2008-01-01

Pitch Tracking and Voiced/Unvoiced Detection in Noisy Environment using Optimat Sequence Estimation

Moshe Wasserblat
Technological University Dublin

Mikel Gainza
Technological University Dublin, Mikel.Gainza@tudublin.ie

David Dorran
Technological University Dublin, david.dorran@tudublin.ie

Yuval Domb
Technological University Dublin

Recommended Citation

Wasserblat, M., Gainza, M., Dorran, D. & Domb, Y. (2008) Pitch tracking and voiced/unvoiced detection in noisy environment using optimat sequence estimation. *IET Irish Signals and Systems Conference, Galway, Ireland, 2008,* pp.43-48.

Pitch tracking and voiced/unvoiced detection in noisy environment using optimal sequence estimation

Moshe Wasserblat, Mikel Gainza, David Dorran, *Yuval Domb

Department of Electronic Engineering

Dublin institute of Technology, Dublin

email:moshe.wasserblat@gmail.com;

mikel.gainza@dit.ie;david.dorran@dit.ie

** IEEE member, Israel*

***Abstract*— This paper addresses the problem of pitch tracking and voiced/unvoiced detection in noisy speech environments. An algorithm is presented which uses a number of variable thresholds to track pitch contour with minimal error. This is achieved by modeling the pitch tracking problem in such a way that allows the use of optimal estimation methods, such MLSE. The performance of the algorithm is evaluated using the Keele pitch detection database with realistic background noise. Results show best performance in comparison to other state of the art pitch detector and successful pitch tracking is possible in low signal to noise conditions.**

***Keywords* – Pitch tracking, voiced/unvoiced detection, harmonic model.**

I INTRODUCTION

Pitch (fundamental frequency) provides information in speech that is vital in many areas, including speaker identification [1] and emotion detection. By definition, pitch is the perceived fundamental frequency of speech. A candidate for pitch estimation is the position of the maximum autocorrelation function of a voiced frame [2]. This is true in most cases, though in certain cases the position of the maximum can lead to pitch halving, doubling, or other less common errors.

Certain pitch detection algorithms, such as the Modified Autocorrelation Method (AUTOC) [3], Cepstrum Method (CEP) [4] and Average Magnitude Difference Function (AMDF) [5], offer a straight-forward algorithm that perform well on average, but fail regularly a certain percentage of the time. Such pitch detection algorithms are not sufficient if the purpose of the application is to analyze the behavior of the pitch contour. Some algorithms suggest smoothing the pitch counter [6], however, smoothing methods tend to distort the true contour in regions that were detected correctly.

Other pitch detection algorithms, such as [2], offer a tracking method that does consider the pitch as a sequence, however such algorithms lack simplicity as they have many variable thresholds. It is arguable that for an optimal set of these threshold parameters (for a specific signal), this pitch tracking is optimal. However, finding a new set of parameters for each signal is impractical.

Several dynamic programming (DP) methods have been suggested to solve the problem described above. Wang and Seneff [7] developed a spectral domain score function (DLFT) using "template frame" and "cross-frame" spectral correlation functions. A DP search finds a pitch value for each frame. A robust MAP Pitch Tracking was offered in [8], however it requires long sections of voiced speech. The main limitation with this algorithm is that it does not perform the voiced/unvoiced classification, which is a main cause of pitch halving/doubling, a common problem with [4], [5], [6] and other similar algorithms.

This paper provides a solution for the above problems, offering an optimal voiced/unvoiced classification and sub-optimal pitch tracking in some cost sense. The objective is to make the voiced/unvoiced classification, while tracking the pitch as a sequential process. In addition, the proposed algorithm requires that only two variable thresholds be set.

The algorithm proposed in [7] is considered to be state of the art. A comparison of performance between this algorithm and the proposed method is given in the experiment section. Results indicate that both algorithms equally perform for clean speech,

however the proposed method is significantly superior in extreme noise conditions.

II PITCH TRACKING SYSTEM

The system in figure 1 is a complete block diagram of the pitch tracker presented herein. The remainder of this section describes the different blocks.

a) Front End

This module's task is to perform initial cleaning and framing of the audio signal, e.g. passing the signal trough a DC removal filter, detecting silence sections using energy based methods, and performing low pass filtering since pitch frequencies do not exceed 500Hz. The speech is segmented into frames of 26.5ms, with adjacent frames being separated by 10ms. In this paper, t represents a frame index.

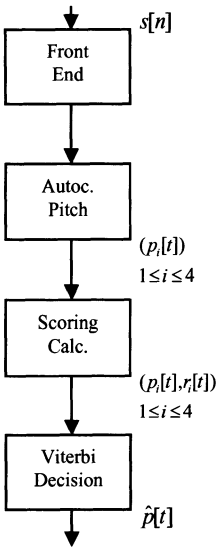

Figure 1: Pitch tracking algorithm flow.

b) Auto-Correlation pitch estimation

This module task is to calculate M-best candidates for pitch estimation in the current frame. First, frames will be classified into voice or un-voiced/silence frames using energy and zero crossing measures. Unvoiced/silence frames will be marked with zero pitch. In an additional process, applied only to voiced frames, the auto-correlation method finds the M-best pitch candidates in the current frames. The M-best candidates are the period of the first M peaks in the auto-correlation function.

In general, **speech** involves three elementary types of excitations – silenced, unvoiced and voiced. Silenced excitations are generally detected easily and are categorized by a constant DC component, usually zero. These excitations are not relevant within this scheme and are not considered further. Unvoiced **speech** excitations include phonemes such as /s/, /f/, /v/, /T/, etc. These excitations are usually modeled by a noisy source that is filtered by the human vocal tract. They, therefore, do not contain much periodicity and have little information regarding pitch. Voiced **speech** excitations include phonemes such as /e/, /E/, /a/, /o/, etc. These excitations are usually modeled by a periodic pulse train source that is filtered by the human vocal tract. These excitations contain periodicity characterized by the period of the source. This period is usually referred to as the pitch period.

Current studies show [9] that all speech contains mixtures of unvoiced and voiced excitations. For simplicity, the proposed algorithm treats frames with strong periodicity as voiced and weak periodicity as unvoiced. Furthermore, it groups silenced and unvoiced speech together, marking them as zero pitch.

The autocorrelation method was used to calculate the M-best pitch-candidates in each voiced frame.

The autocorrelation for a frame is defined as:

$$r[m] = \frac{N}{N-n} \cdot \sum_{n=0}^{N-m-1} s[n]s[n+m] \qquad (1)$$
$$0 \le m \le N-1$$

where N is the length of the speech frame and $s[n]$ is a speech sample. This function has a global maximum at m=0 and local maximums at lags equivalent to its fundamental period and its multiples, as seen in figure 2.

These autocorrelation peaks are found by differentiating the biased periodogram in the range 2ms to 20ms, which correspond to pitch range in speech. The M-best period and value of each peak are recorded in a decreasing order of value. The resulting pairs are defined as the voiced-candidates. Each candidate pair consists of a pitch estimate and its corresponding value.

In order to simplify the algorithm, a maximum of four (M=4) voiced candidates for each frame was used. It was validated in experiment that the true pitch, if it exists, is almost always within the first four candidates.

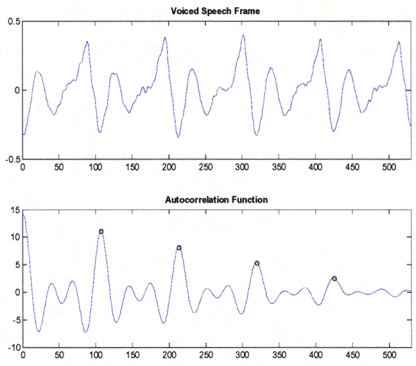

Figure 2 : A strongly periodic speech frame produces distinctive peaks (circled), at pitch multiples. The frame is 26.5ms of voiced speech sampled at 20kHz.

e) Scoring calculation

In this module a normalize score is calculated corresponds for each $p_i[t]$ (the i^{th} pitch candidate in frame *t*). This score represent the periodicity energy of the pitch candidate.

A robust measure of the periodicity in a speech frame can be calculated when given a good estimation of the pitch in that frame. This measure for periodicity is the amount of energy present at the pitch period and its multiples. If a signal has a strong period, most of its energy would exist in that period and its multiples, whereas a non-periodic signal's energy would be distributed otherwise.

Harmonic model for the voiced speech signal is defined as [8]:

$$s_v[n] = \sum_{k=1}^{\infty} A_k \sin(\frac{2\pi n}{kf_0} + \varphi_k) + noise \qquad (2)$$

where A_k is the gain of the *k*'th multiple, f_0 is the estimated pitch value and φ_k is the phase of the *k*'th multiple.

According to detection theory [10] the highest value sampling point of matched filter will gain maximum if the signal present. For a given pitch candidate it is expected that a filter output parameterised with the true pitch-candidate will gain maximum energy in the optimal sampling point. For other pitch candidates the energy value in the best sampling point will be small. This is the rationale for choosing those scores as the likelihood for each pitch candidate to be the true pitch in the current frame. Those likelihood scores will be further used in the *Viterbi* algorithm.

The following, is a mathematical formalization of the above explanation. The exact match filter with the candidate pitch f_0 (true value) is given by:

$$h[n] = \frac{1}{N}\sum_{k=1}^{N} A_k \sin(\frac{2\pi(-n)}{kf_0} + \varphi_k) \qquad (3)$$

where N is the number of pitch multiples taken, A_k is the gain of the *k*'th multiple, f_0 is the estimated pitch value and φ_k is the phase of the *k*'th multiple.

The energy measure is the highest sampled value of the filter's output. For a speech signal obeying the harmonic model the matched filter highest sampled value (with the true pitch candidate) is given by:

$$s_v[n] * h[n]\big|_{n=L} = \frac{1}{N}\sum_{k=1}^{N} A_k^2 \qquad (4)$$

where L is the sampling point, N is the number of pitch multiples taken and A_k is the gain of the *k*'th multiple.

Since the gains A_k and the phases φ_k are unknown, an approximation of this energy is calculated using a breakdown of the above filter. A sequence of filters is used:

$$h_k[n] = -A \cdot \sin(\frac{2\pi n}{kf_0}) \qquad k = 1,2,..,N \qquad (5)$$

where A is the maximum value in the frame, N is the number of pitch multiples taken and f_0 is the estimated pitch value.

The signal is passed through each filter individually, and the highest output value of each filter is sampled. For speech signal obeying the harmonic model:

$$s_v[n] * h_k[n]\big|_{n=L_k} \approx A_k^2 \qquad (6)$$

where L_k is the optimal sampling point for the *k*'th multiple and A_k is the gain of the *k*'th multiple. The sum of these filters' outputs is then divided by the number of multiples to yield the desired result.

f) Viterbi decision (Optimal sequence estimation)

In this module a *Viterbi* framework is used to find the best sequence of states (pitch candidates). Each state is represented by a likelihood score, as explained in the previous section.

The transmitted pitch is modeled as a finite-state Markov chain. The states are the set of the discrete values of periods in the continuous range 2ms to 20ms, and the special state zero. In order to minimize calculations, the time index is limited to a smaller number of states of the voiced and unvoiced candidates. The special state zero, that is, the unvoiced candidate, is always considered. The other states considered, that is, the voiced candidates, are those states whose probabilities are highest. These probabilities correspond to the scores of the candidates, as discussed in previous sections.

The last stage of our algorithm is to estimate the best route through this trellis of states. A sub-optimal *Viterbi* algorithm was used to estimate this route.

As mentioned above, each state is described by an ordered pair:

$$(p_i[t], r_i[t])$$
$$1 \le i \le 4 \qquad (7)$$

where p_i is the pitch candidate, r_i its normalized score and t is the time index.

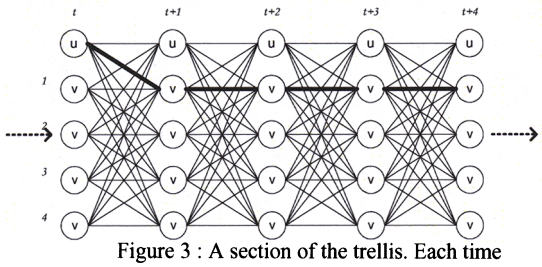

Figure 3 : A section of the trellis. Each time index has one unvoiced state and a few voiced states. The branches show the prosodic routes through the trellis. Each branch has a cost related to it.

A trellis diagram is constructed, example showed in figure 3, and a cost function calculated over its branches:

$$C(t,t-1) = \qquad (8)$$
$$\begin{cases} (1-c_1)\cdot r_d[t] - c_1 \cdot \left| p_d[t] - p_s[t-1] \right| & vv \\ (1-c_1)\cdot r_d[t] - c_1 c_2 \cdot \bar{p} & vuv \\ (1-c_1)\cdot r_d[t] & uuv \end{cases}$$

where s, d are state indices corresponding to source and destination. c_1 is a normalized constant that distributes the weight between the score and the penalty for a branch, c_2 is a normalized constant that sets the voiced to unvoiced cost and $\bar{p}$ is a rough estimation of the pitch expectation. Following the trellis construction, the sequence that maximizes the overall cost are recursively identified [11]. As an example see figure 3, the bold path has the maximum score and the respective sequence is given by $u \to v_1 \to v_1 \to v_1 \to v_1$.

III EXPERIMENTS AND RESULTS

Testing was preformed over the Keele Pitch Referenced database [12]. This database consists of ten files from ten speakers (five males and five females). The files are of pitch-referenced speech recordings sampled at 20 kHz.

Statistical measures suggested in [13] used to evaluate the performance. These statistical measures are described briefly:

UVR – Percentage of found voiced frames that were marked unvoiced from all frames.

VUR – Percentage of found unvoiced frames that were marked voiced from all frames.

GER – Percentage of gross pitch errors (more than or equal to 1ms) from all frames.

Mean – Mean of all fine pitch errors (less than 1ms) presented in samples and ms.

Std – Standard deviation of all fine pitch errors, presented in samples and ms.

Testing was undertaken for the case of no additional noise and white additional noise.

a) No additive noise

The algorithm was tested over the database using several values of (c_1, c_2). The parameters were chosen empirically close to their optimal values, for this database. The results of these simulations are presented in Table 1 and Figure 4.

Table 1 : No additive noise. $c_1 = 0.01$, $c_2 = 0.1$

File	UVR[%]	VUR[%]	GER[%]	Mean[smp]	Mean[ms]	Std[smp]	Std[ms]
f1	0.00%	13.98%	0.06%	0.993	0.050	4.674	0.234
f2	0.00%	18.91%	0.03%	0.983	0.049	2.941	0.147
f3	0.00%	14.33%	0.00%	1.335	0.067	1.743	0.087
f4	0.00%	26.72%	0.32%	0.484	0.024	6.646	0.332
f5	0.00%	12.98%	0.05%	1.089	0.054	1.317	0.066
m1	0.03%	24.22%	0.32%	0.531	0.027	19.135	0.957
m2	0.00%	18.27%	0.35%	0.119	0.006	12.317	0.616
m3	0.00%	21.28%	0.07%	1.146	0.057	1.709	0.085
m4	0.03%	21.58%	0.12%	2.179	0.109	3.704	0.185
m5	0.00%	25.52%	0.25%	1.619	0.081	2.434	0.122
Average	***0.01%***	***19.78%***	***0.16%***	***1.048***	***0.052***	***5.662***	***0.283***

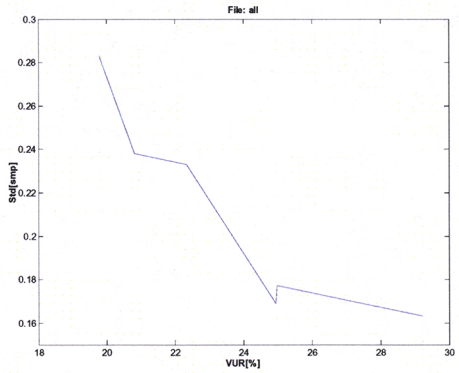

Figure 4: A graph of Std. vs. VUR for different values. Notice the trade-off.

b) With Additive White Gaussian Noise

The algorithm was also tested over the first file while adding white Gaussian noise at different signal-to-noise ratios. This test was repeated several times (Monte Carlo), to average out the results. The results are presented in Table 2 and Figures 5-7. In addition, the proposed method have been compared to the DLFT [6] pitch tracking algorithm with the same test condition. The average mean for the DLFT experiment is added in Figure 6.

Table 2 : Additive White Gaussian Noise at different SNR. $c_1 = 0.01$, $c_2 = 0.1$

SNR [dB]	UVR[%]	VUR[%]	GER[%]	Mean[smp]	Mean[ms]	Std[smp]	Std[ms]
-15	0.18%	32.01%	0.11%	1.090	0.055	4.820	0.241
-12.5	0.18%	26.30%	0.20%	0.962	0.048	5.094	0.255
-10	0.00%	23.63%	0.20%	1.125	0.056	3.246	0.162
-7.5	0.01%	22.40%	0.11%	1.122	0.056	2.420	0.121
-5	0.00%	20.56%	0.07%	1.084	0.054	3.148	0.157
-2.5	0.00%	18.05%	0.09%	1.181	0.059	2.077	0.104
0	0.00%	16.45%	0.07%	1.149	0.057	2.306	0.115
2.5	0.00%	15.34%	0.10%	1.074	0.054	3.429	0.171
5	0.00%	14.52%	0.09%	1.121	0.056	2.775	0.139

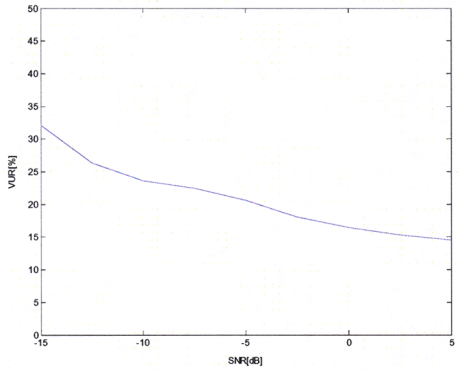

Figure 5 : Average VUR as a function of the SNR for file f1.

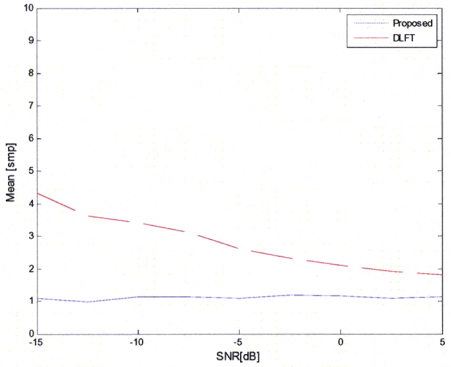

Figure 6 : Average Mean as a function of the SNR for file f1.

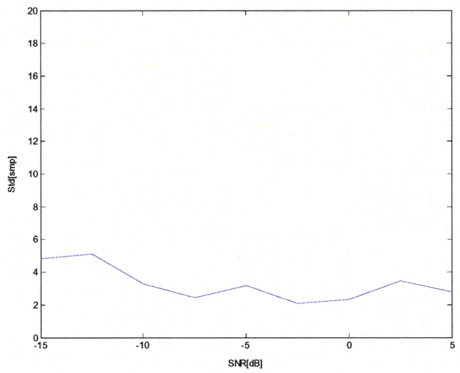

Figure 7 : Average Std as a function of the SNR for file f1.

IV DISCUSSION AND CONCLUSIONS

The key to this algorithm is the sequence estimation. The algorithm calculates a number of possible results for each frame, allowing it to later determine the most appropriate sequence. As a result, the average Mean, as seen in Figure 6, exhibit high robustness for extreme low SNR conditions. Most of the miss-detects occur at the edges of voiced sections, where the energies are low resulting in less reliable scores. This result is desirable, and miss-detect levels could be reduced further by a better choice of threshold parameter

Two factors must be considered when determining the threshold parameters. The first consideration is that the parameters are estimated over a representative sample of the data that the application will need to process. The second consideration is that the optimal choice is dependant on the application's needs. There is a clear trade-off between VUR and Std, as can be seen in Figure 4. This should be taken into account when a working point is set to for an application.

The mean of the pitch tracking is very small, whereas the Std is slightly higher. This difference is mainly due to the sampling error. A good reconstruction of the signal would minimize this affect.

Experiment III(*b*) shows the affect of additive noise. One can see that as the SNR decreases the VUR increases, since more voiced frames become questionable and classified as unvoiced. This explains why the mean and Std are uncorrelated with SNR differences for this algorithm. It is shown that the proposed method outperforms other state of the art pitch detection algorithm for extreme noise conditions.

ACKNOWLEDGMENT

Work supported by the European community under the Information Society Technology (IST) programme of the 6th FP for RTD – project EASAIER contract IST-033902.

REFERENCES

[1] B. S. Atal, "Automatic Speaker Recognition Based On Pitch Contours", J. Acoust. Soc. Amer., vol.52, pp.1687-1697, Dec. 1972.

[2] P. Boersma, "Accurate Short-Term Analysis of the Fundamental Frequency and the Harmonic-to-Noise Ratio of a Sampled Sound", IFA Proceedings 17, pp.97-110, 1993.

[3] J. J. Dubnowski, R. W. Schafer, and L. R. Rabiner, "Real-Time Digital Hardware Pitch Detector", IEEE Trans. Acoust, Speech, Signal Processing, vol.ASSP-24, pp.2-8, Feb. 1976.

[4] R. W. Schafer and L. R. Rabiner, "System for Automatic Formant Analysis of Voiced Speech", J. Acoust. Soc. Amer., vol.47, pp.634-648, Feb. 1970.

[5] M. J. . Ross, H. L. Shaffer, A. Cohen, R. Freudberg, and H. J. Manley, "Average Magnitude Difference Function Pitch Extractor", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-22, pp.353-362, Oct.1974.

[6] D. Hirst and R. Espesser, "Automatic Modelling of Fundamental Frequency Using a Quadratic Spline Function", Travaux de l'Institut de Phonetique d'Aix, vol.15, pp. 75-85, 1993.

[7] C. Wang and S. Seneff, "Robust pitch tracking for prosodic modelling in telephone speech," in *Proc. ICASSP*, 2000.

[8] J. Tabirkian, S. Dubnov, and Y. Dickalov, "Maximum a-posteriori probability pitch tracking in noisy environments using harmonic model", IEEE Transactions on Speech and Audio Processing, 12 (1), pp. 76-87, Jan. 2004.

[9] J.R. Deller, J.G. Proakis, and J.H.L. Hansen, Discrete-Time Processing of Speech Signals, MacMillan, New-York, NY, 1993.

[10] Kay, SM. Fundamentals of Statistical Signal Processing: Detection Theory.

[11] G. D. Forney Jr, "The Viterbi Algorithm", Proceedings of the IEEE, vol. 61, pp. 268-277, Mar. 1973.

[12] G. Meyer, F. Plante and W. A. Ainsworth, "A Pitch Extraction Reference Database", EUROSPEECH 95, Madrid, pp.827-840, 1995.

[13] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg, C. A. McGonegal, "A comparative Performance Study of Several Pitch Detection Algorithms", IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. ASSP-24, pp. 399-418, Oct. 1976.