Musical Instrument Recognition and Classification Using Time Encoded Signal Processing and Fast Artificial Neural Networks

Giorgos Mazarakis[1], Panagiotis Tzevelekos[2], and Georgios Kouroupetroglou[2]

[1] National Technical University of Athens,
Department of Electrical and Computer Engineering
gemazar@mail.ntua.gr
[2] National and Kapodistrian University of Athens,
Department of Informatics and Telecommunications
{taktzev, koupe}@di.uoa.gr

Abstract. Traditionally, musical instrument recognition is mainly based on frequency domain analysis (sinusoidal analysis, cepstral coefficients) and shape analysis to extract a set of various features. Instruments are usually classified using k-NN classifiers, HMM, Kohonen SOM and Neural Networks. In this work, we describe a system for the recognition of musical instruments from isolated notes. We are introducing the use of a Time Encoded Signal Processing method to produce simple matrices from complex sound waveforms, for instrument note encoding and recognition. These matrices are presented to a Fast Artificial Neural Network (FANN) to perform instrument recognition with promising results in organ classification and reduced computational cost. The evaluation material consists of 470 tones from 19 musical instruments synthesized with 5 wide used synthesizers (Microsoft Synth, Creative SB Live! Synth, Yamaha VL-70m Tone Generator, Edirol Soft-Synth, Kontakt Player) and 84 isolated notes from 20 western orchestral instruments (Iowa University Database).

1 Introduction

Automatic music instrument recognition is an essential subtask in many applications regarding music information indexing and retrieval. Computational auditory scene analysis (CASA), automatic music transcription frameworks and content-based search systems, all find such a capability to be extremely helpful. However, musical instrument recognition has not received as much research interest as, for instance, speech and speaker recognition, even though both the amateur music lover and the professional musician would benefit from such systems.

Many attempts in music instrument recognition have taken place in the last thirty years. Most of them have focused on single, isolated notes (either synthesized or natural) and tones taken from professional sound data-bases [1]. Recent works have operated on real-world recordings, polyphonic or monophonic, multi-instrumental or solo [2]. However, the issue is yet far from being solved. The work on recognition from separate notes still remains crucial, since it can lead to further optimization of the methods used and to insights on the recognition of multi-instrumental, commercial recordings.

G. Antoniou et al. (Eds.): SETN 2006, LNAI 3955, pp. 246 – 255, 2006.

The majority of the recognition systems used so far concentrate on the timbral-spectral characteristics of the notes. Discrimination is based on features such as pitch, spectral centroid, energy ratios, spectral envelopes and mel frequency cepstral coefficients [3,4]. Temporal features, other than attack, duration and tremolo, are seldom taken into account. Classification is done using k-NN classifiers, HMM, Kohonen SOM and Neural Networks [5,6]. A limitation of such methods is that in real instruments the spectral features of the sound are never constant. Even when the same note is being played, the spectral components change. One has to take into consideration many timbral components and the way they can vary, which is often rather random, in order to develop a robust recognition system.

In this paper, we present a different instrument recognition approach, based on Time Encoded Signal Processing and Recognition, a time-domain specific feature extraction process. The method encodes signals in a simple and computational lightweight manner, while producing fixed size and dimension structures regardless of the duration or complexity of the signal. Classification is performed using Fast Artificial Neural Networks. For validation, we use isolated, constant-pitch notes. 470 notes produced with 5 velocity scales from 19 instruments, using 5 synthesizers. 28 notes were taken from a public real-instrument database of 20 instruments.

The paper is organized as follows: in Section 2, we describe the recognition and classification methodology used. Section 3 contains the validation procedure and the recognition results. Section 4 concludes this work.

2 Recognition and Classification Method

2.1 Time Encoded Signal Processing

Time Encoded Signal Processing and Recognition, or TESPAR Coding, is a method proposed by King [7, 8] to digitally code speech waveforms. The method is based on infinite clipping (Fig. 1 shows an example), a coding method proposed by Licklidder and Pollack [9]. According to their work, they managed to achieve mean random-word intelligibility scores of 97.9% by differentiating a speech waveform and then removing all amplitude information by performing infinite clipping i.e. preserving only zero-crossing information.

The infinite clipping coding is a direct representation of the duration between the zero crossings of the waveform, i.e. the real zeros of the waveform, thus it is only dependent on the waveform itself and not at the sampling frequency, as long as sampling is performed according to Shannon's theorem.

The above observations on the importance of zeros to the intelligibility of a coded waveform led scientists to further investigate zero-based methods of signal approximation [10, 11]. Author in [11] showed that the introduction of the concept of complex zeros could help overcoming some deficiencies of infinite clipping.

Let a signal waveform of bandwidth W and duration T. The signal contains 2TW zeros, where typically 2TW exceeds several thousand. While the real zeros are easy to determine, complex zeros extraction is a difficult problem involving the factorization of a 2TWth - order polynomial. Such an approach of zeros identification requires significant computational resources and is practically infeasible.

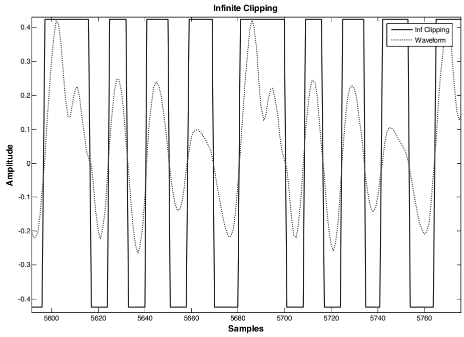

Fig. 1. Infinite Clipping of an oboe waveform

Instead of determining the exact position of complex zeros, which is a complicated task, an approximation of their location could be given. Thus, the waveform is segmented between successive real zeros - the epochs - which comprise the bounds for the complex zeros positions. Complex zeros become visible in the shape of the waveform as minima, maxima or points of inflection and occur in conjugate pairs inside the epoch.

Hence, a band limited waveform may be simply approximated by segmenting it into successive epochs with two features:

- ***Duration (D)*** which is the number of samples between two successive real zeros
- ***Shape (S)*** which is the number of local minima (for a positive epoch) or the number of local maxima (for a negative epoch)

Coding Method. The recorded music waveform is presented to the software implemented TESPAR coder (in Matlab), which segments it into successive epochs. Each epoch is described with a set of numbers representing the Duration and Shape (D/S) of it. This pair is then coded according to a predefined alphabet, representing each epoch by a single "letter". In order to reduce the complexity of this mapping procedure, only the more important D/S pairs are encoded according to an alphabet. The alphabet used depends on the complexity, bandwidth and sampling frequency of the input signal. Most frequency components of a speech signals are in the band of 300Hz to 3 kHz. Authors in [8] use a standard 29 symbol alphabet to encode speech signals sampled at 8 kHz. However, musical waveforms are richer in harmonics so the bandwidth of the signal had to be extended to 100Hz to 5.5 kHz. In order to approximate the music waveform more adequately, the alphabet used was extended to 48 symbols by allowing maximum epoch duration (D) to be 54 samples instead of 37 used in [8]. The aforementioned coding procedure results to a symbol stream, as shown in Figure 2, which can be converted into a fixed-dimension matrix. The N-dimension matrix (where N is the number of the symbols in the alphabet) which contains the number of appearances of each character in the symbol stream is called S-Matrix (Figure 3). Histogram-like, S-Matrices are very descriptive of the waveform from which they were created and can be used for classification purposes. Their fixed dimensions make the

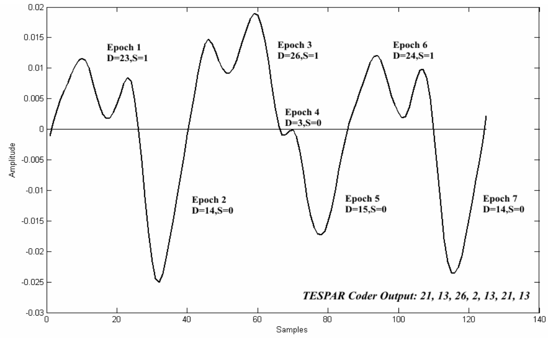

Fig. 2. TESPAR Coding Procedure

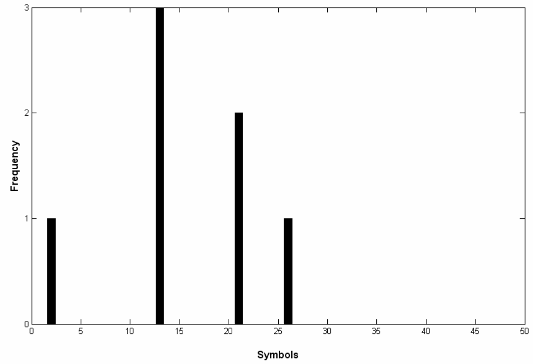

Fig. 3. S-Matrix of figure 2 waveform

classification task using Artificial Neural Networks (ANN) a very enticing solution and the combination of TESPAR with FANNs (Fast Artificial NNs) a very powerful tool for instrument recognition and identification.

2.2 Fast Artificial Neural Networks

FANN is a library which implements a multilayer feedforward ANN, that is, an ANN with neurons ordered in layers, starting with an input layer, continuing with one or more hidden layers and ending with an output layer. The most common networks are fully connected, with connection going only forward, from one layer to the next. The main advantage of this implementation is faster training and testing, compared to similar libraries on systems without a floating point processor, while retaining a comparable performance to other libraries on systems with a floating point processor.

In order to use these networks for classification purposes, two phases must be completed. The first phase is the training phase, where the FANN learns from the imposed input and the requested output. The second phase is the execution phase, where the FANN is presented with unknown input and provides an output. The training process is actually an optimization problem, where the mean square error (MSE) of the entire set of training data must be minimized. The algorithm used to solve this optimization problem is the Backpropagation algorithm. After propagating an input through the network, the error is calculated and then propagated back through the network. In the same time the weights are adjusted in order to make the error smaller. The object of training is to minimize the MSE for all the training data. Training the network on data sequentially one input at a time, instead of training using the whole dataset at once has been proved more efficient. While this means that the order of the data is of importance, this method is a way of avoiding getting stuck in a local minima and stop the training process. A detailed description of FANN library can be found in [12] and a free implementation on different programming languages and platforms is available and maintained under the GNU Lesser General Public License (LGPL) [13].

3 Experimental Dataset and Validation

In order to evaluate the introduced method, several experiments have been conducted with two main objectives: the performance of the system in recognizing synthesized instrument sounds and recognizing instruments from real recordings. All recordings were monophonic, 16-bit wav files downsampled to 11 kHz.

3.1 Synthesized Instruments

Dataset. For this purpose we chose instrument tones, produced with 5 different synthesizers, namely the simple Microsoft Synth, the embedded synthesizer on a Sound Blaster Live! Sound Card, a Yamaha VL-70m Tone Generator, Kontakt player and Edirol Soft-Synth. From each synthesizer 19 instruments (18 instruments for Kontakt player, soprano sax was missing) were selected, each playing C4 note, except the flutes that were all playing C5. Each note was recorded 5 times with 5 different values for velocity (40 for pp-p, 60 for p-mp, 80 for mf-f, 100 for f-ff and 120 for ff-fff) and was named as sample1 - sample5. A total amount of 470 notes was tested.

Validation. For each synthesizer, all notes (19x5=95) were coded with the TESPAR method and the S-Matrices for each note were created. From these matrices, two pairs of datasets were created, each pair used in two experiments accordingly (exp1 and exp2). In exp1, the training data for the FANN was the mean S-Matrix of each instrument (from the 5 note samples) and the test data were all the S-Matrices from the recordings of this synthesizer (95 notes). In exp2, the training data for the FANN were S-Matrices from samples 1,3,5 of each note, while the test data were S-Matrices from samples 2,4. In this experiment training and testing data are completely independent, which is usually the scenario in real-world recognition applications.

3.2 Real Instruments Dataset

Dataset. The evaluation material for testing the system under real conditions is obtained from the original recordings from Iowa University [14]. Recordings from 20 instruments playing C4 and C5 notes (flutes) were used, in vibrato and non vibrato variations and from different strings (for the string family instruments). All instruments were playing in pp, mf and ff, resulting in 3 samples from each note (pp - sample1, mf - sample2 and ff - sample3). A total amount of 84 notes were tested.

Validation. The evaluation method used was almost the same with the one used for the synthesized instruments. S-Matrices were created for each note. In exp1, the training data for the FANN was the mean S-Matrix of each instrument (from the 3 note samples) and the test data were the S-Matrices from the 84 notes. In exp2, the training data were the S-Matrices from pp and ff note samples (sample1,3) and the test data was the S- Matrix from mf note sample (sample2).

3.3 FANN Training

Training a NN is a random procedure that depends on a variety of parameters involving training algorithm, error function, hidden and output layer activation method, learning rate and more. Due to these random results, classification was not based on the results from a single FANN but from 10 parallel FANNs. Five of them were trained using the sigmoid-stepwise function and five using the stepwise function. The averaged result was used for classification purposes. Every one of the 10 parallel FANNs converged after an average of 80 epochs reaching a set Mean Square Error of $MSE \leq 0.01$.

Each FANN has 48 neurons for the input layer plus one bias neuron, one hidden layer with 30 neurons plus one bias neuron and 19 neurons for output (20 for Iowa Music Database [14]). Each output neuron represents one instrument and can take values from 0 to 1. Its value should be 1 in a correct classification of the according instrument, while all others should be 0. This ideal situation results in a MSE of 0.

3.4 Results

Kontakt Player. Tables 1, 2 show the recognition rates for the exp 1, 2 respectively. In both experiments, the higher recognition rates for all the instruments (in bold numbers) correspond to the correct ones. Recognition is successful for all instruments, with all rates rising above 87%, apart from the violin (50% and 58%). The total MSE is 12% in the first experiment and 6% in the second, values that demonstrate the high success of the process for the specific synthesizer. Detailed recognition matrices will not be shown for all tested synthesizers but the brief description that follows is indicative of the effectiveness of the method in all of them.

Microsoft Synth. Highest errors occurred for the violin, the clarinet and the tuba. The total MSE was 53%. In experiment 2, viola was recognized as violin and tuba as trombone. However, in both cases, the correct instruments did get the second higher rate, while the higher rates did remain in the same instrument family group. The total MSE was 50%.

Sound Blaster Live! Synth. In experiment 1, all instruments were recognized successfully, with very high recognition rates (mostly above 90%) and a very low total MSE of 5%. Equivalent results were taken in experiment 2. In both experiments, the french horn delivered the higher MSE, which was still relatively low (65% and 53%).

Yamaha VL70-m Tone Generator. This tone generator uses physical modelling methods to synthesize sound. Thus, the notes produced share the versatility and complexity of natural instrument notes. In both experiments, the higher recognition rates correspond to the correct instruments, while the total MSE is relatively low (36% and 24% respectively). In the second experiment, 8 instruments gather rates above 90%.

Edirol Soft-Synth. In the first experiment all instruments were successfully recognized, while in the second experiment only the piano was mismatched. In both experiments, some instruments gathered high recognition rates while other gathered low. However, in the second experiment, we find very high rates for the violin, the viola, the piccolo, the soprano saxophone, the tenor sax and the trumpet. The MSEs for the two experiments are respectively 45% and 48%.

Iowa Instrument Database. Tables 3, 4 correspond to exp 1, 2 for real-instrument notes obtained from the Iowa Instrument Database. In the first experiment, 26 out of 28 attempts were correctly recognized, while in the second experiment, 22 out of 28. A flute recording was recognized as violin in both experiments. Eb clarinet and bass clarinet were recognized as Bb clarinet in the second experiment. Total MSE is 43% in the first experiment and 58% percent in the second.

Table 1. Kontakt Player Experiment 1

Recognized Stimulus	Violin	Viola	Cello	Contrabass	Piccolo	Flute	Oboe	English Horn	Clarinet	Bassoon	Soprano Sax	Alto Sax	Tenor Sax	Baritone Sax	Trumpet	French Horn	Trombone	Tuba	Piano	MSE
Violin	**58**	0	5	0	17	0	0	0	1	3	0	9	5	0	1	0	2	0	3	**61**
Viola	0	**94**	2	2	0	1	1	0	0	0	0	0	12	0	3	2	0	0	2	**14**
Cello	0	0	**88**	1	3	0	0	0	0	0	0	2	4	3	0	0	2	0	0	**11**
Contrabass	0	0	10	**74**	1	1	0	0	0	0	0	3	1	3	0	0	0	2	0	**30**
Piccolo	5	0	1	0	**93**	0	0	0	0	2	0	1	1	0	0	0	0	1	3	**3**
Flute	0	0	0	1	0	**98**	0	0	0	3	0	3	0	0	0	0	0	0	0	**1**
Oboe	0	0	0	0	0	0	**87**	0	0	0	0	0	0	0	1	0	1	0	0	**2**
English Horn	1	1	0	6	0	1	9	**87**	0	0	0	0	0	0	5	0	0	2	0	**19**
Clarinet	2	0	0	1	0	0	5	0	**93**	0	0	0	1	0	1	0	0	0	1	**3**
Bassoon	1	0	0	0	1	0	0	0	0	**96**	0	0	0	0	0	0	1	0	2	**1**
Soprano Sax	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Alto Sax	3	0	1	0	1	2	0	0	0	0	0	**88**	2	1	0	0	0	0	0	**8**
Tenor Sax	2	8	1	0	1	0	1	1	2	0	0	1	**86**	0	1	0	1	0	1	**11**
Baritone Sax	0	0	11	0	0	0	0	0	0	0	0	0	0	**81**	0	0	0	2	0	**23**
Trumpet	1	0	0	0	2	0	4	1	0	0	0	1	3	1	**93**	0	0	0	0	**5**
French Horn	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	**95**	2	0	0	**2**
Trombone	2	0	0	0	2	0	4	0	0	0	0	0	0	0	0	0	**96**	0	0	**1**
Tuba	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	**99**	0	**0**
Piano	6	0	0	0	8	0	0	0	0	4	0	0	0	0	0	0	0	0	**87**	**13**
																Total MSE				*12*

Table 2. Kontakt Player Experiment 2

Recognized Stimulus	Violin	Viola	Cello	Contrabass	Piccolo	Flute	Oboe	English Horn	Clarinet	Bassoon	Soprano Sax	Alto Sax	Tenor Sax	Baritone Sax	Trumpet	French Horn	Trombone	Tuba	Piano	MSE
Violin	**50**	0	2	0	2	5	0	0	0	0	0	4	4	0	0	0	1	0	2	**55**
Viola	0	**98**	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	**0**
Cello	0	0	**87**	1	0	0	0	0	0	0	0	2	0	1	0	0	0	0	0	**5**
Contrabass	0	0	0	**92**	0	1	0	0	0	0	0	0	0	5	0	0	0	0	0	**5**
Piccolo	1	0	3	0	**99**	0	0	0	0	0	0	0	0	0	0	0	0	0	1	**0**
Flute	1	0	0	1	0	**95**	0	0	0	1	0	15	0	0	0	0	0	0	0	**6**
Oboe	0	0	0	0	0	0	**97**	0	1	0	0	0	0	0	0	0	0	0	0	**0**
English Horn	1	0	0	3	0	0	0	**99**	0	0	0	0	0	0	0	0	0	2	0	**1**
Clarinet	0	0	0	1	0	0	1	0	**99**	0	0	0	0	0	0	0	0	0	0	**0**
Bassoon	1	0	0	1	0	0	0	0	0	**98**	0	0	0	0	0	0	0	0	0	**0**
Soprano Sax	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**-**
Alto Sax	2	0	1	1	0	1	0	0	0	0	0	**97**	2	1	0	0	0	0	0	**1**
Tenor Sax	1	4	5	1	0	0	0	0	0	0	0	1	**95**	0	0	1	0	0	0	**3**
Baritone Sax	0	0	0	8	0	0	0	0	0	0	0	1	0	**87**	0	0	0	0	0	**12**
Trumpet	0	0	0	0	0	0	2	0	0	0	0	0	0	0	**98**	0	0	0	0	**0**
French Horn	0	10	0	0	0	0	1	0	0	0	0	0	3	0	0	**99**	0	0	0	**7**
Trombone	2	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	**93**	0	0	**6**
Tuba	0	0	0	2	0	0	0	5	0	0	0	0	0	0	0	0	0	**99**	0	**5**
Piano	10	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	**95**	**4**
																	Total MSE			***6***

Table 3. Iowa Instrument Database Experiment 1

Recognized Stimulus	Violin	Viola	Cello	Bass	Flute	AltoFlute	BassFlute	Oboe	Bassoon	EbClar	BbClar	BassClar	SopSax	AltoSax	Trumpet	Horn	TenorTromb	BassTromb	Tuba	Piano	MSE
Violin.arco.sulG	**84**	5	0	0	3	5	5	0	1	1	0	6	0	1	1	0	0	0	1	0	**22**
Viola.arco.sulC	4	**75**	0	1	0	0	0	0	0	0	1	0	0	0	19	0	0	0	2	0	**26**
Viola.arco.sulG	17	**92**	0	0	0	0	0	0	0	0	1	0	0	0	16	0	0	0	1	0	**26**
Cello.arco.sulA	0	0	**68**	0	4	2	0	6	0	0	0	0	0	0	1	0	4	0	0	0	**29**
Cello.arco.sulD	0	0	**92**	0	2	2	0	3	0	8	0	0	0	0	0	0	2	0	0	0	**13**
Cello.arco.sulG	1	0	**78**	1	0	2	13	2	0	0	0	0	1	0	0	0	0	1	0	0	**32**
Bass.arco.sulD	0	2	3	**85**	0	0	0	1	0	0	0	7	1	0	0	0	0	0	0	4	**18**
Bass.arco.sulD	0	14	0	**97**	0	0	0	1	0	1	0	2	1	0	0	0	0	0	0	4	**12**
flute.vib	0	0	0	0	**65**	3	56	0	17	0	0	0	1	0	0	0	1	1	0	0	**89**
flute.novib	**32**	0	17	0	29	6	6	1	21	0	0	0	0	0	1	0	11	0	0	0	**124**
AltoFlute	6	0	13	1	16	**62**	26	2	9	0	0	0	2	0	5	3	10	1	0	0	**91**
BassFlute	4	1	0	0	27	3	**78**	0	11	0	0	1	3	0	1	0	1	1	0	4	**47**
oboe	0	0	2	0	0	0	0	**79**	0	0	0	0	0	0	1	0	2	0	0	0	**11**
Bassoon	0	0	0	0	0	0	0	0	**91**	0	0	0	0	0	0	1	0	1	8	0	**7**
EbClar	0	2	0	1	0	0	0	1	0	6	**64**	2	0	5	0	1	0	1	0	0	**150**
BbClar	0	4	0	0	0	0	0	0	2	3	**69**	4	0	4	5	0	0	0	13	0	**37**
BassClarinet	0	8	1	9	0	0	0	0	0	13	10	**40**	0	3	0	0	0	0	0	2	**67**
SopSax.NoVib	0	0	0	0	0	0	0	0	0	0	0	2	**58**	0	0	0	1	1	0	8	**42**
SopSax.Vib	0	0	0	3	0	0	0	7	0	0	0	5	**69**	0	0	0	2	7	0	6	**45**
AltoSax.NoVib	0	2	0	0	0	0	0	0	0	1	33	0	0	**77**	0	1	0	2	8	8	**52**
AltoSax.Vib	1	2	0	0	0	0	0	0	0	1	21	1	0	**79**	0	1	0	0	6	14	**42**
Trumpet.novib	0	9	0	0	0	0	0	14	0	0	4	0	0	0	**95**	0	0	0	0	0	**11**
Trumpet.vib	0	10	0	0	0	0	0	16	0	0	3	0	0	0	**95**	0	0	0	0	0	**13**
Horn	0	0	5	0	0	0	0	1	11	0	1	0	0	0	0	**57**	0	5	11	7	**59**
TenorTrombone	0	0	0	0	2	5	0	6	2	0	0	0	0	0	0	0	**95**	8	0	0	**8**
BassTrombone	0	0	0	0	0	0	3	0	18	0	0	0	0	0	0	22	3	**72**	2	0	**49**
Tuba	0	4	0	0	0	0	0	0	16	0	1	0	0	4	1	1	0	0	**75**	1	**31**
Piano	0	0	0	0	0	0	11	0	0	3	0	3	1	10	0	0	0	0	4	**72**	**39**
																			Total MSE		***43***

Table 4. Iowa Instrument Database Experiment 2

Recognized Stimulus	Violin	Viola	Cello	Bass	Flute	AltoFlute	BassFlute	Oboe	Bassoon	EbClar	BbClar	BassClar	SopSax	AltoSax	Trumpet	Horn	TenorTromt	BassTromb	Tuba	Piano	MSE
Violin.arco.sulG	**89**	0	0	0	4	3	3	0	0	1	0	1	8	0	6	0	10	0	2	0	**31**
Viola.arco.sulC	0	**45**	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	5	0	**49**
Viola.arco.sulG	6	**84**	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	**11**
Cello.arco.sulA	0	1	**49**	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	**38**
Cello.arco.sulD	0	0	**88**	0	0	4	1	0	0	0	0	0	1	0	0	0	0	0	0	0	**9**
Cello.arco.sulG	0	2	**77**	0	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0	0	**21**
Bass.arco.sulD	3	13	0	**70**	0	0	0	0	0	1	0	28	8	0	0	0	0	1	0	0	**62**
Bass.arco.sulD	4	2	0	**97**	0	0	0	0	0	1	0	14	0	0	0	0	0	0	0	0	**10**
flute.vib	0	0	0	0	**77**	13	42	4	0	0	0	0	5	0	0	10	0	2	0	0	**63**
flute.novib	**35**	0	5	0	31	8	1	8	0	0	0	0	0	0	0	0	1	0	0	0	**99**
AltoFlute	18	0	16	1	37	**51**	12	1	0	10	0	0	5	0	10	0	11	2	0	0	**105**
BassFlute	0	0	1	1	23	9	**50**	0	0	0	0	0	10	0	0	0	0	1	0	0	**61**
oboe	0	0	0	0	0	0	0	**96**	0	0	0	0	0	0	6	0	0	0	0	0	**2**
Bassoon	0	0	0	0	5	0	0	0	**90**	0	0	0	0	0	0	1	0	1	0	0	**5**
EbClar	0	0	0	0	0	0	0	0	0	2	**77**	0	0	20	0	0	0	1	2	0	**181**
BbClar	0	0	0	0	0	0	0	0	2	0	**43**	0	0	14	0	0	0	0	5	0	**61**
BassClarinet	0	0	0	2	0	0	0	4	1	9	**32**	9	0	0	0	0	0	10	0	0	**135**
SopSax.NoVib	0	0	3	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	**6**	**100**
SopSax.Vib	0	0	0	0	4	5	8	0	0	0	0	0	7	1	0	0	4	7	0	**9**	**113**
AltoSax.NoVib	0	1	0	0	0	0	0	0	0	0	3	0	0	**71**	0	0	0	1	39	2	**53**
AltoSax.Vib	0	12	15	0	0	0	0	0	0	0	7	2	0	**67**	0	0	0	0	34	0	**67**
Trumpet.novib	0	4	0	0	1	0	0	9	0	0	0	0	0	0	**85**	0	0	0	0	0	**10**
Trumpet.vib	0	0	0	0	1	0	0	26	0	0	0	0	0	0	**96**	0	0	0	0	0	**16**
Horn	0	0	0	0	0	0	0	0	**19**	0	0	0	0	1	0	15	0	15	2	9	**110**
TenorTrombone	0	0	0	0	9	3	1	0	8	0	0	0	0	0	0	8	**95**	10	0	0	**24**
BassTrombone	0	0	0	0	0	2	1	0	17	0	0	0	0	1	0	10	3	**35**	20	6	**89**
Tuba	0	0	0	0	0	0	0	0	7	0	0	0	0	4	1	0	0	2	**66**	0	**27**
Piano	0	0	8	0	0	1	1	0	0	0	0	3	1	1	0	0	0	0	2	**18**	**85**
																				Total MSE	***58***

5 Conclusions

In this paper, we presented a promising method for music instrument recognition and classification, using Time Encoded Signal Processing and Fast Artificial Neural Networks. The method proved to provide high recognition rates with notes produced from synthesizers, as well as with notes from real-instrument recordings.

Future works include evaluation with notes having wider pitch range, from a wider range of synthesizers and natural-instrument recordings. Depending on the results of these tasks, one can continue with instrument identification in multi-instrumental, commercial recordings.

References

1. K.D. Martin: Sound-Source Recognition: A Theory and Computational Model, Ph.D. thesis, MIT, 1999
2. A. Livshin, X. Rodet: Musical Instrument Identification in Continuous Recordings, Proc. of the 7th Int. Conference on Digital Audio Effects (DAFX-04), Naples, Italy, October 5-8, 2004
3. A. Eronen, A. Klapuri: Musical Instrument Recognition Using Cepstral Coefficients and Temporal Features, Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2000, pp. 753-756

4. T. Kitahara, M. Goto, H. Okuno: Musical Instrument Identification Based on F0-Dependent Multivariate Normal Distribution, Proc. of the 2003 IEEE Int'l Conf. on Acoustic, Speech and Signal Processing (ICASSP '03), Vol.V, pp.421-424, Apr. 2003
5. A. Eronen: Musical instrument recognition using ICA-based transform of features and discriminatively trained HMMs, Proc. of the Seventh International Symposium on Signal Processing and its Applications, ISSPA 2003, Paris, France, 1-4 July 2003, pp. 133-136
6. G. De Poli, P. Prandoni: Sonological Models for Timbre Characterization, Journal of New Music Research, Vol 26 (1997), pp. 170-197, 1997
7. J. Holbeche, R. D. Hughes, R. A. King: Time Encoded Speech (TES) Descriptors As A Symbol Feature Set For Voice Recognition Systems. IEE International Conference On Speech Input/Output; Techniques And Applications, pp. 310-315, London, March 1986
8. R. A. King, T. C. Phipps: Shannon, TESPAR and Approximation Strategies. ICSPAT 98, Vol. 2, pp. 1204-1212. Toronto, Canada, September 1998
9. J. C. R. Licklidder, I. Pollack: Effects of Differentiation, Integration, and Infinite Peak Clipping Upon the Intelligibility of Speech. Journal of the Acoustical Society of America, vol. 20, no. 1, pp. 42-51, Jan. 1948
10. F. E. Bond, C. R. Cahn: A Relationship between Zero Crossings and Fourier Coefficients for Bandwidth-Limited Functions. IRE Trans. Information Theory, vol. IT-4, pp. 110-113, Sept.1958
11. E. C. Titchmarsh: The Zeros of Certain Integral Functions. Proc. progres. Math. Soc., vol. 25, pp. 283-302, May 1926
12. S. Nissen: Implementation of a Fast Artificial Neural Network Library (FANN). Report, Department of Computer Science University of Copenhagen (DIKU), 31 October 2003
13. Fast Artificial Neural Network Library (fann): http://leenissen.dk/fann/
14. Univ. of Iowa Electr. Music Studios: http://theremin.music.uiowa.edu/index.html