Pitch Estimation Enhancement Employing Neural Network- Based Music Prediction

Marek Szczerba & Andrzej Czyżewski
Sound & Vision Engineering Department
Technical University of Gdańsk
ul. Narutowicza 11/12, PL-80952 Gdańsk, Poland
tel. +48 (58) 347 1301
marek@akustyka.com

ABSTRACT

In this paper a new method for pitch estimation enhancement was presented. Pitch estimation methods are widely used for extracting musical data from digital signal. A brief review of these methods is included in the paper. However, since processed signal may contain noise and distortions, the estimation results can be erroneous. The proposed method was developed in order to override disadvantages of standard pitch estimation algorithms.

The new -approach is based on both pitch estimation in terms of signal processing and pitch prediction based on musical knowledge modeling. First, signal is partitioned into segments roughly analogous to consecutive notes. Thereafter, for each segment an autocorrelation function is calculated. Autocorrelation function values are then altered using pitch predictor output. A music predictor based on artificial neural networks was introduced for this task. The description of the proposed pitch estimation enhancement method is included and some details concerning music prediction are discussed in the paper.

1. INTRODUCTION

Pitch estimation is one of the mostly investigated and developed areas of signal processing [2][4]. Pitch estimation methods are widely used for music transcription – acquisition of musical data from digital signal and for music instrument timbre parameterization [7]. These methods and their music transcription performance are briefly reviewed in the paper. However, pitch estimation methods applied for automatic music transcription from acoustic signal cause numerous processing errors. There are two main types of such errors: transient errors and octave errors [1]. Conversely, in many of such cases human listeners can determine pitch of signal evidently.

Above remarks state the motivation for the presented work. Revising psychophysiologic constraints [3] two main resultant assumptions were introduced: human listeners integrate pitch within the duration of singular note (i.e. between transients) and they can predict pitch of consecutive notes, so they are able to correct music information in spite of noisy or distorted signal. These assumptions are fundamental for the new pitch estimation method. Signal is processed within segments roughly equivalent to consecutive notes (pitch integration) and then predicted for each note (pitch prediction).

2. PITCH ESTIMATION METHODS AND THEIR PERFORMANCE

There are numerous methods of pitch estimation developed by a number of researchers. These methods are mainly categorized in terms of functional domain: there are time, frequency, time-frequency and cepstrum methods [4].

Pitch estimate can be evaluated in time-domain by identifying periodicity features within the sound wave. The most commonly used time-domain pitch estimation methods include: threshold-crossing analysis methods, parallel processing method, envelope analysis, autocorrelation and AMDF methods.

Frequency-domain pitch estimation can be evaluated by identifying certain features within the short-term spectra of musical signal. Block diagram of a general frequency-domain pitch estimator is shown in Figure 1.

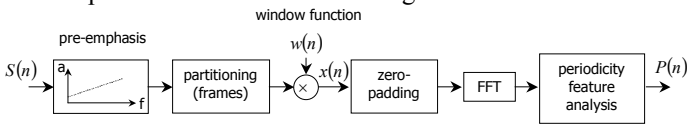

Figure 1. General frequency-domain pitch estimator diagram.

Frequency-domain pitch estimators include the following: Schroeder's histogram and spectral compression methods, comb-filter method and Beauchamp's method.

Pitch can be also estimated using some time-frequency methods as sub-band processing based on Meddis-Hewitt model [6] and McAulay-Quatieri method. Another, yet popular method is estimation of pitch in cepstral domain.

All of the above pitch estimation methods were implemented and their performance has been tested using numerous signals. The signals were generated using common wavetable synthesizer (Sound Blaster PCI card). The excerpts from real recordings (oboe and violin solo) were also used for the experiments. All files were monophonic, sampled using 22.05 kHz sample rate and 16-bit resolution.

Pitch estimator's performance was evaluated using the following measures:

$$e = \frac{1}{N}\sum_{n=1}^{N}\left|\hat{f}_0 - f_0\right| \quad (1)$$

where: $\hat{f}_0$ is the estimated and f_0 - the real fundamental frequency,

$$df_{pr} = \frac{100}{N}\sum_{n=1}^{N} r(n) \quad (2)$$

where:

$$r(n) = \begin{cases} 1 & \text{dla} \quad f_0\sqrt[12]{2^{-1}} < \hat{f}_0 < f_0\sqrt[12]{2} \\ 0 & \text{dla} \quad \hat{f}_0 < f_0\sqrt[12]{2^{-1}} \vee \hat{f}_0 > f_0\sqrt[12]{2} \end{cases} \quad (3)$$

giving the percentage of correctly estimated fundamental frequencies with semitone precision.

2.1 Experiments

A set of synthesized signals was prepared for the experiments. These include singular notes as well as music phrase using synthesized timbres of flute, oboe, organ and piano. Pitch estimation methods' performance was also verified using real music recordings. Two excerpts were used for the experiments: an excerpt from Fantasia no. 1 from "12 Fantasies for Oboe Solo" by G. Ph. Telemann [14] and an excerpt from Capriccio A-minor no. 24 from "24 Capriccios for Violin" by N. Paganini [11].

Experiments were performed using the following pitch estimation methods: autocorrelation, comb-filter, cepstral and Meddis-Hewitt model. Performance was evaluated using *e* (1) and df_{pr} (2) measures relatively to the generated reference dataset – phrase definition based on music notation, verified by an expert.

Experiments concerning the autocorrelation method confirm that weighting and attenuation significantly improve pitch estimation performance. The highest performance rating (df_{pr}) for the weighted and attenuated autocorrelation was around 86% whereas for standard method – 68%. Comb-filter method was evaluated for a variable fading factor. Performance rating was slightly lower than for the weighted and attenuated autocorrelation method. However, since obtained optimal fading factor was invariable, the comb-filter method seems to be more universal and not to require adaptation accordingly to signal's characteristics.

Revising cepstral pitch estimator's performance based on the tests with the synthesized signals an additional cepstrum attenuator was introduced. The attenuated cepstrum is defined as:

$$C_M(n) = \log(n+1)^k C(n) \quad (4)$$

where $C(n)$ denotes the cepstrum and k is an attenuation rate. The accuracy rating achieved for the cepstral pitch estimator was lower than for the autocorrelation and comb-filter methods – around 75%.

Tests for the Meddis-Hewitt model-based pitch estimator proved it's high accuracy and comprehensiveness. An absolute rating was slightly lower than for autocorrelation method (around 83%), though Meddis-Hewitt model-based estimator does not require adaptation for individual signal's characteristics as in case of the other methods. It should be noted however, that since Meddis-Hewitt model-based pitch estimator performs complex signal analysis in sub-bands it requires high computing effort.

For all the methods implemented pitch estimation inaccuracies were examined. There are two main error categories: errors caused by transient noises and distortions occurring between notes (transient errors) and errors originated from temporal harmonic structure inconsistencies causing octave shifts (octave errors).

As noted in the Introduction two main psychoacoustic merits for the pitch estimation: pitch integration through time and pitch prediction were employed to reduce pitch estimation errors. The details concerning introduced pitch estimation support method are presented in the further sections of this paper.

3. PITCH INTEGRATION

Based on the evaluation of several pitch estimation algorithms presented above a new method incorporating pitch integration through time was elaborated. The autocorrelation method was selected as a fundamental processing routine for pitch estimation.

3.1 Signal Segmentation

The aim of the signal segmentation method is to divide a monophonic music signal into segments roughly corresponding to individual notes within an excerpt.

Conventional methods perform segmentation upon amplitude envelope analysis. Such solution can be efficient in case of signals consisting of clearly separated subsequent notes. However in case of real music recordings including legato articulation and significant reverberation an alternate solution should be formulated.

It has been observed, that during transient portions of signal maximum values of autocorrelation function can decrease significantly. Also signal amplitude can drop off momentarily. Amplitude variation can be analyzed in terms of a following function:

$$\frac{1}{M}\sum_{m=1}^{M}\log|x(m)| \quad (5)$$

where M is the period of the lowest prospective fundamental frequency period. Consequently a partition function $s_g(n)$ was commenced as follows:

$$s_g(n) = \max_i \rho_n(i)\left[1 + \frac{1}{wM}\sum_{m=1}^{M}\log|x(nh+m)|\right] \quad (6)$$

where $\rho_n(i)$ is a weighed and attenuated autocorrelation function and w is a scaling factor (according to the initial experiments scaling factor was set to $w = 10$) and h is a leap size of the analysis. Transients location within a signal can be estimated upon the position of minimums of the $s_g(n)$ function.

However, in case of legato articulation and significant reverberation signal energy as well as maximum value of the autocorrelation function may not fade out within transient. Therefore, the segmentation routine based on

$s_g(n)$ function may not work correctly. During steady sections of a signal, local autocorrelation peaks do not alter considerably. However within transients some autocorrelation peaks fade away whereas the new ones appear. Consequently, a second segmentation function $s_h(n)$ was introduced:

$$s_h(n) = \sum_{l=1}^{L} \rho_n(l_{\max}) \rho_{n-1}(l_{\max}) \tag{7}$$

where $l_{\max}$ denotes l-th in terms of amplitude local maximum of the autocorrelation function $\rho_n(i)$.

Both segmentation functions can be concerned complementary. Therefore, basing on initial experiments a following comprehensive segmentation function was established:

$$s_f(n) = \sqrt{s_g^2(n) + [q s_h(n)]^2} \tag{8}$$

where q is a scaling factor set tentatively to $q = 4$.

An illustration of the segmentation function $s_f(n)$ is presented in Figure 2.

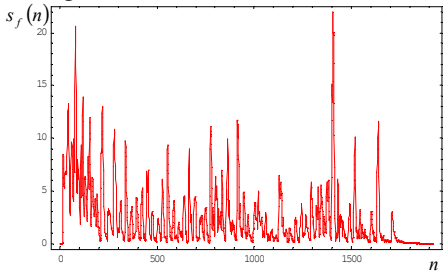

Figure 2. $s_f(n)$ function plot.

Accordingly, segmentation may be executed upon location of minima within $s_f(n)$ function. It can be assumed that transients cannot occur densely through time. Therefore segmentation points positioned in distance less than 1000 samples from the previous ones are eliminated.

Initial experiments proved the method's capability to partition a musical excerpt into segments roughly equivalent to individual subsequent notes. It has also been observed that the introduced segmentation method may work incorrectly in case of glissando-like pitch alterations.

3.2 Pitch Estimation within Segments

It can be assumed, that pitch may alter insignificantly within a designated segment. Consequently, pitch can be estimated for an entire segment. Additionally, partitioning function $s_f(n)$ can be used as a scaling factor for the pitch estimation. Scaling may cause reduction of a transient component effect on the pitch estimate and consequently better pitch estimation performance.

Autocorrelation function within a segment can be estimated as:

$$\hat{\rho}(k) = \frac{1}{s_{k+1} - s_k} \sum_{n=s_k}^{s_{k+1}} s_f(n) \rho(n) \tag{9}$$

where s_k is a k-th segment onset location.

Analogically to the standard autocorrelation method pitch can be then estimated according to the formula:

$$\hat{f}_0(k) = \frac{f_s}{m[\hat{\rho}(k)]} \tag{10}$$

where f_s is a sample rate and $m[\hat{\rho}(k)]$ is an autocorrelation estimate peak location. An illustration of the pitch estimation within segments is shown in Figure 3. The circles indicate pitch estimated incorrectly.

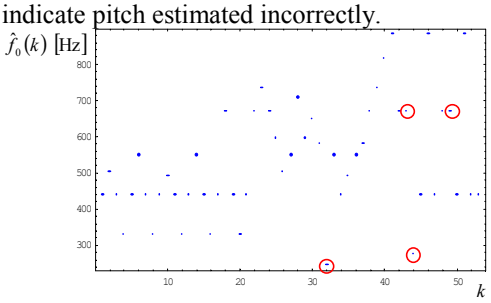

Figure 3. Pitch estimation within segments.

Subsequently, pitch can be estimated precisely for each leap within a segment. Accurate pitch estimates are determined upon locations of local autocorrelation peaks nearest to the global peak location of autocorrelation estimate $\hat{\rho}(k)$.

In Figure 3 incorrect pitch estimates were indicated. An effect of erroneous estimate can also be observed in case of accurate pitch estimation with integration consequently. In the next sections a solution intended to increase pitch estimation accuracy based on pitch prediction is presented.

4. PREDICTIVE SUPPORT FOR MUSIC TRANSCRIPTION

A neural music predictor [13] was developed as a pitch estimation supporting unit. The applied solution is based on the Shannon's concept of predictive data coding, employed by Moradi and others for tests with the English text [9]. A block diagram of neural music predictive encoder is presented in Figure 4.

The data is collected within a buffer. The predictor guesses the next note upon the collected data stored in a buffer. Thereafter a prediction process is repeated until a predicted value match actual note. In case of neural predictor succeeding predictions are emulated by networks activation output values.

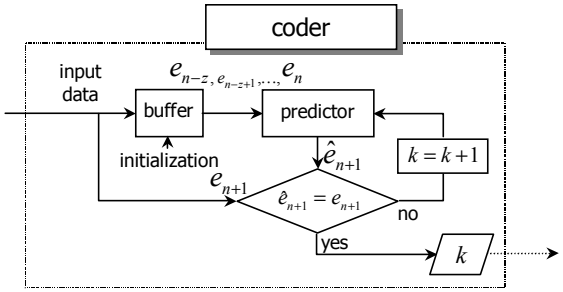

Figure 4. Music predictive encoder.

4.1 Representation and accumulation of data

Music data representation is essential for the prediction performance. Three main pitch representation techniques were examined: binary, modified Hörnel's method and

modified Mozer's method. All representation methods employed represent a relation between the actual pitch and a pitch of a preceding note. In case of binary representation pitch relation is coded using a vector of 27 bits as shown for an example in Tab. 1.

Table 1. An example of binary pitch relation coding.

-octave	interval size (semitones)																										+octave
	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0

Original Hörnel's method was intended for diatonic interval coding [5]. The method was then modified to allow chromatic interval coding. Each pitch relation is then coded using 11 parameters as shown in Table 2.

Table 2. Modified Hörnel's interval representation.

interval - semitones	direction bits	octave bit	interval size representation						
-3	1 0 0	0	0	0	0	0	0.5	1	0
-2	1 0 0	0	0	0	0	0	0	1	0.5
-1	1 0 0	0	0	0	0	0	0	0.5	1
0	0 1 0	0	1	0	0	0	0	0	0.5
+1	0 0 1	0	1	0.5	0	0	0	0	0
+2	0 0 1	0	0.5	1	0	0	0	0	0
+3	0 0 1	0	0	1	0.5	0	0	0	0

Mozer's pitch representation method characterize pitch as an absolute value. Therefore a modification of Mozer's representation was introduced to allow relative representation of interval size. A representation was also enhanced by adding direction parameter and octave bit. Contrary to unipolar binary and modified Hörnel's representation, modified Mozer's representation is bipolar. Intervals are coded as shown in Table 3.

Data can be accumulated using two types of a buffer: fixed-size buffer and fading memory model [15]. In case of fixed-size buffer (where z is a buffer size) data for z events (notes) is accumulated using z individual representation vectors. Fading memory model allows coding of z events using a singular representation vector where actual and faded preceding vector values are accumulated according to the formula:

$$b_n = \sum_{k=1}^{n} e_k r^{n-k} \tag{11}$$

Table 3. Modified Mozer's interval representation.

interval - semitones	direction	octave bit	interval size					
-3	-1	0	+1	+1	+1	−1	−1	−1
-2	-1	0	+1	+1	−1	−1	−1	−1
-1	-1	0	+1	−1	−1	−1	−1	−1
0	0	0	−1	−1	−1	−1	−1	−1
+1	+1	0	−1	−1	−1	−1	−1	+1
+2	+1	0	−1	−1	−1	−1	+1	+1
+3	+1	0	−1	−1	−1	+1	+1	+1

where b_n denotes buffer values for the n-th event, e_k denotes the k-th event and r – a fading factor within the range of (0, 1).

4.2 Predictor implementation

The predictor was implemented using Stuttgart Neural Network Simulator (SNNS) [16]. Three pitch representation methods presented above were examined in terms of prediction efficiency. For the fixed-size buffer its size was set to 5, 10 and 20 samples (music events). For the fading memory model fading factor r was set accordingly to $r = \{0.2; 0.5; 0.8\}$. Expected output of the trained predictor should correspond to a subsequent event e_{k+1} using a selected representation method.

Data for the music prediction experiments were collected from the MIDI database consisting of fugues from *Das Wohltemperierte Klavier* by J. S. Bach. Neural networks were trained using all parts without the uppermost voice. The part left was used for predictor performance tests.

For the experiments a feed-forward neural network model with a single hidden layer was used. Prediction efficiency was tested using the following measures: correctness at the first guess, average number of guesses required for correct prediction and lower and upper F_N entropy bounds given as:

$$\sum_{i=1}^{M} i\left(q_i^N - q_{i+1}^N\right)\log_2 i \le F_N \le \sum_{i=1}^{M} -q_i^N \log_2 i q_i^N \tag{12}$$

where q_i^N denotes frequency of correct answers at the i-th guess and M is the number of possible output patterns.

According to the interval representation characteristics subsequent guesses can be characterized as a vector consisting of all possible output representation vectors ordered in terms of matching measure:

$$o_m = \sum_{i=1}^{N} \left| \hat{e}_{n+1,i} - e_{m,i} \right| \tag{13}$$

where $m = \{1, 2, \ldots, M\}$, i is a representation parameter mark, N denotes number of representation parameters, $\hat{e}_{n+1}$ is a predicted output and e_m is a considered possible output pattern. Number of guesses required for prediction can then be figured out by locating expected output pattern within a set of ordered possible output patterns.

Based on the test results the following pitch prediction characteristics can be deduced. The finest prediction accuracy (correct prediction rate for the first guess more than 0.97) was obtained for the modified Hörnel's representation and fixed-size buffer. For the fading memory type buffer prediction efficiency was significantly lower, however fading memory representation requires considerably less computing power for training and evaluation. Modified Mozer's interval coding is not efficient for music prediction.

4.3 Predictive support for pitch estimation

Evaluated music predictor was implemented to support pitch estimation. Block diagram of the system supporting fundamental frequency estimation within a signal segment $x(k)$ is presented in Figure 5.

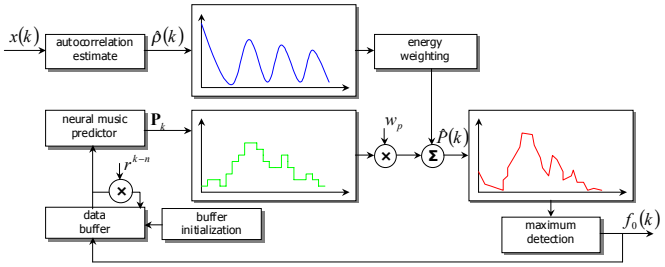

Figure 5. Block diagram of prediction-supported pitch estimator.

Pitch is estimated in a following way. Signal is analysed within segments. For each segment an autocorrelation function is estimated and normalized. Energy weighted autocorrelation function is given as:

$$\hat{\rho}_i^w(k) = \frac{\hat{\rho}_i(k)}{\max_i \hat{\rho}(k)} \tag{14}$$

where $\hat{\rho}_i(k)$ is a autocorrelation function maximum within i-th band and i is an indicator of a semitone sub-band.

At the beginning predictor data buffer is initialised with zero values, thus for initial frames the predictor is not used for supporting pitch estimation. However, pitch values of consecutive notes estimated on the basis of autocorrelation function peak location are stored within a buffer. After a certain amount of data was accumulated within the buffer the predictor starts calculating probable pitch values for the forthcoming notes. A predictor output vector $\mathbf{P}_k$ (see representation methods in section 4.1) is distributed among semitone-wide subbands, scaled using a weighting factor w_p and added to current autocorrelation function values. Pitch $\hat{f}_0(k)$ is then estimated on the basis of peak location of the resultant function. Pitch predictor is implemented to adjust pitch estimate $\hat{f}_0(k)$ within k-th segment.

Initial experiments indicate, that if the autocorrelation function peaks appear at the edge of two adjacent sub-bands pitch estimation may fail. Therefore, pitch predictor output is distributed among the other sub-bands as follows:

$$p_i^m = \sum_{j=0}^{12} \frac{p_{i-j}(k) + p_{i+j}(k)}{j+1} \tag{15}$$

where $p_i(k)$ is the i-th output vector element.

Accordingly, pitch in a k-th segment can be estimated as follows:

$$\hat{P}_i(k) = \frac{\hat{\rho}_i(k)}{\max_i \hat{\rho}(k)} + w_p \sum_{j=0}^{12} \frac{p_{i-j}(k) + p_{i+j}(k)}{j+1} \tag{16}$$

where w_p is a weighting factor. As noted before signal segmentation module may fail in case of smooth inter-note transients. Hence, while using fixed-size type buffer an uncontrolled data shift within a sequence may occur. For this reason in a predictive supported pitch estimator the fading memory model was used. Neural predictor has been trained using parts from "12 Fantasies for Oboe Solo" by G. Ph. Telemann excluding the excerpt used for the experiments.

4.4 Pitch estimation performance

Pitch estimation performance was analyzed using df_{pr} measure (2). A comparison of autocorrelation-based pitch estimator performance with and without signal partitioning is shown in Figure 6.

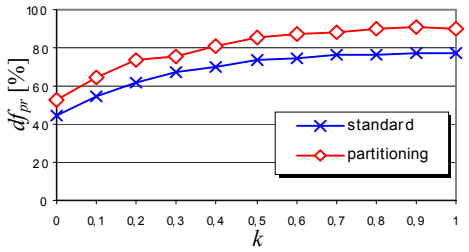

Figure 6. Pitch estimation performance with and without partitioning.

Subsequently, pitch estimation tests for the system incorporating neural music predictor were performed. Experiments were performed for the neural predictors containing one hidden layer of 50 and 100 units (indications 1x50 and 1x100 accordingly) and two hidden layers of 50 units (indication 2x50). Tests were performed for variable fading memory coefficient $r = \{0.2; 0.5; 0.8\}$. Based on the results obtained without using prediction support, linear attenuation function $a(n)$ coefficient was set to $k = 0.8$.

First the correlation between predictor output and the pitch in terms of music context was analyzed. It has proved, that the predictor's output corresponds to music. However, it has been found that the predictor can forecast a different pitch value than the one found in the music. Such a situation is mainly caused by a limited number of patterns within learning sets. In other cases predictor can indicate a few possible solutions. In such a case the choice of a representation method is very important since in some cases (i.e. Mozer's method) it can be unmanageable to decode such information.

It is also very important to avoid influence of prediction errors on the estimate of following notes. In such a case a predictor may be lost and start "composing" it's own data stream. To keep the predictor "on track" lower fading memory model factor values can be used. The predictor has been then connected with the pitch estimation system to perform experiments regarding cooperation between signal analysis and pitch prediction. Experiments were performed using variable values of weighting factor w_p and variable values of fading memory factor.

Initially, the systems behaviour was analyzed using the segments for which the pitch was erroreously estimated without using the predictor. It has been found out, that the predictor output can correspond with the musical contents. Consequently, adding weighted predictor's output to autocorrelation function estimate can alter distorted peaks relevant to the actual pitch.

Accordingly, experiments with longer music signal were performed using an excerpt from the *Fantasia for oboe solo* by G. Ph. Telemann [14]. The maximum gain of pitch prediction accuracy by using pitch prediction was about 2

percent points in terms of df_{pr} measure (90.1% accuracy without and 91.8% accuracy with pitch prediction support). However, system parameters has to be carefully adjusted. In case of improper adjustments (fading memory coefficient, weighting factor w_p etc.) the predictor tends to diminish pitch estimation accuracy. For example, if the w_p factor value is too high, the elaborated system tends to "generate" music coarsely related to the analyzed music excerpt.

5. CONCLUSIONS

Upon the performed experiments and the results presented following conclusions were deduced:

- segmentation of signal using the introduced method can significantly increase pitch estimation accuracy,
- neural prediction support for pitch estimation can furthermore increase accuracy.

Taking into account signal characteristics (articulation, rapid tempo, reverberation etc.) the obtained pitch estimation accuracy (maximum df_{pr} value more than 91%) should be considered as high. It should be also noted, that the reference pitch sequence was adjusted by matching individual note duration by auditory comparison with the recorded excerpt used for the experiments. The reference pattern adjustment technique might cause additional pitch estimation errors.

The presented pitch estimation enhancement technique incorporating segmentation and prediction can also be implemented using another fundamental frequency estimators such as comb-filter, cepstral or Meddis-Hewitt model-based methods.

Acknowledgments

Research was subsidized by the Foundation for Polish Science and by the Committee for Scientific Research, Warsaw, Poland. Grant No. 4 T11D 014 22.

REFERENCES

[1] Beauchamp, J. W., "Estimation of Musical Pitch from Recorded Solo Performances", *Proc. of the 94th AES Convention*, Berlin, 16-19 March 1993.

[2] Cook, P. R., Morill, D., Smith, J. O., "An Automatic Pitch Estimation and MIDI Control System for Brass Instruments", *Proc. of Special Session on Automatic Pitch Estimation ASA*, New Orleans, November 1992.

[3] Gelfand, S.A., *Hearing: An Introduction to Psychological and Psychological Acoustics*, Marcel Dekker, N. York, 1998.

[4] Hess, W., Pitch Determination of Speech Signals: Algorithms and Devices, Springer Verlag, Berlin, Heidelberg, New York, Tokyo, 1983.

[5] Hörnel, D., "MELONET I: Neural Nets for Inventing Baroque-Style Chorale Variations", *Advances in Neural Information Processing* 10 (NIPS 10), M. I. Jordan, M. J. Kearns, S. A. Solla (eds.), MIT Press, 1997

[6] Klapuri, A., "Wide-band Pitch Estimation for Natural Sound Sources with Inharmonicities", *Proc. of 106th AES Convention*, Preprint 4906, Munich, May 8-11, 1999.

[7] Kostek, B., Żwan, P., "Wavelet-Based Automatic Recognition of Musical Instrument Classes", *ISMIR 2001* (in print).

[8] McAulay, R.J., Quatieri, T.F., "Sinusoidal Coding", *Speech Coding and Synthesis*, W. B. Kleijn & K. K. Paliwal (eds.), pp. 121-131, Elsevier Science B. V., 1995.

[9] Moradi, H., Grzymała-Busse, J.W., Roberts, J. A., "Entropy of English Text: Experiments with Humans and a Machine Learning System Based on Rough Sets", *Information Sciences*, 104 (1-2), pp. 31-47, 1998.

[10] Mozer, M. C., "Connectionist Music Composition Based on Melodic, Stylistic, and Psychophysical Constraints", *Music and Connectionism*, P. M. Todd & D. G. Loy (eds.), pp. 195-211, The MIT Press, Cambridge, Massachusetts, London, England, 1991.

[11] Paganini, N., *24 Capricci op. 1*, Alexander Markov, CD, Erato 2292-45502-2, 1990.

[12] Rabiner, L., Cheng, M.J., Rosenberg, A.E., Gonegal, C.A., "A Comparative Performance Study of Several Pitch Estimation Algorithms", *IEEE Trans. ASSP 1976*, 24, pp. 399-418.

[13] Szczerba, M., "Recognition and Prediction of Music: A Machine Learning Approach", *Proc. of 106th AES Convention*, Munich, May 8-11, 1999.

[14] Telemann, G. Ph., *Twelve Fantasies For Oboe Solo*, Heinz Holliger, CD, Nippon Columbia, Denon, 38C37-7089, 1984.

[15] Todd, P. M. "A Connectionist Approach to Algorithmic Composition", *Music and Connectionism*, P. M. Todd & D. G. Loy (eds.), pp. 173-194, The MIT Press, Cambridge, Massachusetts, London, England, 1991

[16] Zell, A. u.a., *SNNS – Stuttgart Neural Network Simulator User Manual*, Ver. 4.1.