GENETIC ALGORITHM APPROACH TO POLYPHONIC MUSIC TRANSCRIPTION FOR MIREX 2008

Gustavo Reis
Polytechnic Institute of Leiria, Portugal
gustavo.reis@estg.ipleiria.pt

Francisco Fernandez
University of Extremadura, Spain
fcofdez@unex.es

Anibal Ferreira
University of Porto, Portugal
ajf@fe.up.pt

ABSTRACT

This paper describes our method, submitted to MIREX 2008 task "Multiple Fundamental Frequency Estimation & Tracking". This task restricted the problem of Multiple F0 Estimation and Tracking to three cases: i) Estimate active fundamental frequencies on a frame-by-frame basis; ii) Track note contours on a continuous time basis (as in audio-to-midi); iii) Track timbre on a continuous time basis. The presented method is the first genetic algorithm approach to multi-timbral music transcription and is based on a previously published genetic algorithm approach for automatic music transcription originally designed to transcribe piano songs. This algorithm was adopted to support different timbres.

1 INTRODUCTION

Music Transcription is the process of extracting the musical score of an acoustical music signal. In *Automatic Music Transcription* musical notes are extracted from the musical signal by a computer.

Although traditional methods to Polyphonic Music Transcription rely mainly on digital signal processing techniques, Music Transcription can also be addressed as a search space problem, where the goal is to find the notes of the musical signal. Search space approaches to this problem may seem impracticable due to the huge (almost infinite) size of the search space. Nevertheless Genetic Algorithms[1] have proven to be an excellent tool in these problems since they only need to use a small subset of the entire search-space to find good solutions. This kind of approach can also be used as a mean to improve other existing techniques, using their results as the starting point.

The submitted method for polyphonic music transcription is based on a previously published Genetic Algorithm approach[2], which was designed for polyphonic piano transcriptions. For the task "Multiple Fundamental Frequency Estimation & Tracking", the algorithm was adopted to support different timbres. Since the additional features of the algorithm exponentially increase the the search space of the problem, the Gene Fragment Competition operator[3] was implemented to improve the results as well as other optimization and fragmentation techniques techniques, which will be described on the following sections.

2 METHOD DESCRIPTION

Our method will be briefly explained here. For more details and information please see [2].

2.1 Avoid Harmonic Overfitting

In a previous proposed Genetic Algorithm approach to Polyphonic Music Transcription, Reis et al.[2] noticed that the genetic algorithm tends to create additional notes (with lower amplitudes) in harmonic locations of the original notes to overcome the timbre differences between the internal samples and original piano sounds.

In order to solve this problem we have created harmonic gains, which boost or cut the value of the first 20 harmonic peaks. This acts almost like an equalizer, but instead of operating on fixed frequency bands, it operates on each note harmonic. From the implementation point of view, this is not done with real filters, but by changing the values of the FFT bins that correspond to the note harmonic locations.

This means, that each chromosome, besides having a sequence of note events as their candidate solution to the problem, also includes additional parameters to help the internal synthesizers to get a timbre more similar with the original instrument (see Figure 1). The gain of the F0 of the note is always 1.

2.2 Inharmonicy Evolution

Sometimes the harmonics are not located in integer multiples of the Fundamental Frequency. Those harmonics are often shifted some bins to the left or to right of the real multiple corresponding frequency bin. To solve this problem, the amount of shifting for each harmonic in the harmonic structure was also encoded within the chromosomes, among with the harmonic structure (see Figure 1). This way each Individual had his own set of synthesizers with complete

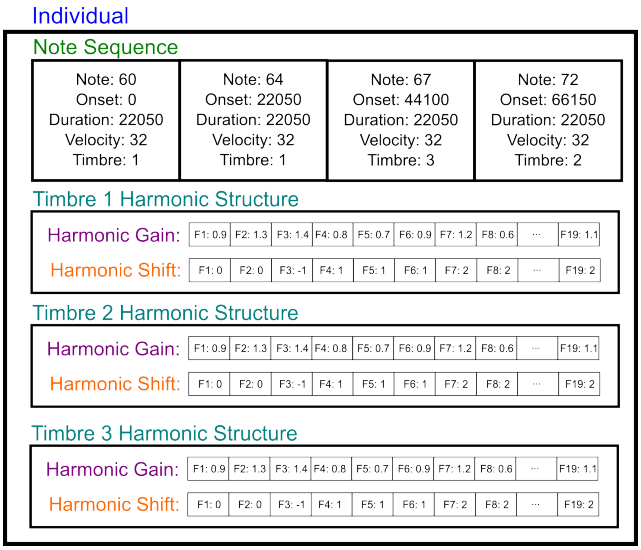

Figure 1. Encoding of the Individual with the Harmonic Structure.

evolving harmonic structures towards the original synthesizers and also with evolving notes towards the original song's notes. The shift of the Fundamental Frequency of the note - F0 - is always 0.

2.3 Multi-Timbre Support

Since the evaluation files of the task "Multiple Frequency Estimation and Tracking" have several and different instruments: piano, bass, acoustic guitar, flute, violin, cello, viola, clarinet, oboe, horn and bassoon, we decided to create an internal synthesizer for each instrument. The information about which synthesizer plays each note was encoded as a note event parameter (see Figure 1).

The internal synthesizers were made from samples of the Musical Instrument Sound Database - RWC Music Database [4]. The harmonic structure of each synthesizer was encoded inside the individuals genome (see Figure 1) to avoid the harmonic overfitting in each instrument.

2.4 New Recombination and Mutation Operators

By introducing the harmonic gains, the harmonic shifts also the timbre information in the individuals genotype, recombination and mutation must support those additional features:

2.4.1 Recombination

The recombination operator still splits the note events by applying a random point of cut in time as it already did. Two more random points of cut are used for each harmonic structure: one for splitting the harmonic series and another for splitting the harmonic shifting.

2.4.2 Mutation

Regarding mutation operator, three new mutations were created: one that changes and harmonic gain up to ± 0.50, another which changes the harmonic overfitting up to $\pm 3 bins$ and one mutation that changes the instrument of a note.

2.5 Note discard

Note discard was also implemented to avoid the harmonic overfitting. Note discart is based on the idea that within the note local range, most notes have similar dynamics. Considering that each note has a dynamic scale between 1 and 128 (as in MIDI), this feature will discard all notes present at transcription that have a dynamic difference of 20 between its note dynamics and the maximum value of dynamics of other notes existing during the note duration.

2.6 Dynamic Range

Since harmonic overfitting can also be caused by noise, weak harmonics or even harmonics frequency neighborhood, a dynamic range feature was also implemented, where each time frame the FFT bin with the highest value is used as reference, and all bins with values 40dBs below this reference have their values are set to 0.

2.7 Fitness Function - Individual Evaluation

To evaluate an individual some kind of comparison between his synthesized stream and the original stream must be done: both are cut in time frames with 4096 samples (fs = 44.100kHz) and an overlapping of 75%, a Hanning window is applied to decrease spectrum leakage and the fitness values are based on the difference between the FFT bins over time (Equation 1). Additional work has been done in exploring other fitness domains (like: FFT with logarithmic scale, Cepstrum, SACF, ACF, etc.), but to the moment, linear FFT differences presented the best results so far.

$$Fitness = \sum_{t=0}^{tmax} \sum_{f=27.5Hz}^{\frac{fs}{2}} \frac{||O(t,f)| - |X(t,f)||}{f} \quad (1)$$

Fitness value is computed from frame slot 0 to $tmax$, traversing all time from the beginning to the end. The lower part of the frequency spectrum is limited to the fundamental frequency of the first note of the piano's keyboard, i.e., the fitness function is created using the difference of the FFT bins in the frequency range between the lowest piano's keyboard note (from MIDI-note 21 - 27,5 Hz - to 22.100 Hz - the Nyquist frequency of 44.100 kHz).

Although Genetic Algorithms usually consider higher fitness values for the best individuals, our approach considers

	CL1	CL2	DRD	EBD1	EBD2	EOS	MG	PI1	PI2	RFF1	RFF2	RK	VBB	YRC1	YRC2
Accuracy	0.36	0.49	0.5	0.45	0.45	0.47	0.43	0.6	0.62	0.21	0.18	0.61	0.54	0.62	0.67
Accuracy Chroma	0.4	0.52	0.56	0.5	0.5	0.55	0.5	0.64	0.66	0.27	0.23	0.66	0.57	0.66	0.69

Table 1. Overall results task1

the opposite since we are trying to minimize the error between the original audio stream and the individual's audio stream.

2.7.1 Frequency Normalization

To avoid the increase of impact of higher octaves, since FFT bins are not equally distributed by all octaves (higher octaves are spread over much more FFT bins than the lower octaves), it is important to create a frequency normalization process. The division by f in Equation 1 acts as a frequency normalization. The $|O(t, f)|$ is the magnitude of frequency f at time frame t in the source audio signal, and $|X(t, f)|$ is the same for each individual's audio signal.

3 IMPLEMENTATION

In order to decrease the complexity of the problem, several divide-and-conquer techniques were applied: it is easier to transcribe one second of music rather than transcribing an whole song. Each 30 second music fragment is splitted into 20 fragments of 1.5 seconds, the algorithm transcribes each fragment and then merge each fragment solution into a global solution (similarly with the Parisian approach[5] and with the system proposed by Fonseca[6]). Finally, in order to solve the frontier issues between fragmens, an hill-climber is applied on the notes near the fragment borders, as proposed by Fonseca[6].

To increase the performance, the algorithm employed in each fragment is based on the Gene Fragment Competion[3] approach. With the inclusion of the harmonic structures inside the individuals genome, we had to adopt the Gene Fragment Competition operator to work with non-decomposable data such has the harmonic structure of each synthesizer. Therefore, instead of creating just one individual with the best part of both parents, our new operator creates two new individuals. Both have the same notes, which are the best parts of each parent, and different harmonic structures, generated in the same way the classic "1 point of cut" crossover operator[1] works.

4 RESULTS

The main target of our submission was the Piano Transcription subtask, since this algorithm was initially designed for automatic transcription of piano music. Unfortunately the Piano Transcription subtask was cancelled due to lack of participants (< 3). The results presented here are from the adopted version of the algorithm to the multi-timbral music (RFF1 and RFF2) for both task 1 and task 2.

	Precision	Recall	Accuracy	Etot	Esubs	Emiss	Efa
PI2	0.88	0.69	0.66	0.36	0.05	0.26	0.05
PI1	0.88	0.67	0.64	0.38	0.06	0.28	0.05
EBD2	0.79	0.55	0.5	0.54	0.09	0.36	0.09
YRC2	0.77	0.81	0.69	0.4	0.08	0.13	0.19
EBD1	0.76	0.56	0.5	0.56	0.1	0.34	0.13
VBB	0.75	0.65	0.57	0.51	0.09	0.27	0.16
RK	0.75	0.77	0.66	0.41	0.1	0.13	0.18
YRC1	0.74	0.79	0.66	0.43	0.09	0.13	0.22
CL2	0.72	0.6	0.52	0.56	0.11	0.29	0.16
EOS	0.7	0.64	0.55	0.55	0.11	0.24	0.19
RFF1	0.64	0.29	0.27	0.79	0.12	0.6	0.07
RFF2	0.63	0.24	0.23	0.81	0.11	0.66	0.05
DRD	0.61	0.74	0.56	0.65	0.16	0.1	0.39
MG	0.56	0.67	0.5	0.71	0.2	0.13	0.39
CL1	0.4	0.84	0.4	1.61	0.16	0	1.44

Table 2. Detailed chroma results of task1

Table 1 shows the overall summary results of the task1. Although our algorithm do not performs very well, Table 2 shows that the precision of the proposed approach is above 60%, which means that around 63% of the pitches found by our system are correct. Our system fails because only 29% of the original pitches (recall value) were found and correctly classified by our system.

Table 3 shows the overall summary results of the task2.

5 CONCLUSIONS

In this extended abstract we presented the first genetic algorithm approach to multi-timbral music transcription. Although the algorithm can cope well with polyphonic piano transcriptions, its adaptation to multi-timbral musical is on an early state and needs more work to improve the technique. Additional operators (for instance: mutation operators) need to be tested and implemented for a more robust transcription system.

In order to reduce the computational time, the algorithms only performed 150 generations over each fragment instead of the regular 1000 generations. We strongly believe that this makes all the difference about obtained results.

	EBD1	EBD2	EOS	PI1	PI2	RFF1	RFF2	RK	VBB	YRC	ZR1	ZR2	ZR3
Ave. F-measure (Onset-Offset)	0.176	0.158	0.236	0.247	0.192	0.028	0.032	0.337	0.197	0.355	0.261	0.263	0.278
Ave. F-measure (Onset-Offset Chroma)	0.189	0.169	0.268	0.251	0.195	0.038	0.042	0.352	0.208	0.362	0.297	0.3	0.313
Ave. F-measure (Onset Only)	0.417	0.384	0.503	0.47	0.396	0.14	0.132	0.614	0.521	0.552	0.518	0.52	0.53
Ave. F-measure (Onset Only Chroma)	0.47	0.429	0.561	0.52	0.446	0.177	0.168	0.655	0.547	0.576	0.575	0.577	0.586

Table 3. Overall results of task2

6 REFERENCES

[1] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, January 1989.

[2] Gustavo Reis, Nuno Fonseca, and Francisco Fernandez. Genetic algorithm approach to polyphonic music transcription. *Proceedings of WISP 2007 IEEE International Symposium on Intelligent Signal Processing*, pages 321–326, 2007.

[3] Gustavo Reis, Nuno Fonseca, Francisco Fernndez de Vega, and Anibal Ferreira. Hybrid genetic algorithm based on gene fragment competition for polyphonic music transcription. In *EvoWorkshops*, volume 4974 of *Lecture Notes in Computer Science*, pages 305–314. Springer, 2008.

[4] Masataka Goto and Takuichi Nishimura. Rwc music database: Music genre database and musical instrument sound database. pages 229–230, 2003.

[5] Pierre Collet, Evelyne Lutton, Marc Schoenauer, P. Collet, E. Lutton, F. Raynal, and M. Schoenauer. Polar ifs + parisian genetic programming = efficient ifs inverse problem solving. *Genet. Programm. Evolvable Mach. J*, 1:361, 2000.

[6] Nuno Fonseca. Fragmentation and frontier evolution for genetic algorithms optimization in music transcription. In *Iberamia 2008*, Lecture Notes in Computer Science. To appear, pages 305–314. Springer.