ROBUST MULTIPITCH ESTIMATION FOR THE ANALYSIS AND MANIPULATION OF POLYPHONIC MUSICAL SIGNALS

Anssi Klapuri[1], *Tuomas Virtanen*[1], *Jan-Markus Holm*[2]

[1]Tampere University of Technology, Signal Processing Laboratory, P.O.Box 553, FIN-33101 Tampere, Finland
[2]University of Jyväskylä, Department of Musicology, P.O.Box 35, FIN-40351, Jyväskylä, Finland
{klap,tuomasv}@cs.tut.fi, jan-markus.holm@jyu.fi

ABSTRACT

A method for the estimation of the multiple pitches of concurrent musical sounds is described. Experimental data comprised sung vowels and the whole pitch range of 26 musical instruments. Multipitch estimation was performed at the level of a single time frame for random pitch and sound source combinations. Note error rates for mixtures ranging from one to six simultaneous sounds were 2.1 %, 2.4 %, 3.8 %, 8.1 %, 12 %, and 18 %, respectively. In musical interval and chord identification tasks, the algorithm outperformed the average of ten trained musicians. Particular emphasis was laid on robustness in the presence of other sounds and noise. The algorithm is based on an iterative estimation and separation procedure and is able to resolve at least a couple of most prominent pitches even in ten sound polyphonies. Sounds that exhibit inharmonicities can be handled without problems, and the inharmonicity factor and spectral envelope of each sound is estimated along with the pitch. Examples are given of musical signal manipulations that become possible with the proposed method.

1. INTRODUCTION

Pitch perception plays an important part in human hearing and in understanding acoustic complexes [1]. While listening to musical signals, humans are able to resolve and perceive the fundamental frequencies of several simultaneous sounds. Computational modeling of this function has been relatively little explored compared to the massive efforts in estimating the pitch of monophonic speech signals for communication purposes [2]. It is generally admitted that single pitch estimation methods are not appropriate as such for multipitch estimation.

Until these days, computational multipitch estimation (MPE) has fallen clearly behind humans in accuracy and flexibility. First attempts were made in the field of automatic transcription of music, but were severy limited in regard to the polyphony (i.e., the number of simultaneous sounds), pitch range, or variety of sounds involved [3]. In recent years, further progress has taken place. Martin proposed a system that utilized musical knowledge in transcribing four voice piano compositions [4]. Kashino et al. describe a model which was able to handle several different instruments [5]. Goto's system was particularly designed to extract melody and bass lines from real-world musical recordings [6]. Psychoacoustic knowledge has been succesfully utilized e.g. in the models of Brown and Cooke [7], Godsmark et al. [8], and de Cheveigne and Kawahara [9]. Also, some purely mathematical approaches have been proposed [10].

The aim of this paper is to propose a general purpose MPE algorithm which operates reliably in rich polyphonies, at a wide pitch range, and for a variety of sound sources. Applications of this are numerous, including the automatic transcription of music, content based music indexing and retrieval, sound separation, and timbre parameter estimation in polyphonic signals. The example application here is sound separation and application of digital audio effects to a musically meaningful part of incoming signals.

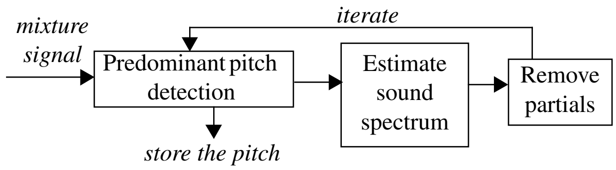

Figure 1: *The iterative estimation and separation approach to multipitch estimation.*

Organization of this paper is as follows. In Section 2, the MPE algorithm is described. This is followed by validation experiments and comparison to musicians' performance in Section 3. A database of sounds in diverse noise conditions was used for statistical evaluation, and listening tests were conducted to make the comparison to human performance. In Section 4, a sound separation mechanism is described, and this is used along with the MPE algorithm to apply audio effects to polyphonic musical signals.

2. MULTIPITCH ESTIMATION

The algorithm consists of two main parts that are applied in an iterative succession, as illustrated in Fig. 1. The first part, predominant pitch estimation, finds the pitch of the most prominent sound in the interference of other harmonic and noisy sounds. In the second part, the spectrum of the detected sound is estimated and subtracted from the mixture. The estimation and subtraction steps are then repeated for the residual signal. For a review and discussion on the earlier iterative approaches, see [11,9].

2.1. Predominant pitch estimation

An overview of the predominant pitch estimation algorithm is to calculate independent pitch estimates at separate frequency bands, and then combine the results to yield a global estimate. This approach was taken to handle sounds that exhibit inharmonicities and to provide robustness in the case of badly corrupted signals where only a fragment of the whole frequency range is good enough to be used. For the sake of computational efficiency, bandwise processing is done in the frequency domain. A single fast Fourier transform is needed, after which local regions of the spectrum are separately processed.

Figure 2 illustrates the processing sequence of the predominant pitch estimation algorithm. First, a discrete Fourier transform

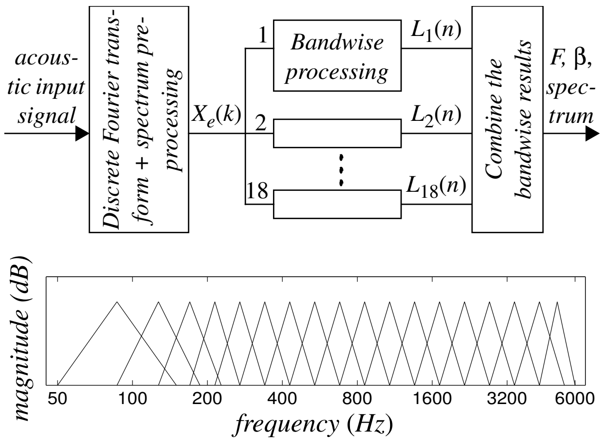

Figure 2: *Processing sequence of the predominant pitch estimation algorithm and the frequency bands at which the calculations take place.*

$X(k)$ is calculated for a Hamming-windowed time domain signal $x(k)$. Before passing the spectrum to pitch analysis, a certain amount of preprocessing takes place in order to eliminate noise and to provide robustness for sounds with irregural spectra. Enhanced spectrum $X_e(k)$ is obtained by taking a logarithm of the magnitude spectrum and highpass liftering the result.

The enhanced spectrum $X_e(k)$ is processed in 18 logarithmically distributed bands that extend from 50 Hz to 6 kHz, as illustrated in Fig. 2. Each band comprises a 2/3-octave region of the spectrum that is subject to weighting with a triangular window. In a logarithmic amplitude scale, this approximates roughly the critical band response of human hearing. The overlap of adjacent windows is 50 %, making them sum to unity.

At each band B, $B \in \{1, 2, \ldots, 18\}$, a fundamental frequency likelihood vector $L_B(n)$ is calculated. The resolution of the vector is the same as that of the enhanced spectrum, each frequency sample $X_e(n)$ having a corresponding fundamental frequency likelihood sample $L_B(n)$. The capital letter F is used to denote fundamental frequency, and the lower case letter f to denote frequency. Sample n corresponds to fundamental frequency value $F=f_s(n/N)$, where N is the size of the time frame in samples and f_s is the sampling rate. Frequency samples $X_e(k)$ at band B are defined to be in the range $k \in [k_B, k_B + K_B - 1]$, where k_B is the lowest sample and K_B is the number of samples at the band.

The bandwise fundamental frequency likelihoods $L_B(n)$ are calculated by finding such a series of every n^{th} spectrum samples at band B that maximizes the likelihood

$$L_B(n) = \max_{m \in M} \left\{ W(H) \sum_{h=0}^{H-1} X_e(k_B + m + hn) \right\}, \qquad (1)$$

where $m \in M$, $M = \{0, 1, \ldots, n-1\}$ is the offset of the series of partials. The value of m is varied to find the maximum value, which is then stored into $L_B(n)$. Different offsets have to be tested because the series of higher harmonic partials may have shifted due to inharmonicity. $H = \lceil (K_B - m)/n \rceil$ is the number of harmonic partials in the sum, and $W(H) = 0.75/H + 0.25$ is used as a normalization factor, because H varies for different n and m. The coefficients in $W(H)$ are important, and were found by training with musical samples in varying conditions.

In the final phase, the bandwise likelihoods are drawn together to yield global pitch likelihoods $L(n)$. Straightforward summation across the likelihood vectors does not associate likelihoods appropriately, since the fundamental frequencies at different bands do not match for inharmonic sounds. Inharmonicity appears as a rising tendency in fundamental frequency as a function of the center frequency of the bands. To overcome this, the inharmonicity factor must be estimated and taken into account [12]. Also, it was found useful to raise the likelihoods to a second power prior to summing in order to provide robustness in strong interference, where the pitch may be observable only at a limited frequency range.

The maximum global likelihood $L(n)$ is used to determine the true fundamental frequency. The output of the algorithm consists of the fundamental frequency F, inharmonicity factor β, and of the frequencies and amplitudes of the harmonic series of the sound. An optional further step is to use these three to calculate a perceptually corrected pitch value according to psychoacoustic measurements [13]. In general, inharmonicity causes a slight rise to the perceived pitch.

2.2. Extension to multipitch estimation

The presented pitch model is capable of making robust predominant pitch detections in polyphonic signals. Provided that the time frame is long enough, one of the correct pitches was found with 99 % certainty even in six-voice polyphonies. Moreover, the precise places of each individual harmonic can be calculated using the fundamental frequency and inharmonicity factor of the detected sound. A natural strategy towards extending the algorithm to MPE is to remove the partials of the detected sound from the mixture, and to apply the pitch algorithm iteratively to the residual spectrum.

Detected sounds can be most efficiently separated in the frequency domain. Two things are needed to remove a sinusoidal partial from the mixture spectrum. First, good estimates of the frequency, amplitude, and phase of the partial must be obtained. Here it will be assumed that these parameters remain constant in the analysis frame. Second, using the estimated parameters of the partial, its spectrum is approximated in the vicinity of the partial, and then linearly subtracted from the mixture spectrum.

Initial estimates for the amplitude a_s, angular frequency ω_s, and phase θ_s of each sinusoidal partial $s(t) = a_s \cos(\omega_s t + \theta_s)$ of a sound are produced by the predominant pitch estimation algorithm. Efficient techniques for estimating more precise values of the parameters have been proposed e.g. in [14]. A method widely adopted to use is to apply Hamming windowing and zero padding in the time domain, and then use quadratic interpolation of the spectrum.

A continuous short-time Fourier transform of $s(t)$ is defined as

$$S(\omega) = \int_0^T [w(t)s(t)e^{-it\omega}]dt, \qquad (2)$$

where $w(t)$ performs temporal weighting by a window function, defined as

$$w(t) = a_w + b_w \cos(\omega_w t + \theta_w), \; t \in [0, T], \qquad (3)$$

and is zero elsewhere. This window model is expressive enough to accommodate e.g. the Hamming window, standard sine window, and a rectangular window. The integral in Eq. (2) can be

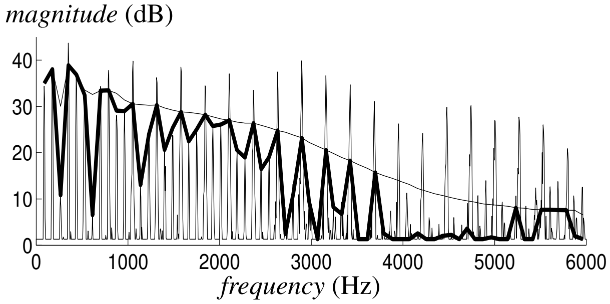

Figure 3: *Illustration of the spectral smoothing principle. The enhanced spectrum contains two sounds, from which the lower has been detected first. See text for details.*

solved analytically in a closed form using straightforward algebra. After this, $S(\omega)$ can be expressed as a function of ω and the parameters of the sinusoid and the window.

It is then an easy matter to apply the solution in the discrete domain to calculate efficiently the desired few Fourier transform samples in the vicinity of a known sinusoid. The solution contains twelve exp-operations, but is still significantly more efficient than generating samples of $s(t)$ in the time domain and calculating their discrete Fourier transform. Parameter estimation, local magnitude spectrum calculation, and subtraction is then repeated for each partial of the sound to be removed from the mixture spectrum.

Simulations were run to evaluate the performance of the described iterative estimation and separation approach. Distribution of remaining errors revealed one more problem to fix. In cases where two sounds are in a rational number relation, a lot of partials from the two sounds coincide, i.e., share the same frequency. When the firstly detected sound is removed, the coinciding harmonics of a remaining sound are also removed in the subtraction procedure. In some cases, and particularly after several iterations, the remaining sound gets too corrupted to be correctly analyzed in the coming iterations.

There is a solution to this problem that is both intuitive, efficient, and psychoacoustically valid: the spectra of the detected sounds must be smoothed before subtracting them from the mixture. The idea is derived from psychoacoustics, since the human auditory system prefers to associate a series of partials to a single acoustic source if they have a smooth spectrum and decreasing amplitude as a function of frequency [1,p.232]. Harmonics that are raised in intensity will segregate more readily from others, and will stand out as an independent sound.

Consider the enhanced spectrum $X_e(k)$ of a two-sound mixture in Fig. 3. The lower-pitched sound has been detected first, and the coinciding partials tend to have higher magnitudes than the other ones. However, when the sound spectrum is smoothed, these partials rise above the smooth spectrum, and thus remain in the residual after subtraction. The smoothing operation was implemented by calculating a moving average over the amplitudes of the harmonic partials. An octave wide hamming window is centered at each harmonic, and a weighted mean is calculated in this window. This smooth spectrum is illustrated by a thin horizontal line in Fig. 3. Then a minimum among the original and the averaged amplitudes is taken, as illustrated by the thick line in Fig. 3. Using the smoothed amplitude values in the subtraction

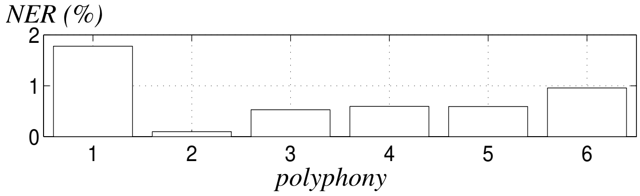

Figure 4: *Note error rates for the predominant pitch estimation in different polyphonies.*

stage made a drastic improvement in simulations, approximately halving the error rate in all polyphonies.

3. SIMULATION RESULTS AND COMPARISON TO HUMAN PERFORMANCE

3.1. Simulation results

A large amount of simulations was run to monitor the behaviour of the proposed algorithm. Test material consisted of a database of sung vowels plus 26 different musical instruments comprising plucked and bowed string instruments, flutes, and brass and reed instruments. These introduce several different sound pruduction mechanisms, and a variety of spectra. Semirandom sound mixtures were generated by first allotting an instrument, and then a random note from its whole playing range, however, restricting the pitch over five octaves between 65 Hz and 2100 Hz. A desired number of simultaneous sounds was allotted, and them mixed with equal mean square levels. Acoustic input was fed to the MPE algorithm that estimated the pitches in a single time frame.

Note error rate (NER) metric was taken into use to measure the pitch estimation accuracy. A correct pitch is defined to deviate less than half a semitone (± 3 %) from the correct value, making it "round" to a correct note in a western musical scale. NER is defined as the sum of the pitches in error divided by the number of pitches in the reference transcription. The errors are of three types. Substitution and deletion errors together can be counted from the number of pitches in the reference that could not be correctly estimated by the system. Insertion errors have occurred if the number of detected pitches exceeds that in the reference.

Figure 4 shows the NERs for predominant pitch estimation in different polyphonies. A predominant pitch estimate was defined to be correct if it matched the true pitch of one of the component sounds. Random mixtures of one to six sounds were generated, five hundred instances of each. Pitch estimation was performed in a single 190 ms time frame. This may seem very long from speech processing point of view but is actually not that long for musical chord identification tasks, where the frequency partial density may be very high in mixtures of low pitches.

The NER of the predominant pitch detection stays around 1 % even in six-note mixtures, showing significant robustness for polyphonic signals. Surprisingly, increasing polyphony even helps to detect at least one of the true pitches. This phenomenon was consistently observed also in MPE, where e.g. the NER for the first three pitch detections was smaller for four-note than for three-note mixtures. The explanation seems to be that richer mixtures are more probable to contain at least one clear sound with no irregularities, which is then detected first, and the more difficult cases remain to subsequent iterations.

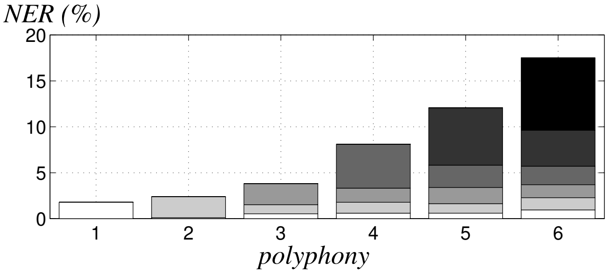

Figure 5: *Note error rates for multipitch estimation in different polyphonies. Bars represent the overall NERs, and the different shades of gray the error cumulation in iteration.*

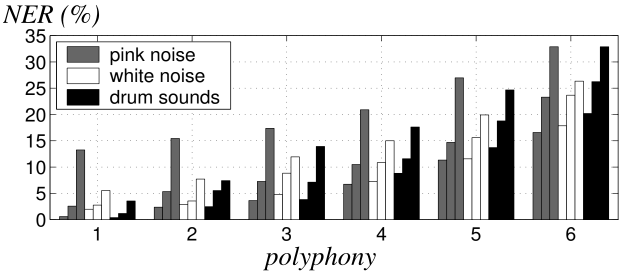

Figure 6: *The effect of additive noise and interfering percussive sounds. Note error rates as a function of polyphony. Three different noise levels are given for each noise type, -15 dB, -5 dB, and 0 dB, reading from left to right.*

Results for multipitch estimation in different polyphonies are shown in Fig. 5. Again, random mixtures were generated and the estimator was then requested to find *N* pitches in a single 190 ms time frame 100 ms after the onset of the sounds. Here the number of sounds to extract, i.e., the number of iterations to run, was given along with the acoustic mixture signal. In Figure 5, the bars represent the overall NERs as a function of the polyphony, where e.g. the NER for random four-voice polyphonies is 8.1 % on average. The different shades of grey in each bar indicate the error cumulation in the iteration, errors occurred in the first iteration at the bottom, and errors of the last iteration at the top.

As a general impression, the system works reliably and exhibits graceful degradation in increasing polyphony, with no abrupt breakdown in any point. This is the strongest advantage of the chosen iterative approach. Performance of the predominant pitch detection can be observed in the bottom slices of each bar, and was discussed above. Analysis of the error cumulation reveals that the errors occurred in the last iteration account for approximately half of the errors in all polyphonies, and the probability of error increases rapidly in the course of iteration. Besides indicating that the subtraction process does not work perfectly, the conducted listening tests suggest that this is a feature of the problem itself, rather than only a symptom of the algorithms used. In most mixtures, there is a sound or two that are very difficult to hear out because their spectrum is virtually buried under the other sounds.

Figure 6 illustrates the effect of different types and levels of additive noise. Pink and white noise was generated in the band between 50 Hz and 10 kHz. Percussion instrument interference was generated by randomizing drum samples from Roland MK–II drum machine. The test set comprised 33 bass drum, 41 snare, 17 hi-hat, and 10 cymbal sounds. Drum samples were set on at the same time with the harmonic sounds. The mean square levels of the harmonic sounds in each mixture were equalized, and the noise level was set in relation to individual sounds in the analysis frame. Thus the noise levels represent signal-to-noise ratios from the viewpoint of each individual sound, not the mixture. A 190 ms frame in 100 ms offset position was applied.

Experiments with different time frame lengths revealed that shortening the frame from 190 ms and 93 ms approximately doubles the error rate in all polyphonies. This is partly caused by the fact that the applied technique was sometimes not able to resolve the pitch with the required ±3 % accuracy. Also, irregularities in the sounds themselves, such as vibrato, are more difficult to handle in short frames. Despite these reservations, the fact remains that reliable MPE seems to require significantly longer time frames than single-pitch estimation.

3.2. Comparison to human performance

Listening tests were conducted to measure the human pitch identification ability, particularly the ability of trained musicians to transcribe polyphonic sound mixtures. Detailed analysis of the results is beyond the scope of this paper, and will be published elsewhere by Holm and Klapuri. Only a summary of the main findings can be reviewed here.

Test stimuli consisted of computer generated mixtures of simultaneously onsetting sounds that were reproduced using sampled Steinway grand piano sounds from McGill University Master Samples collection. The number of co-occurring sounds varied from two to five. The gap between the highest and the lowest pitch in each individual mixture was never wider than 16 semitones in order to make the task feasible for those subjects that did not have absolute pitch, i.e., the rare ability to name the pitch of a sound without a reference tone. Mixtures were generated from six partly overlapping pitch ranges. Here results are reported for three different ranges. The low register extended from 33 Hz to 130 Hz, the middle register from 130 Hz to 520 Hz, and the high register from 520 Hz to 2100 Hz. In total, the test comprised 200 stimuli from 20 different categories.

The task was to write down the musical intervals, i.e., pitch relations, of the presented sound mixtures. Absolute pitch values were not asked, and the number of sounds in each mixture was told in beforehand. Thus the test resembles the musical interval and chord identification tests that are part of the basic musical training in western countries.

A total of ten subjects participated the test. All of them were trained musicians in the sense of having taken several years of musical ear training. Seven subjects were students of musicology at a university level. Two were more advanced musicians, possessing absolute pitch and distinguished pitch identification abilities. One subject was an amateur musician of similar musical ability as the seven students.

Figure 7 shows the results of the listening test. Chord error rates (CER) are plotted for different stimulus categories. CER is the percentage of sound mixtures where one or more pitch identification error occurred. The labels of the categories consist of a number which signifies the polyphony, and of a letter which tells the pitch register used. Letter "m" refers to the middle, "h" to the high, and "l" for the low register. Performance curves are aver-

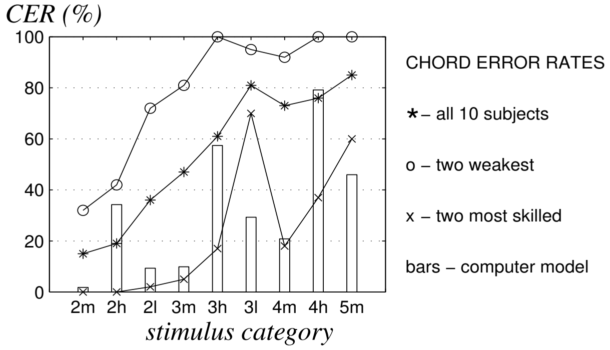

Figure 7: *Chord error rates of the human listeners and of the computational model for different stimulus categories.*

aged over three different groups. The lowest curve represents the two most skilled subjects, the middle curve the average of all subjects, and the highest curve two clearly weakest subjects.

The CERs cannot be directly compared to the NERs given in Fig. 5. The CER metric is more demanding, accepting only sound mixtures where all pitches are correctly identified. It had to be taken into use because absolute pitch values were not asked. In this case, there are several ways of matching pitch intervals with the reference transcription, if the intervals are not all correct. As a rule of thumb, however, about half of the erroneously identified three-note mixtures were cases, where only one of the notes remained undetected. In four-note mixtures, there were usually several incorrect pitches, however, the most skilled subjects having only one note in error, if any.

For the sake of comparison, the stimuli and performance criteria used in the listening test were used to re-evaluate the proposed computational model. Five hundred instances were generated from each category included in Fig. 7, using exactly the same software code that produced samples to the listening test. These were fed to the described MPE system without tailoring its code or parameters. The CER metric was used as a performance measure.

The results are illustrated with bars in Fig. 7. As a general impression, only the two most skilled subjects perform better than the computational model. However, performance differences in high and low registers are quite revealing. The devised algorithm is able to resolve combinations of low sounds that are beyond chance for human listeners. This seems to be due to the good frequency resolution applied. On the other hand, human listeners perform relatively well in the high register. This is likely to be due to an efficient use of the temporal features, onset asynchrony and different decay rates, of high piano tones. These were not available in the single time frame given to the MPE system.

4. APPLICATION TO SIGNAL MANIPULATION

MPE is intimately linked with auditory scene analysis [1,p.240]. The presented algorithm not only outputs the pitches of the mixed sounds, but also indicates the spectrum components that belong to each source. Motivated by this, a sound separation system was developed that attempts to extracts the original time-domain waveforms of each sound before mixing. A dedicated mechanism had to be developed for this purpose, since the MPE system itself operates only in the frequency spectrum of a single time frame.

4.1. Sound Separation

To enable the manipulation of selected parts of a signal, sinusoidal modeling was chosen for signal representation. In a standard sinusoidal model, the signal is analyzed in short frames. In each frame, prominent spectral peaks are located, their frequencies, amplitudes and phases are solved, and then connected to form frame-to-frame trajectories. The output of the model is a set of sinusoids with time-varying frequencies, amplitudes and phases. These can be synthesized in time-domain to represent the harmonic components of the signal as a sum of these trajectories. Sinusoidal model allows the manipulation of signals in parametric form by altering the sinusoidal parameters before resynthesis. Also, the sinusoids can be regrouped to different sound sources in order to synthesize the sounds separately, or make different manipulations to different sounds.

The applied system differs from the stardand sinusoidal model in a few ways. Since the MPE algorithm gives the frequencies of the harmonic components, they do not need to be located but only their time-varying amplitudes and phases are estimated. Also, frame-to-frame tracking is not needed because the frequencies of the harmonic components are assumed constant inside a single MPE window, which is much longer than one sinusoidal modeling frame. Unfortunately, this method fails to detect small changes in the fundamental frequency, such as vibrato.

For a set of sinusoids with known frequencies, the amplitudes and phases can be solved e.g. using the least-square solution presented in [15]. The method gives good results especially in the case that the frequencies of the sinusoids are close to each other – a situation where other methods like obtaining the amplitudes and phases directly from the short-time amplitude spectrum perform poorly. If the the frequencies of two or more sinusoids are too close to each other, their amplitudes cannot be resolved directly. Instead, the parameters or the resulting summary sinusoid are stored, and the component sinusoids are later deduced using the procedure described below.

The amplitudes and phases of the sinusoids are estimated in each time frame. After doing this, the parameters of the coinciding components that could not be directly resolved have to be deduced from their sum. If the frequencies of two components are not exactly the same, the amplitude envelope of the sum of the components modulates at a rate which is the difference between the frequencies of the components. Assuming that the original amplitude envelopes were slowly-varying, we can solve the mixed components as follows. The first amplitude envelope is obtained by lowpass filtering the envelope of the mixed components, and the other by subtracting the first from the original, and then half-wave rectifying and lowpass filtering the difference. Association of the two separated amplitude curves to their due sources of production is done by comparing the curves to other, already solved amplitude envelopes that were not overlapping. This comparison can be done for example using perceptual distance measures presented in [16]. If more than two harmonic components are overlapping, their amplitudes are simply interpolated using the other, already solved components of each sound.

4.2. Manipulation experiments

Further simulations were run to validate the separation procedure, and to experiment with audio effects that process only a meaningful part of an incoming musical signal. Some audio examples are available at http://www.cs.tut.fi/~klap/iiro/dafx2000.

The first experiment aimed at applying basic audio effects on one of the concurrently playing notes in a musical performance. The target sound was selected using varying criteria, separated, and then subtracted from the mixture to obtain a residual signal. Then the chosen sound was manipulated with the desired effect and remixed with the residual signal. Enabled processings comprise basic effects like vibrato or chorus, and more complicated ones, such as sliding between successive pitch values in a melody or breaking chords into notes and playing them in arpeggio.

As a general observation, the separation mechanism is able to extract sounds reliably from mixtures, but when the number of concurrent sounds increases or several harmonics coincide, the quality of the result decreases rapidly. A single misclassified sinusoid may have a very disturbing audible effect on the separated sound when listened to in isolation. The problem is not that outstanding when the separated and manipulated sound is played along with the residual, but the problem still exists. However, if the timbre (i.e. the instrument) of the detected sound is changed, separation is needed only to produce the residual signal, whereas the separated note can be reproduced using another, clean sound.

In the second set of experiments, the analyzed signals were resynthesized using symbolic information only, i.e. the pitch values produced by the MPE system. Separation is not needed in this case, since an acoustic database, instead of separated sounds, provides material to play the MIDI-like information. Enabled processings include the inevitable change in timbre, transposition to a higher or lower pitch register, and rule-based addition or removal of supplementary "play along" parts. The main drawback of this approach is that when the concept of an acoustical residual is renounced, the detected pitches should include *all* the voices present at each time, not only the most prominent ones for which the effects were probably aimed to be applied. It turned out to be very difficult to estimate the number of concurrent voices reliably without utilizing the musical context. On the other hand, detection of some of the weakest sounds is often difficult or impossible.

The third set of experiments aimed at extracting only expressive control information from the original complex musical signal in order to make the instrument changes sound more natural in their original context. Most often when a sound cannot be completely separated from a mixture, some of its harmonics can still be tracked without interference. These can be used to monitor the loudness and pitch contour of e.g. brass and reed instruments, and then to drive the same parameters of the resynthesis samples to make them sound less mechanistic.

5. CONCLUSIONS

Multipitch estimation can be performed quite accurately at the level of a single time frame, with no temporal features available. This applies both to the proposed computational method and to human listeners at a wide pitch range. For a variety of musical sounds, *a priori* knowledge of the involved sounds is not necessary. The presented algorithm works rather reliably in rich polyphonies and in the presence of noisy sounds, such as drums. The presented processing examples demonstrate that the system is generic and reliable enough to enable some novel and more flexible ways of processing polyphonic musical mixtures. However, more efficient utilization of musical predictions and the context is needed to enhance the quality of separated sounds, and to detect more reliably the weakest sounds in rich polyphonies.

6. REFERENCES

[1] Bregman, "Auditory Scene Analysis," MIT Press, 1990.

[2] Rabiner, L. R., Cheng, M. J., Rosenberg, A. E., and McGonegal C. A. (1976). "A Comparative Performance Study of Several Pitch Detection Algorithms," IEEE Trans. Acoust., Speech, and Signal Processing, Vol.ASSP-24, No.5, 399–418.

[3] Klapuri, A. P. (1998). "Automatic Transcription of Music," MSc thesis, Tampere University of Technology, 1998.

[4] Martin, K. D. (1996). "Automatic Transcription of Simple Polyphonic Music: Robust Front End Processing", Massachusetts Institute of Technology Media Laboratory Perceptual Computing Section Technical Report No. 399.

[5] Kashino, K., Nakadai, K., Kinoshita, T., and Tanaka, H. (1995). "Organization of Hierarchical Perceptual Sounds: Music Scene Analysis with Autonomous Processing Modules and a Quantitative Information Integration Mechanism," Proc. International Joint Conf. on Artificial Intelligence, Montréal.

[6] Goto, M. (2000). "A robust predominant-F0 estimation method for real-time detection of melody and bass lines in CD recordings," Proc. IEEE International Conf. on Acoust., Speech, and Signal Processing, Istanbul, Turkey.

[7] Brown, G. J., and Cooke, M. P. (1994). "Perceptual grouping of musical sounds: A computational model," J. of New Music Research 23, 107–132.

[8] Godsmark, D., and Brown, G. J. (1999). "A blackboard architecture for computational auditory scene analysis," Speech Communication 27, 351–366.

[9] de Cheveigné, A., and Kawahara, H. (1999). "Multiple period estimation and pitch perception model," Speech Communication 27, 175–185.

[10] Sethares, W. A., and Staley, T. W. (1999). "Periodicity Transforms," IEEE Trans. Signal Processing, Vol. 47, No. 11.

[11] de Cheveigné, A. (1993). "Separation of concurrent harmonic sounds: Fundamental frequency estimation and a time-domain cancellation model of auditory processing," J. Acoust. Soc. Am. 93 (6), 3271–3290.

[12] Klapuri, A. P. (1999). "Wide-band Pitch Estimation for Natural Sound Sources with Inharmonicities," 106th Audio Engin. Soc. Convention preprint No. 4906, Munich, Germany.

[13] Järveläinen, H., Verma, T., and Välimäki, V. (2000). "The effect of inharmonicity on pitch in string instrument sounds," Proc. International Computer Music Conf., Berlin.

[14] Rodet, X. (1997). "Musical Sound Signal Analysis/Synthesis: Sinusoidal+Residual and Elementary Waveform Models," IEEE Time–Frequency and Time–Scale Workshop, Coventry, Grande Bretagne, août 1997.

[15] Depalle, Ph. Hélie, T. (1997). "Extraction of Spectral Peak Parameters Using a Short-Time Fourier Transform And No Sidelobe Windows". IEEE Workshop on Applications of Signal Processing to Audio and Acoustics". Mohonk, New York.

[16] Virtanen, T., Klapuri, A. P. (2000). "Separation of Harmonic Sound Sources Using Sinusoidal Modeling," Proc. IEEE International Conf. on Acoust., Speech, and Signal Processing, Istanbul, Turkey.