SPECTRUM INTERPOLATION SYNTHESIS FOR THE COMPRESSION OF MUSICAL SIGNALS

Ashish Kumar Malot, Preeti Rao, V. M. Gadre

Department of Electrical Engineering
IIT Bombay
malot@emuzed.com, prao@ee.iitb.ac.in, vmgadre@ee.iitb.ac.in

ABSTRACT

The spectrum interpolation synthesis model has recently been applied in the high quality synthesis of harmonic musical sounds. In this work we investigate the performance of the model in the compression of music signals. Efficient methods for the automatic analysis, parameter extraction and synthesis of musical signals are presented. The system is tested on several examples of segments from wind and bowed string instruments. It is found that typically a perceived quality matching the original is obtained even when large portions of the waveform are generated by interpolation, implying that a high degree of compression is possible. Further, there is a graceful degradation in quality as the extent of interpolation is increased which makes the model well suited for use in a scalable audio coding framework.

1. INTRODUCTION

There are several musical instruments that generate nearly harmonic sound spectra. Such sounds can be characterized by a sequence of pitch cycle waveforms with shapes that evolve slowly over time. This is true even of single instrument notes where, during the sustain portion, the pitch is generally steady but amplitude spectra change with time due to the fact that the player exercises continuous control over the course of the note. A suitable model for the compression of such musical signals should exploit the harmonicity of the signal while allowing the efficient description of its slowly time-varying spectrum. Analysis-synthesis based on spectrum interpolation synthesis [1] offers such a model. In the context of an audio coding scheme, a desired sound can be analyzed, data reduced and streamed as a set of parameters of the sound model.

Spectrum interpolation synthesis (SIS) belongs to the class of wavetable synthesis methods. It uses a model based on the interpolation of available amplitude spectra to reproduce short-time spectral variations. The "wavetable" amplitude spectra can be extracted by analysis of real instrument sounds. While additive synthesis is more general, SIS provides an accurate representation of quasi-periodic sounds which are naturally described in terms of pitch-cycle waveforms. The time varying controls exercised by the player influence the instrument's pitch cycle waveform shape [2]. The evolution of the waveform is represented by the amplitudes of the spectral harmonics which may vary from period to period.

In this paper we investigate the application of SIS for the compression of harmonic musical signals. We describe the implementation of analysis and synthesis algorithms based on [1] and propose effective methods for automatic pitch extraction, spectrum estimation and synthesis of musical signals generated by harmonic instruments. A new error criterion is proposed for the selection of spectra for synthesis. The musical signals we consider are generated by instruments with discrete pitches. Since it is expected that fundamental frequency typically varies much less rapidly with time than do the spectral amplitudes, interpolation of spectral amplitudes can lead to considerable data reduction in the signal representation. The system is tested on several examples of music segments from wind and bowed string instruments. It is found that the sound reconstructed from the estimated fundamental frequency and harmonic amplitudes for selected pitch periods matches the original even at relatively high degrees of interpolation. Further, there is a graceful degradation in quality as the proportion of selected frames is decreased to achieve even higher compression.

The remainder of this paper is organized as follows. In the next section an overview of the technique used for analysis/synthesis for SIS is presented. In Section 3 pitch contour extraction is explained. Analysis and synthesis using linear spectral interpolation is described in Section 4. In Section 5 the performance of SIS in a scalable coding framework is discussed. Finally conclusions and directions for future work are presented.

2. OVERVIEW

Fig. 1 gives an overview of the steps involved for the automatic analysis/synthesis of harmonic signals using SIS. Our input signals are 16-bit PCM signals at a sampling fre-

quency of 16 kHz. The first step is to detect whether an input frame of samples is harmonic, inharmonic or silence. For harmonic frames, a pitch value is estimated for each frame using a frequency-domain approach. Next a smooth pitch contour for the signal is obtained using post-processing. Based on the pitch value the signal is divided into constant-pitch segments. The SIS technique is then applied separately to each segment. Within a segment, the amplitude spectrum for each pitch period is estimated using pitch-synchronous DFT. From the set of DFT spectra a few are chosen based on an error criterion. At the synthesizer linear interpolation is used to reconstruct the remaining spectra. The synthesis is done on a period-by-period basis by adding the outputs of sinusoidal generators driven by the harmonic amplitudes.

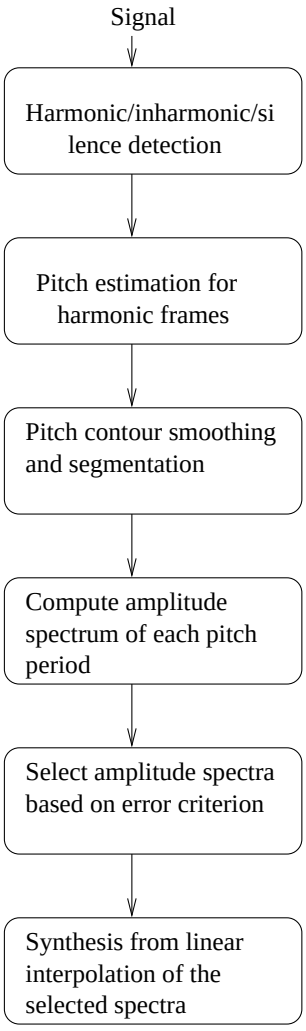

Figure 1: *Steps in the analysis and synthesis of harmonic signals with SIS.*

3. PITCH CONTOUR EXTRACTION

Automatic extraction of fundamental frequency is a difficult problem and various techniques have been proposed in the literature. These techniques can be broadly categorized into - time-domain analysis, frequency-domain analysis and hybrid which uses both time- and frequency- domain analysis. A study by Rabiner [4] concludes that although every algorithm had it's advantages and disadvantages, no algorithm had distinctive lead over others. We have used a pitch-adaptive frequency-domain approach for pitch estimation.

Before doing pitch detection for a frame harmonic/inharmonic/silence detection is done using autocorrelation of the windowed signal (Eq. 1).

$$\phi(k) = \sum_{n=0}^{N-k-1} x(n)x(n+k) \qquad (1)$$

In the above equation k represents the kth lag and N is the total number of samples in the windowed signal. If $\phi(0)$ is very small (close to 0) then silence is detected. If the ratio of first peak (after $\phi(0)$) to $\phi(0)$ is less than 0.4 then the frame is declared as inharmonic, else it is declared as harmonic. For the frames declared as inharmonic or silence a pitch value of zero is assigned. For the frames declared as harmonic the pitch is detected using the procedure explained next.

3.1. Frequency-Domain Based Pitch Estimation

Frequency-domain techniques for pitch detection make use of the fact that the spectra of periodic signals exhibit quasi-harmonic spectral structures manifested by regularly spaced peaks in the magnitude spectrum. We have used an adaptive Hamming window for getting optimum time-frequency resolution. For the first frame, Hamming window of length four times the pitch period corresponding to the minimum frequency expected is chosen. For subsequent frames the window size is adapted to four times the pitch period estimated for the previous frame. Four times is chosen for resolving the peaks reliably in the spectra of the signal. The estimate for frequency is obtained for every 100 samples (6.25 ms at 16 kHz sampling rate). For the windowed signal 4096 point FFT is used for obtaining the spectra. At sampling frequency of 16 kHz it corresponds to a frequency resolution of 3.9 Hz. In the magnitude spectra peaks and their corresponding bin numbers are obtained. The peaks which are very weak relative to the highest peak value are removed. The highest peak in the spectrum either corresponds to the fundamental or to the harmonic of the underlying signal. If the ratio of bin number corresponding to the highest peak(bin_{hp}) to the smallest bin number corresponding to a peak(bin_{sm}) is within 0.2 of an integer then bin_{sm} is chosen for the pitch candidate. The ratio for bin_{hp} with

$bin_{sm}+1$ and $bin_{sm}-1$ is also obtained. Out of these three ratios the one which is closest to an integer gives the bin number corresponding to the fundamental of the windowed signal. Thus an estimate of pitch is obtained for each input frame.

3.2. Pitch Contour Smoothing

Once pitch period is estimated for all the frames, post processing is carried to obtain a smooth pitch contour. Smoothing is required to eliminate the pitch errors which generally occur during the attack or release of a note or at the note transition boundary. The pitch value of a frame is compared with the previous and next frame pitch values. If the pitch value is equal to that of one of the neighbouring frames then it is retained otherwise it is set equal to that of the neighbouring frame pitch value closest to it. This eliminates any abrupt changes in the pitch contour which have occurred for just one frame. After this first level of smoothing we determine the number of consecutive frames having same pitch value to obtain the distinct pitch segments in the given signal. As we know both, the number of samples in each frame and the pitch period, we can compute the number of periods associated with a pitch value. In general for a harmonic signal we obtain a set of distinct pitch values given by $\{f_1, f_2, f_3, ...\}$ and a set of number of periods associated with each pitch value given by $\{N_1, N_2, N_3, ...\}$. This representation is useful in applying SIS technique to a harmonic signal comprising of several notes. For additional smoothing of the pitch contour we compare each of these pitch values, say f_k, to the neighbouring pitch values f_{k-1} and f_{k+1}. If f_k is greater than 100 Hz from both f_{k-1} and f_{k+1} and the number of periods N_k associated with that f_k is also small, then f_k is replaced by either of f_{k-1} and f_{k+1} whichever is closer. This is done to eliminate any spurious pitch detected for a few frames which generally happens during note transition. Fig. 2 shows the pitch contour of a trumpet signal (an excerpt from Haydn trumpet concerto) [2] both before and after smoothing.

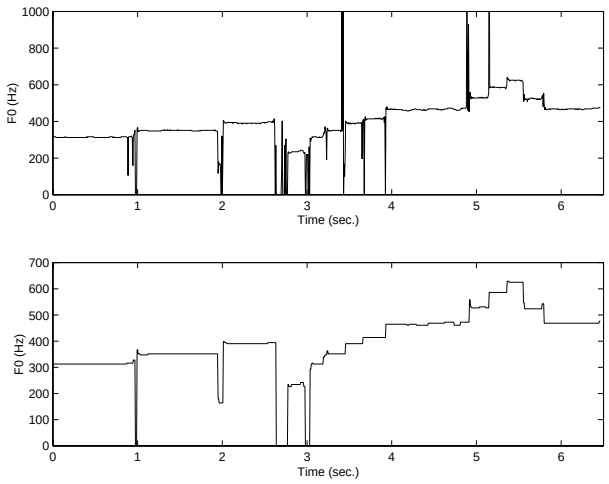

Figure 2: *Pitch contour for a trumpet signal before and after smoothing.*

4. SPECTRAL ANALYSIS AND SYNTHESIS

The spectral analysis of each constant-pitch segment is carried out by pitch-synchronous DFTs. A pitch-synchronous DFT over one pitch period provides the amplitude and phase of each harmonic. The DFT of the ith period of the discrete signal $x(n)$ is defined by Eq. 2

$$X_h^{(i)} = \sum_{n=0}^{P_k-1} x(iP_k+n)\exp^{-j2\pi hn/P_k} \qquad (2)$$

$$0 \le h \le P_k - 1, 0 \le i \le N_k$$

Where P_k is the length of the period in number of samples and N_k is the total number of periods in the k th segment of the signal. The pitch period can have a non-integer number of samples, while we can take DFT of an integer number of samples only. So we have taken the DFT of rounded-off number of samples in one pitch period. By experimenting with the synthetic signals with known spectral values, we observed that the difference in actual spectral values and those computed by this method is less than 0.5 db for the significant harmonics. We calculate H harmonics, $H \le (P_k - 1)/2$, ignoring some higher harmonics. We evaluate the sum given in Eq. 2 directly for each harmonic h, $1 \le h \le H$. The DFT's result in a vector of amplitudes for each pitch period of the segment.

$$S^{(i)} = [a_1(i), a_2(i), ..., a_H(i)] \qquad (3)$$

Where $a_h(i) = |X_h^{(i)}|$. $S^{(i)}$ is called as spectrum measured at ith period. The set of DFT spectra with their time indices,

$$\{(n_0, S^{(0)}), (n_1, S^{(1)}), ..., (n_{N_k-1}, S^{(n_{N_k-1})})$$

, is called the spectral envelope of the tone. Where n_i is the starting time sample index of the ith period.

Once the spectra for all the periods in a segment are available the successive spectra are sent to the synthesis block. Since we are doing synthesis on a period-by-period basis and the pitch period can have non-integer number of samples, we have to keep track of the phases to ensure continuity in the synthesized signal [5]. For each distinct constant pitch segment, say kth, the jth period of total N_k periods is synthesized using the Eq. 4. One period of the waveform is obtained by adding H sine waves, each scaled by corresponding amplitude and the index m runs from $stper$ to $endper_j$ chosen in such a way that the period starts with a non-negative sample and ends with a non-positive sample. This can be ensured because the component sine waves go

to zero both at the beginning and end of the period from the same direction. Synthesis in this fashion also ensures the continuity of the signal even when the change in amplitudes is drastic.

$$x(m) = \sum_{h=0}^{H-1} a_h sin(\frac{2\pi hm}{P_k} + \phi) \quad (4)$$

For first period $stper$ is equal to 0, while for the jth period it is equal to $endper_{j-1} + 1$. The Eq. 4 is repeated until all the N_k periods are synthesized. Since we cannot exactly locate the pitch change position at the note transition boundary, we can not use the DFT values reliably for synthesizing the transition part of the signal. So whenever a note change is detected the last five periods for a segment and first five periods of next segment are synthesized using the amplitude values obtained by linear extrapolation from neighbouring periods, instead of actual values. This ensures smooth transition from one note to the other. From the index of last sample for the kth segment the phase difference, created from the synthesis of kth segment, is calculated using Eq. 5.

$$\phi_k = 2\pi(\frac{1 + endper_{N_k} - P_k * N_k}{P_k}) \quad (5)$$

The phase difference obtained for the kth segment is added to the earlier phases to obtain the total phase (ϕ) to be used for the synthesis of next (k + 1th) segment. This whole process is repeated till all the segments (that is the complete waveform) are synthesized.

5. PERFORMANCE OF SIS

For the k th segment of the signal we have a list of N_k spectra. To achieve compression we choose to transmit Q_k spectra out of N_k based on an error criterion. At the synthesizer the missing spectra are computed from the transmitted Q_k spectra using suitable interpolation. The interpolation between two spectra $S^{(i)}$ and $S^{(j)}$ (successive in the synthesis) can be expressed in terms of their individual harmonics.

$$a_h^{(ij)}(n) = c(n)a_h^{(i)} + d(n)a_h^{(j)} \quad (6)$$

$$n_i \leq n < n_j, 1 \leq h \leq H$$

Since we have used linear interpolation from $S^{(i)}$ to $S^{(j)}$, the effective spectrum at sample n is given by Eq. 7

$$S^{(ij)}(n) = [1 - sp(n)]S^{(i)} + sp(n)S^{(j)} \quad (7)$$

Where $sp(n) = \frac{n-n_i}{n_j-n_i}$ and $n_i \leq n < n_j$.

If the interpolated spectrum on period l is denoted by $S^{(ij)}(l) = S^{(ij)}(n_l)$, where $n_l = lP_k$ and $i \leq l < j$. Then Eq. 7 can be rewritten in the form

$$a_h^{(ij)}(l) = a_h^{(i)} + \frac{n_l - n_i}{n_j - n_i}(a_h^{(j)} - a_h^{(i)}) \quad (8)$$

Comparing the amplitudes of the harmonics of the actual spectrum $S^{(l)}$ with values given by Eq. 8 produces an error $E_l^{(ij)}$. The error criterion given in [1] is sensitive to scaling of spectral amplitudes. From Table 1 we can see that for the same threshold error value (E_{max}) different number of spectra are chosen for different scaling of spectral amplitudes, implying a lack of robustness to changes in overall sound level. This causes difficulty in choosing the value of threshold error. To overcome this we have used a normalized error criterion given in Eq. 9. Table 1 shows that for the normalized error criterion the number of chosen spectra is unaffected by the scaling of amplitudes for a fixed E_{max} value.

$$E_l^{(ij)} = \frac{\sum_{h=1}^{H}(a_h^{(ij)}(l) - a_h^{(l)})^2}{\sum_{h=1}^{H}(a_h^{(l)})^2} \quad (9)$$

The global error $E^{(ij)}$ within the spectral ramp $S^{(ij)}$ is defined as

$$E^{(ij)} = \sum_{l=i+1}^{j-1} E_l^{(ij)} \quad (10)$$

Name	ASF	N	E_{max} (Non-norm.)	Q	E_{max} (Norm.)	Q
$flute_1$	1	872	0.5	56	0.05	68
$flute_2$	2	872	0.5	84	0.05	68
$flute_3$	3	872	0.5	117	0.05	68

Table 1: Comparison of error criteria

If the error $E^{(ij)}$ is less than the tolerated threshold E_{max}, we extend the spectral ramp to the next period $(j + 1)$ and compute $E^{(i,j+1)}$. Otherwise we store the data defining the previous spectral ramp $S^{(i,j-1)}$ and compute the next ramp starting at spectrum $(n_{j-1}, S^{(j-1)})$. In this way we have the Q_k spectra required for the synthesis of the tone.

The choice of threshold error determines the amount of data reduction and the quality of reconstructed sound. We can trade off the quality for bit rate by varying E_{max}. So this also provides for the scalable coding of the harmonic signals. Table 2 shows the results of applying linear spectral interpolation in a low bit rate scalable coding framework. The flute signal has seven notes and significant spectral variations. The clarinet signal has two distinct notes and very slow and small spectral variations. The saxophone signal has three notes with increasing pitch value and has small but rapid spectral variations. The trumpet signal has several notes and has both slow and rapid variations in the pitch and spectral values. In Table 2, R is the ratio of number of selected spectra to the total spectra, Q/N, expressed in percentage. BL stands for the base layer and EL stands for enhancement layer. The base layer contains

the spectra chosen using a high E_{max} value of 0.5. This layer can provide a basic quality of the synthesized signal. We have chosen two enhancement layers representing decreasing values of E_{max} (only the intermediate spectra are stored). Segmental-SNR is calculated for measuring the objective quality of the synthesized signal. It is computed w.r.t the waveform obtained by transmitting all the spectra to the synthesizer. A segment length of 50 ms is chosen for the computation. From the R values we can see that a significant amount of data reduction is obtained for all the signals. From the segmental SNR values we can see that the synthesized signal is very close to the original signal - fact that is further confirmed by listening. Further, there is a graceful degadation in quality with decreasing R. BR gives a rough estimate of bit rate computed by assuming that each parameter is coded using a byte. The bit rate will be mainly governed by the number of chosen spectra. So BR is proportional to the number of spectra transmitted per second, $Q/duration$, multiplied by the number of harmonics. For obtaining some rough estimates for BR we have assumed that the average number of harmonics per spectrum is 20 for a signal sampled at 16 kHz, since we do not expect rapid pitch variations we have assumed that there are 10 pitch values to be sent per second, also the index of each transmitted spectra ($Q/duration$ per sec.) is coded by a byte. So BR is given by $20*8*\frac{Q}{duration}+8*\frac{Q}{duration}+8*10$. From the table we can see that the base layer provides a coarse quality (low SNR) of the synthesized signal while using more layers improves the quality (higher SNR) of the synthesized signal at the cost of higher bit rate (higher BR). The test signals can be found at $http://www.ee.iitb.ac.in/\sim prao/sis.htm.$

Name (Duration)	Layer	R (%)	BR (kbps)	SNR (dB)
flute	BL+EL1+EL2	10.96	16.64	28.61
(3.45 sec.)	BL+EL1	7.80	11.86	24.44
	BL	4.64	7.09	20.37
clarinet	BL+EL1+EL2	1.98	4.37	34.14
(3.92 sec.)	BL+EL1	1.46	3.25	31.39
	BL	0.95	2.14	28.75
saxophone	BL+EL1+EL2	6.29	6.27	29.26
(10.47 sec.)	BL+EL1	4.29	4.30	25.49
	BL	2.28	2.33	21.81
trumpet	BL+EL1+EL2	15.64	10.43	27.59
(6.46 sec.)	BL+EL1	11.00	7.36	23.96
	BL	6.37	4.29	19.90

Table 2: Results of SIS performance in scalable coding

6. CONCLUSIONS

The SIS model (originally proposed for the high quality synthesis of harmonic musical sounds) has been explored for the low bit-rate coding application. Automatic procedures for the analysis, parameter extraction and synthesis of harmonic musical sounds are presented. A new error criterion is proposed for the reduction of spectral frames. Subjective listening and segmental SNR indicate that high quality is achieved even with a large reduction in the number of spectral frames. Further, there is a graceful degradation of quality with increasing compression, making the technique suitable for the scalable coding of music signals. Rough estimates of bit rate are provided for example segments from wind instrument music. Further work is needed to address the issues of parameter quantisation and of the proper handling of transient and other non-harmonic sounds. (See [1], [2] and [3]).

7. ACKNOWLEDGEMENT

The authors wish to thank Sasken Communication Technologies Private Limited for their financial support. Thanks are due to Pushkar Patwardhan for his invaluable inputs and feedback.

8. REFERENCES

[1] M. H. Serra, D. Rubine, and R. Dannenberg, "Analysis and Synthesis of Tones by Spectral Interpolation," J. Audio Eng. Soc., vol. 38, pp. 111-128, March 1990.

[2] I. Derenyi and R. B. Dannenberg, "Synthesizing Trumpet Performances," Proceedings of the International Computer Music Conference, 1998.

[3] P. Masri, and A. Bateman, "Improved Modelling of Attack Transients in Music Analysis-Resynthesis," Proceedings of the International Computer Music Conference, pp. 100-103, 1996.

[4] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg and C. A. McGonegal, "A Comparative Performance Study of Several Pitch Detection Algorithms," IEEE Trans. ASSP, vol. ASSP-24, pp. 399-418, October 1976.

[5] C. Roads, "The Computer Music Tutorial," The MIT Press, London, 2000.