Francois Germain – Slide Presentation IV – Thursday 12 November 2009

REAL-TIME PITCH DETECTION

Summary

Introduction

Pitch extraction is a common research issue in the field of Sound Processing. Many methods have been developed since the first works the 1960's and a lot of approaches have been explored and refined.

Pitch definition

The main objective of pitch extraction could be subject to controversy since there is no exact model for human pitch perception. However, there is an obvious link between the perceived pitch and the fundamental frequency f0 of the sound signal. That's why pitch extraction is also called fundamental frequency detection.

Another issue of the definition of the pitch detection is the definition of the instant frequencies of a signal. If the decomposition in partials is possible for pseudo-periodic signals, it's not the same for quick-varying signals or noisy signals (e.g., unvoiced speech, drum sounds), where such decomposition is difficult or impossible.

Finally, there is a definition problem between monophonic and polyphonic sounds. In the case of monophonic sound, the obvious definition is to pick the lowest partial as the fundamental frequency. In the case of polyphonic sounds, resulting either from one source (e.g., a piano) or from many sources (e.g., orchestra, choir), the definition is far more difficult, and approaches close to the problem of source separation should be used.

Applications of real-time pitch extraction

Pitch extraction was originally designed for speech applications on the problem of decision between voiced and unvoiced sounds (Noll 1967) or for speaker identification. In the field of music, the pitch information is used for real-time music transcription, or for conversion of a live performance into MIDI information. Finally, real-time pitch extraction is used in pitch and/or time-scaling algorithms which require this information to work like the PSOLA algorithm (Moulines and Charpentier 1990), or the Lent's algorithm (Lent 1989).

Requirements for a real-time pitch tracking algorithm

A good algorithm should fulfil the following requirements (Cuadra et al. 2001), in particular if it has to perform in a live performance environment.

The first obvious one is to be able to work real-time. The basic techniques usually fulfil this requirement. The limitation applies mostly on the improvements such as error checking or overlapping of the analysis frames.

The next requirement is to have a minimal latency. This latency results from the algorithm computational time, but also from its convergence delay, since it can take time to find the right pitch, in particular during transients. This issue is critical in the case of MIDI conversion.

The third one is to be robust to noise. This noise can result from the environment where the analysis is performed as a live scene (e.g., reverberation echoes), or from the recording equipment (e.g., electronic noise).

Finally, the algorithm has to fit the sensitivity requirements of the detection. For an example, in the case of western music, the algorithm should have a frequency resolution superior to one semi-tone, and adapted to the possible tuning of the recorded instruments.

Classes of algorithms

As said before, there are numerous different algorithms designed for pitch extraction. Most of them can perform real-time if adapted parameters are chosen. These algorithms are usually split in three categories (Cuadra et al. 2001; Gerhard 2003): time-domain, frequency-domain and statistical methods.

Time-domain methods

This class of methods uses the property of periodicity of the signal. This characteristic makes this approach intrinsically quite weak in the case of inharmonic signals or in signals with most of the power in high frequencies.

The first example is the zero-crossing rate method. In this case, the measure of the distance between two following zero-crossing events is extracted as being the period of the signal. Even if this algorithm has obvious limitations, it gives a useful information that could be used in conjunction with other algorithms.

The second example refers to the similarity measurement method. This approach has been deeply explored since it uses an obvious property of pseudo-periodic signals. This measure is based on different mathematical functions. The most popular ones are the autocorrelation function and the average mean difference. Efficient and robust techniques have been derived from these estimators (Kobayashi 1995; de Cheveigné and Kawahara 2001). In particular, the YIN method presents really promising experimental results. It includes several steps of error cancellation and interpolation for improving the results of the basic algorithm.

Finally, the estimation of the fitting of the signal curve with a set of reference signals gives an estimation of the frequency. This approach has been explored with sinusoids (Choi 1995), showing that the correlation of the signal is high with the sinusoids corresponding to its partials. Furthermore, a property of this correlation allows calculating the correlation with only few frequencies.

Frequency-domain methods

This class of methods exploits the partial lines existing on the spectrum of the signal.

The harmonic product spectrum technique (Noll 1969) measures the coincidence of harmonics in the Fourier transform of a signal frame. This simple approach has some downsides, including the need to enhance the resolution with zero padding and frequent octave errors. The algorithm accuracy depends also on the harmonicity of the signal spectrum.

The maximum likelihood algorithm (Noll 1969) selects the best match between a set of possible spectra. This algorithm could have a good resolution without using large windows. However, the accuracy of the result is obviously bound to the size of the spectra set. The results are quite good for fixed tuning instruments due to the possibility to have a low resolution.

Another popular technique is based on the cepstrum (Noll 1967). In this case, we pick the first peak in the signal synthesized from the log-magnitude of the Fourier transform. This peak would correspond to the fundamental frequency of the signal. This algorithm tends to perform quite well in noisy conditions. However, it handles uneasily inharmonic sounds since it’s based on the assumption of evenly spaced partials

Statistical methods

This class of methods uses the similarities between sounds related to the same pitch. The approach is similar to classify a frame as being part of the set corresponding to its pitch. The obvious downside of this approach is that it often requires a training of the algorithm, which affects the output accuracy.

A model has been developed using neural networks (Barnard et al. 1991) to extract the pitch information in order to classify speech sounds among voiced and unvoiced features. Another approach consists in using Hidden Markov Models. This approach has been used (Bach and Jordan 2005) to do one-singer and multiple-singer pitch tracking

General improvements of the algorithms

A lot of improvements can be added to the presented algorithms such as pre-processing, post-processing or extra information adding.

Pre-processing refers usually to a filtering of the signal prior being processed in the algorithm. This kind of improvement is quite accurate for speech processing where frequencies are clearly band-limited. In the case of music, this method seems less interesting.

Post-processing refers to interpolation or smoothing of the output. These techniques allow to improve the algorithm resolution and to avoid errors for an example in transient frames.

Adding extra information such as using an auditory model or looking at the zero-crossing rate information is another useful way to detect incoherent outputs. Algorithms based on auditory models have been developed (de Cheveigné 1991) but they are not as accurate as it could be expected, while their used in conjunction with other models could lead to better results.

Algorithm evaluation

The evaluation of the compared performance is made difficult due to few issues. Common errors of algorithms are harmonic, subharmonic and transient errors. These error rates are more likely to occur in different conditions.

The absence of tests performed on comparable databases. Usually, the algorithms are tested the class of signals for which it has been designed, where they perform better than in the general case. There is an obvious need of a reference test database in speech (Plante 1995) and music to make comparison. The problem is that it’s necessary to find first the ground truth for each sample, what is not easy since no pitch perception model can automatically label them.

The usual comparison criteria are the gross error rate, the fine error rate and the difference between model outputs. The two first require a ground truth for the tested samples, while the last one is only a test among the outputs of several algorithms.

Conclusion

The problem of pitch extraction in the case of monophonic sounds can be now considered as quite solved. The numerous algorithms available allow the user to pick the more adapted (usually developed for his purpose) to do pitch tracking in the present environment. However, characteristics related to robustness, or latency can still be further improved. Algorithms such as the YIN method have been demonstrated as being accurate and multi-purpose. The problem of multi-pitch extraction stays an open problem where source separation and pitch tracking merge.

References

Abe, T., T. Kobayashi, and S. Imai. 1995 Harmonics tracking and pitch extraction based on instantaneous frequency. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*: 756–59.

Bach, F., and M. Jordan. 2005. Discriminative training of Hidden Markov Models for multiple pitch tracking. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*.

Barnard, E., R.A. Cole, M.P. Vea, and F.A. Alleva. 1991. Pitch detection with a neural-net classifier. *IEEE Transactions on Signal Processing* 39 (2): 298–307.

Choi, A. 1997. Real-time fundamental frequency estimation by least-square fitting. *IEEE Transactions on Speech and Audio Processing*: 201–5.

Cuadra, P., A. Master, and C. Sapp. 2001. Efficient pitch detection techniques for interactive music. *Proceedings of the International Computer Music Conference*.

de Cheveigné, A. 1991. Speech f0 extraction based on Licklider's pitch perception model. *Proceedings of the International Congresses of Phonetic Sciences*.

de Cheveigné, A., and H. Kawahare. 2002. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America* 111 (4): 1917–30.

Gerhard, D. 2003. Pitch extraction and fundamental frequency: history and current techniques. Technical Report TR-CS 2003-6, University of Regina Department of Computer Science.

Goldstein, J. 1973. An optimum processor theory for the central formation of the pitch of complex tones. *Journal of the Acoustic Society of America* 54 (6): 1496–1516.

Kobayashi, H., and T. Shimamura. 1995. A weighted autocorrelation method for pitch extraction of noisy speech. *Proceedings of the Acoustical Society of Japan*: 343–4.

Lent, K. 1989. An efficient method for pitch shifting digitally sampled sounds. *Computer Music Journal* 13 (4).

Licklider, J. 1951. A duplex theory of pitch perception. *Experientia* 7 (4): 128–134.

Moulines, Eric, and Francis Charpentier. 1990. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communications* 9 (5-6): 453–67.

Noll, M. 1967. Cepstrum pitch determination. *The Journal of the Acoustical Society of America* 41 (2): 293–309.

Noll, M. 1969. Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate. *Proceedings of the Symposium on Computer Processing ing Communications*: 779–97.

Plante, F., G. Meyer, and W. Ainsworth. 1995. A pitch extraction reference database. *Proceedings of EUROSPEECH*.

Wise, J., J. Caprio, and T. Parks. 1976. Maximum likelihood pitch estimation. *IEEE Transaction on Acoustics, Speech, Signal Processing* 24 (5): 418–23.