak values from current frame
  for each peak p in $X(n)$ **do** {p is a 2-order tuple *(magnitude, bin)*}
    $P \leftarrow P + p$ {P is the list of peaks}
  end for
  sort P according to the power magnitude
  $P \leftarrow$ first α elements of P
  for each peak p in P **do**
    $beginNote \leftarrow$ first MIDI note belonging to bin $p.bin$
    $endNote \leftarrow$ first MIDI note belonging to bin $p.bin + 1$
    for $i = beginNote$ to $endNote - 1$ **do**
      $newNote.note = i$
      $newNote.start = n \times R$
      $newNote.duration = R$
      for each $note$ in $possibleNotes$ **do**
        if $note$ overlaps with $newNote$ **then**
          $note.duration \leftarrow note.duration + R$
        else
          $possibleNotes \leftarrow possibleNotes + newNote$
        end if
      end for
    end for
  end for
end for
return $possibleNotes$

The peak-picking algorithm considers as a peak a local maximum, and only takes into account the previous and next bins of the current bin. The algorithm does not perform any kind of interpolation or spectrum whitening.

ACKNOWLEDGMENT

The authors would like to thank to V. Emiya for sharing his results, N. Fonseca for the Hybrid Decay/Sustain Score framework, to all the researchers that shared their algorithms so that we could perform the reported comparisons, and to P. Chavez for his hard work on configuring the blade machines so that they could perform all the tests.

REFERENCES

[1] V. C. Shields, "Separation of added speech signals by digital comb filtering," Ph.D. dissertation, Mass. Inst. of Technol., Cambridge, 1970.
[2] J. A. Moorer, "On the transcription of musical sound by computer," *Comput. Music J.*, vol. 1, no. 4, pp. 32–38, 1977.
[3] M. Piszczalski and B. A. Galler, "Automatic music transcription," *Comput. Music J.*, vol. 1, no. 4, pp. 24–31, 1977.
[4] C. Yeh, A. Roebel, and X. Rodet, "Multiple fundamental frequency estimation and polyphony inference of polyphonic music signals," *IEEE Trans. Audio, Speech, Lang. Process.* vol. 18, no. 6, pp. 1116–1126, Aug. 2010.
[5] D. E. Goldberg, *Genetic Algorithms Search, Optimization, Machine Learning*. Boston, MA: Addison-Wesley, Jan. 1989.
[6] A. Klapuri, "Multiplitch analysis of polyphonic music and speech signals using an auditory model," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 2, pp. 255–264, Feb. 2008.
[7] K. D. Martin, A blackboard system for automatic transcription of simple polyphonic music MIT Media Lab, Perceptual Computing Section, Tech. Rep. 385, Jul. 1996, Tech. Rep..
[8] K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka, "Organization of hierarchical perceptual sounds: Music scene analysis with autonomous processing modules and a quantitative information integration mechanism," in *Proc. Int. Joint Conf. Artif. Intell.*, 1995, pp. 158–164.
[9] P. J. Walmsley, S. J. Godsill, and P. J. W. Rayner, "Bayesian graphical models for polyphonic pitch tracking," in *In Diderot Forum*. San Francisco, CA: Morgan Kaufmann, 1999, pp. 1–26.
[10] P. J. Walmsley, S. J. Godsill, and P. J. W. Rayner, "Polyphonic pitch tracking using joint bayesian estimation of multiple frame parameters," in *Proc. IEEE Workshop Audio Acoust.*, New Paltz, NY, 1999, pp. 119–122.
[11] C. Raphael, "Automatic transcription of piano music," in *Proc. ISMIR '02*, 2002.
[12] C. F. , "Automatic harmonic description of musical signals using schema-based chord decomposition," *J. New Music Res.*, vol. 28, pp. 310–333(24), Dec. 1999.
[13] M. Marolt, "Networks of adaptive oscillators for partial tracking and transcription of music recordings," *J. New Music Res.*, vol. 33, pp. 49–59(11), Mar. 01, 2004.
[14] L. Ortiz-Berenguer, F. Casajus-Quiros, and S. Torres-Guijarro, "Multiple piano note identification using a spectral matching method with derived patterns," *J. Audio Eng. Soc.*, vol. 53, no. 1/2, pp. 32–43, Jan./Feb. 2005.
[15] G. Garcia, "A genetic search technique for polyphonic pitch detection," in *Proc. Int. Comput. Music Conf. (ICMC)*, Havana, Cuba, Sep. 2001.
[16] D. Lu, Automatic music transcription using genetic algorithms and electronic synthesis, Computer Science Undergraduate Research, Univ. of Rochester. Rochester, NY.
[17] The Complete MIDI 1.0 Detailed Specification, MIDI Manufacturers Association, Sep. 1995.
[18] G. Reis and F. Fernandez, "Electronic synthesis using genetic algorithms for automatic music transcription," in *Proc. GECCO '07: Proc. 9th Annu. Conf. Genetic and Evolut. Comput.*, New York, 2007, pp. 1959–1966, ACM Press.
[19] G. Reis, N. Fonseca, and F. Fernandez, "Genetic algorithm approach to polyphonic music transcription," in *Proc. WISP '07 IEEE Int. Symp. Intell. Signal Process.*, 2007, pp. 321–326.
[20] G. Reis, N. Fonseca, F. Fernandez, and A. Ferreira, "A genetic algorithm approach with harmonic structure evolution for polyphonic music transcription," in *Proc. 8th IEEE Int. Symposium on Signal Processing and Information Technology*, Dec. 2008, pp. 491–496.
[21] G. Reis, F. Fernandez, and A. Ferreira, "Transcripción de música multitimbre mediante algoritmos genéticos," in *Proc. MAEB 2009 VI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*, Feb. 2009.
[22] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2003.
[23] L. G. Martins, "A computational framework for sound segregation music signals," Ph.D. dissertation, Univ. of Porto, Porto, Portugal, Sep. 2008.
[24] S. Dixon, "Onset detection revisited," in *Proc. 9th Int. Conf. Digital Audio Effects*, 2006, pp. 133–137.
[25] H. Fletcher, E. D. Blackham, and R. Stratton, "Quality of piano tones," *J. Acoust. Soc. Amer.* vol. 34, no. 6, pp. 749–761, 1962 [Online]. Available: http://link.aip.org/link/?JAS/34/749/1
[26] V. Emiya, R. Badeau, and B. David, "Automatic transcription of piano music based on HMM tracking of jointly-estimated pitches," in *Proc. Eur. Conf. Signal Process. (EUSIPCO)*, Aug. 2008.
[27] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 6, pp. 1643–1654, Aug. 2010.
[28] N. H. Fletcher and T. D. Rossing, *The physics of musical instruments/ Neville H. Fletcher, Thomas D. Rossing*, 2nd ed. New York: Springer, 1998.
[29] B. Wei and J. Gibson, "Comparison of distance measures discrete spectral modeling," in *Proc. 9th DSP Workshop and 1st Signal Process. Education Workshop*, 2000.
[30] J. H. Holland, *Adaptation Natural and Artificial Systems*. Ann Arbor, MI: Univ. of Michigan Press, 1975.

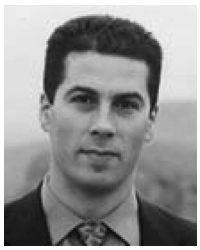
[31] B. Bogert, M. Healy, and J. Tukey, "The quefrency alanysis of time series for echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum and Saphe Cracking," in *Proc. Symp. Time Series Anal.*, 1963, pp. 209–243.
[32] J. H. Holland, *Adaptation Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, Artificial Intelligence*. Cambridge, MA: MIT Press, April 1992.
[33] W. Hart, N. Krasnogor, and J. Smith, "Memetic Evolutionary Algorithms," in *Recent Advances Memetic Algorithms*. New York: Springer, 2004.
[34] V. Emiya, Transcription automatique de la musique de piano, THESE, Télécom ParisTech, Oct. 2008 [Online]. Available: http://hal.inria.fr/pastel-00004867/en
[35] V. Nostrand, "Harmonic Mean," in *Mathematics of Statistics*, ser. 3. Princeton, NJ: , 1962, vol. 1, pp. 57–58.
[36] A. T. Cemgil, H. J. Kappen, and D. Barber, "A generative model for music transcription," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 2, pp. 679–694, Mar. 2006.
[37] E. Vincent, N. Bertin, and R. Badeau, "Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription," in *Proc. Int. Conf. Audio Speech Signal Process. (ICASSP)*, Las Vegas, NV, Mar. 2008.
[38] N. Bertin, R. Badeau, and G. Richard, "Blind signal decompositions for automatic transcription of polyphonic music: NMF and K-SVD on the benchmark," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP*, 2007, vol. 1, pp. 65–68.
[39] M. Marolt, "A connectionist approach to automatic transcription of polyphonic piano music," *IEEE Trans. Multimedia*, vol. 6, no. 3, pp. 439–449, Jun. 2004.
[40] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 3, pp. 528–537, Mar. 2010.
[41] M. Ryynänen and A. Klapuri, "Polyphonic music transcription using note event modeling," in *Proc. IEEE Workshop Applicat. Signal Process. Audio Acoust.*, New Paltz, NY, Oct. 2005, pp. 319–322.
[42] J. S. Downie, "The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research," *Acoust. Sci. Technol.*, vol. 29, no. 4, pp. 247–255, 2008.
[43] J. Downie, A. Ehmann, M. Bay, and M. Jones, "The music information retrieval evaluation exchange: Some observations and insights," in *Advances Music Information Retrieval*, ser. Studies Computational Intelligence, Z. Ras and A. Wieczorkowska, Eds. Berlin/Heidelberg, Germany: Springer, 2010, vol. 274, pp. 93–115.
[44] N. Fonseca and A. Ferreira, "Measuring music transcription results based on a hybrid decay/susta evaluation," in *Proc. ESCOM '09—7th Triennial Conf. Eur. Soc. for the Cognitive Sci. of Music*, Finland, 2010.

Gustavo Reis (M'10) received the Diploma degree in information systems and computer science from the School of Technology and Management, Polytechnic Institute of Leiria, Leiria, Portugal, in 2004 and in 2007, respectively, and the M.Sc. degree in information technology (genetic algorithms for polyphonic music transcription) from the University of Extremadura, Badajoz, Spain. He is currently pursuing the Ph.D. degree in University of Extremadura.

In 2004, he became Lecturer for the Computer Science Department of School of Technology and Management, Leiria. His research interest focuses on evolutionary computation, artificial intelligence, and music signal analysis and processing.

Francisco Fernández de Vega (M'04–SM'11) received the Ph.D. degree from the University of Extremadura, Badajoz, Spain, in 2002.

He is currently an Associate Professor at the Computers and Communication Technology Department, Centro Universitario de Mérida, University of Extremadura. His main interests include computational intelligence and parallel and distributed computing.



Aníbal Ferreira (M'92) was born in Penafiel, Portugal, in 1963. He graduated in electrical and computer engineering from the School of Engineering, University of Porto (Faculdade de Engenharia da Universidade do Porto—FEUP), Porto, Portugal, in 1988 and received the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Porto in 1992 and 1998, respectively.

From 1995 to 1998, he was a Lecturer Assistant at FEUP, University of Porto, and since then he has been a Professor Assistant in the Department of Electrical and Computer Engineering, FEUP. In 1988, he had his first international research experience joining the VLSI Group at Philips Research Labs in Eindhoven (NatLab), The Netherlands, in the context of a project addressing automatic silicon compilation. In 1990/1991 and later in 1993, he joined the Signal Processing Research Department headed by Dr. Nikil Jayant, at AT&T Bell Laboratories, Murray Hill, NJ, where he coauthored the development of the AT&T Perceptual Audio Coder (PAC). He has been affiliated with INESC Porto since 1987. Currently, he is the head of the Audio Processing Group coordinating both undergraduate and postgraduate research work. He has participated in several European research projects and has been the coordinator of four Portuguese research projects, one of which has supported his participation in the (ISO/IEC) MPEG-4 Audio standardization activities. He has published over 20 papers and holds two patents. His interests include psychoacoustics, audio analysis and coding, multirate filter banks and algorithms for real-time digital audio applications.

Dr. Ferreira is a recipient of the 1998 IBM Scientific Prize that distinguishes relevant research work on applied computing. He is a member of the AES and ASA.