BaNa: A Hybrid Approach for Noise Resilient Pitch Detection

He Ba*, Na Yang*, Ilker Demirkol†§, and Wendi Heinzelman*,
*Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA
Email: {ba,nayang,wheinzel}@ece.rochester.edu
†Department of Telematics Engineering, Universitat Politecnica de Catalunya, Barcelona, Spain
§Fundacio i2CAT, Barcelona, Spain. Email: ilker.demirkol@entel.upc.edu

***Abstract*—Pitch is one of the essential features in many speech related applications. Although numerous pitch detection algorithms have been developed, as shown in this paper, the detection ratio in noisy environments still needs improvement. In this paper, we present a hybrid noise resilient pitch detection algorithm named BaNa that combines the approaches of harmonic ratios and Cepstrum analysis. A Viterbi algorithm with a cost function is used to identify the pitch value among several pitch candidates. We use an online speech database along with a noise database to evaluate the accuracy of the BaNa algorithm and several state-of-the-art pitch detection algorithms. Results show that for all types of noises and SNR values investigated, BaNa achieves the best pitch detection accuracy. Moreover, the BaNa algorithm is shown to achieve around 80% pitch detection ratio at 0dB signal-to-noise ratio (SNR).**

***Index Terms*—Pitch detection, noise resilience, harmonics, Viterbi algorithm.**

I. Introduction

Subjective pitch is defined by the relative highness or lowness of a tone as perceived by the human ear, and is caused by vibrations of the vocal cords. For perfectly periodic speech signals, pitch is the same as fundamental frequency (F_0), which is the inverse of the speech signal's largest period. However, due to the aperiodicity of the glottal vibration itself and the movement of the vocal tract that filters the source signal, human speech is not perfectly periodic, making the detection of speech pitch rather difficult. Therefore, pitch detection has always been an important challenge of speech signal analysis.

Among the modern state-of-the-art pitch detection algorithms, YIN [1] and Praat [2] are based on the well-known autocorrelation method in the time domain, while the Cepstrum method [3] [4] and Harmonic Product Spectrum (HPS) [5] are based on the spectrum in the frequency domain. YIN uses a difference function to search for the period, and further refines the pitch detection result by two error-reduction steps. Praat, on the other hand, considers the maxima of the autocorrelation of a short segment of the sound as pitch candidates, and chooses the best pitch candidate for each segment by finding the least cost path through all the segments using the Viterbi algorithm. Cepstrum is found by taking the Fourier transform of the log-magnitude Fourier spectrum, which shows a peak corresponding to the period in frequency. HPS multiplies the original signal with downsampled signals, to line up the peak at the pitch value for isolation.

A variety of applications can benefit from a more precise and robust pitch detection algorithm. For example, pitch detection is essential in speech recognition, where homophones can be differentiated by recognizing tones [6]. Also, music notation programs use pitch detection to automatically transcribe real performances into scores [7]. Moreover, in emotion detection or other affective measurement, it has been found that prosodic variations in speech are closely related to one's emotional state, and the pitch information is crucial to identification of this state [8]. Some health care providers and researchers even put pitch detectors and other behavior sensing technologies on mobile devices, such as smart phones, for patient monitoring or behavioral studies [9].

When performing pitch detection in real scenarios, the quality of the input speech signal may be greatly degraded, due to noise introduced by the recording devices or audible background noise. As existing pitch detectors do not perform well for noisy input data, we are motivated to design a noise resilient pitch detection algorithm that is better suited for practical uses. In this paper, we propose a hybrid pitch detection algorithm named BaNa, which combines the idea of using the ratios of harmonic frequencies and the Cepstrum approach to find the pitch from a noisy signal. We test our BaNa algorithm on real human speech samples corrupted by various types of realistic noise. Evaluations show the high noise resiliency of BaNa compared to the state-of-the-art pitch detection algorithms.

II. BaNa Pitch Detection Algorithm

A. Preprocessing

Given a digital audio signal, preprocessing is performed before the extraction of the pitch values. In the BaNa algorithm, we filter the speech signal with a bandpass filter. Since human speech is normally higher than 50 Hz, and lower than 600 Hz, the lower bound of the bandpass filter is set to 50 Hz. The upper bound is set to 3000 Hz, which is 5 times the normal range of human speech at 600 Hz, in order to capture enough harmonics that will later be used for pitch detection.

[1] This research was supported by the Eunice Kennedy Shriver National Institute of Child Health and Human Development Grant R01HD060789.

TABLE I

TOLERANCE RANGE FOR HARMONIC RATIOS.

Ratios	F_0	F_1	F_2	F_3
F_1	[1.9 2.1]			
F_2	[2.8 3.2]	[1.42 1.59]		
F_3	[3.8 4.2]	Discarded	[1.29 1.42]	
F_4	[4.8 5.2]	[2.4 2.6]	[1.59 1.8]	[1.15 1.29]

B. Determination of the pitch candidates

Since harmonics are regularly spaced at integer multiples of the fundamental frequency F_0 in the frequency domain, we use this characteristic of the speech in the proposed BaNa algorithm to achieve the noise resiliency. If we know the frequency of a harmonic and its ratio to F_0, then F_0 can be easily obtained. However, even if a harmonic is discovered, its ratio to F_0 is unknown. Therefore, we propose a pitch detection algorithm that looks for the ratios of potential harmonics and finds the the pitch based on them by applying the following steps.

Step 1: Search for harmonic peaks

Spectral peaks with high amplitudes and low frequencies are preferred to be considered for pitch candidates, since peaks with high amplitudes are less likely to be caused by noise, and peaks with low frequencies are easier to be identified to be harmonics by calculating the ratios. Therefore, we consider the five peaks higher than a certain threshold and with the lowest frequencies to derive pitch candidates. The absolute value of the Fourier transform of the windowed digital signal is given by $|H(k)| = \left| \sum_{n=0}^{N-1} x(n) \cdot w(n) \cdot e^{-j2\pi k \frac{n}{N}} \right|$, where $w(n)$ is a Hann window, and N is set to 2^{16} to provide a good frequency resolution. We use the peak detection algorithm provided in [10] to search for the peaks in the spectrum. We set 1/15th of the maximum amplitude as the amplitude threshold and $40Hz$ as the bandwidth parameter for smoothing in the peak detection function.

Let $\hat{F}_i$ and $\left|\hat{H}_i\right|$ represent the frequencies and magnitudes of the 5 spectral peaks with the lowest frequencies, respectively, where $i = 0, \cdots, 4$. We place the 5 peaks in ascending order of $\hat{F}$. For most human speech, energy concentrates in the low frequency part, thus some or all of the 5 peaks are likely to be at the first 5 harmonics, which are at $m \times F_0$, $m = 1, \cdots, 5$, where F_0 is the fundamental frequency, i.e., pitch. For each frame, pitch candidates are derived from the ratios of $\hat{F}$ using the following algorithm.

Step 2: Calculate pitch candidates

$\forall \hat{F}_i, \hat{F}_j$, where $i < j, i, j = 0, \cdots, 4$, we calculate ratio $R_{ij} = \hat{F}_j / \hat{F}_i$. If the calculated ratio R_{ij} falls into any tolerance range of the harmonic ratios shown in Table 1, whose values were found by tests, we are able to find to which harmonics $\hat{F}_i$ and $\hat{F}_j$ correspond. Thus, a potential pitch candidate can be obtained by dividing the harmonic by its ratio to F_0: $\tilde{F} = \hat{F}_i / m$, where $m = 1, \cdots, 5$. Note that due to the imperfect periodicity of human speech, the harmonics may not be exactly on integer multiples of F_0, and we observed that higher order harmonics have even larger drift than lower order harmonics in practice. Therefore, we set a smaller ratio tolerance range for lower order harmonics, and we set a larger ratio tolerance range for higher order harmonics. In total, $C_2^5 = 10$ ratio values are calculated between every pair of $\hat{F}$. Since both ratios of F_1/F_0 and F_3/F_1 are equal to 2, it is not trivial to differentiate to which harmonics this ratio belongs. In our algorithm, we assume it belongs to F_1/F_0 and calculate the pitch candidate based on that. In addition, we include the peak with the smallest frequency value as one of the pitch candidates, since we have noticed that in some cases only the F_0 peak is high enough to be detected.

In the BaNa algorithm, we also include the pitch value found by the Cepstrum method as an additional candidate to the ones derived by the harmonic ratio analysis. The reason is that the five spectral peaks we choose mainly belong to low frequency values. For some rare cases, the higher order harmonics (e.g., 5th to 10th harmonics) are found to yield higher spectral peak values compared to the low order harmonics. In that case, the spectral peaks at low frequencies are more vulnerable to noise. However, since cepstrum depicts the global periodicity of the spectrum, and considers all spectral peaks, it can help to detect the pitch in those rare cases. In Section III, we show the benefit of including the detected pitch from cepstrum as a candidate.

Let K denote the number of candidates that are derived, where $K \leq C_2^5$. Pitch candidates that are out of the 50-600 Hz human voice range are discarded, and the number of candidates is reduced from K to K'. If no candidate is derived from any ratio, we set the pitch value to 0 Hz. Then for the K' candidates, if two or more candidates are within 10 Hz of each other, those close candidates are considered to be one distinctive candidate. We set the number of such close candidates to be the confidence score V for the corresponding distinctive candidate. Among the D distinctive candidates, the ones with higher confidence scores are more likely to be the pitch.

C. Selection of the pitch from the candidates

In II-B, the distinctive candidates of each frame are obtained independently. However, the pitch values of neighboring frames may correlate, since the pitch values of human speech exhibit a slow time variation, and hence, large pitch jumps among subsequent frames are rare. Therefore, we use the Viterbi algorithm [11] for post-processing to go through all the candidates in order to correct pitch detection errors. We aim to find a path that minimizes the total cost. The cost consists of two parts: the frequency jumps between the candidates of two consecutive frames, and the inverse of the confidence score of each distinctive candidate.

Let $\tilde{F}_i^n$ denote the i^{th} pitch candidate of frame n, and let $\tilde{F}_j^{n+1}$ denote the j^{th} pitch candidate of the next frame. Let N_{frame} denote the number of frames in the speech segment. For every frame n, p_n is the index of the chosen candidate. Thus, $\{p_n | 1 \leq n \leq N_{frame}\}$ defines a path through the candidates. For each path, the path cost is defined to be

$$PathCost(\{p_n\}) = \sum_{n=1}^{N_{frame}-1} Cost\left(\tilde{F}_i^n, \tilde{F}_j^{n+1}\right), \quad (1)$$

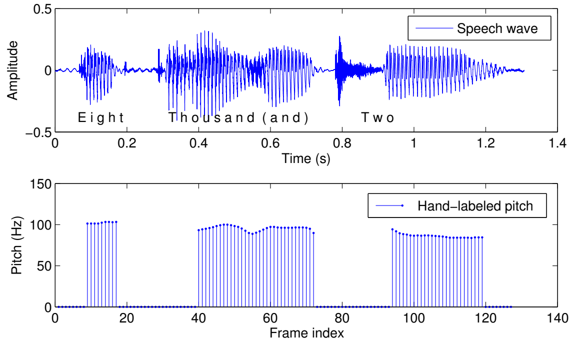

Fig. 1. Speech waveform and hand-labeled pitch values.

where $Cost$ is used to calculate the cost of adjacent frames. We define the function $Cost$ by using the pitch differences between the adjacent frames and the confidence score of the candidates. Since the perceived pitch difference has a logarithm relation with frequency difference, as defined by the Mel scale for pitch, we also model that in the cost function. The larger the pitch difference, the higher the $Cost$ should be. Also, we should assign a lower cost to candidates with higher confidence score, thus we use the inverse of the confidence score in the expression of the cost. A weight w is introduced to balance the two parts. We set its value to 0.2 as determined by tests. Then, $Cost$ is defined mathematically as

$$Cost\left(\tilde{F}_i^n, \tilde{F}_j^{n+1}\right) = \left|\log_2 \frac{\tilde{F}_i^n}{\tilde{F}_j^{n+1}}\right| + w \times \frac{1}{V_i^n}. \qquad (2)$$

We use the Viterbi algorithm to find the minimum cost path, i.e., the path that reduces the pitch jumps the most while giving priority to the pitch candidates with higher confidence scores. The optimal path is found for each voiced part in the speech. Whenever the Viterbi algorithm meets an unvoiced part or irregularly voiced portion of the speech (diplophony, creak), the path cost is reset to 0 and the Viterbi algorithm starts all over again from the next voiced part. The complete BaNa algorithm is given in Algorithm 1.

III. Pitch Detection Evaluation

We use real speech samples to evaluate the pitch detection accuracy of the proposed BaNa algorithm and compare it with Praat, YIN, HPS, and Cepstrum, for different types of additive background noise and for a wide range of signal-to-noise ratio (SNR) values.

A. Parameter settings

The frame length is set to 60 ms, with a frame shift set to 10 ms in order to obtain smooth pitch detection results.

Ten speech samples from the prosody database [12] are used for the pitch detection test, with one male English speaker and four female English speakers. The sampling rate of the ten audio files is 22.05 kHz. Since these original speech samples are clean, with very little background noise, we use the hand-labeled pitch values of the original speech as the ground-truth pitch values and the voiced/unvoiced delineation. Hand-labeling is performed by manually labeling the frequency of the first spectral peak, i.e., the first harmonic for every voiced frame. Fig. 1 shows an example of a clean speech record with hand-labeled pitch values as the ground truth. When evaluating the accuracy, pitch values that differ by more than 10% from the ground truth values are counted as errors.

Algorithm 1 The BaNa Pitch Detection Algorithm

```
// For each frame:
// Select harmonic peaks
select $\hat{F}$: the 5 peaks with lowest frequencies
// Calculate pitch candidates
$\tilde{F}_1 \leftarrow 0$, number of candidates $K \leftarrow 1$
for $i = 1$ to 5, $j = i + 1$ to 5 do
    ratio $R_{ij} = \hat{F}_j / \hat{F}_i$
    for $m = 1$ to 5, $n = m + 1$ to 5 do
        if $R_{ij}$ falls in Table 1 and close to $\frac{n}{m}$ then
            $\tilde{F}_K \leftarrow \hat{F}_i / m$, $K \leftarrow K + 1$
        end if
    end for
end for
add spectral peak with the lowest frequency $\tilde{F}_K = \hat{F}_1$,
    $K \leftarrow K + 1$
add Cepstrum pitch, $\tilde{F}_K = CepstrumPitch$ , $K \leftarrow K+1$
discard $\tilde{F}$ that are out of 50-600Hz
$K' \leftarrow$ number of remaining pitch candidates $\tilde{F}$
if $K' \geq 1$ then
    number of distinctive candidates $D \leftarrow 1$
    for $k = 1$ to $K'$ do
        if $\tilde{F}_k \neq null$ then
            conf. score $V_D \leftarrow 1$, $\tilde{F}_D \leftarrow \tilde{F}_k$, $D \leftarrow D + 1$
            for $l = k + 1$ to $K'$ do
                if $\left|\tilde{F}_l - \tilde{F}_k\right| \leq 10$Hz then
                    $\tilde{F}_l \leftarrow null$, $D \leftarrow D - 1$, $V_D \leftarrow V_D + 1$
                end if
            end for
        end if
    end for
else
    $F_0 \leftarrow 0$
end if
// For all frames within a voiced segment:
// Choose pitch from pitch candidates
for $n = 1$ to number of frames $N_{frame}$ do
    for $i, j = 1$ to $D$ do
        $Cost\left(\tilde{F}_i^n, \tilde{F}_j^{n+1}\right) = \left|\log_2 \frac{\tilde{F}_i^n}{\tilde{F}_j^{n+1}}\right| + w \times \frac{1}{V_i^n}$
    end for
end for
return $\{p_n\}$ of $\min\{PathCost\} \leftarrow Viterbi\,(Cost)$,
    where path $\{p_n\}$ denotes $F_0$ for all frames
```

To test the noise resilience of the proposed algorithm, 8 types of noise are added to the original signal with different SNR values. The noise database we use is [13]. We chose 6

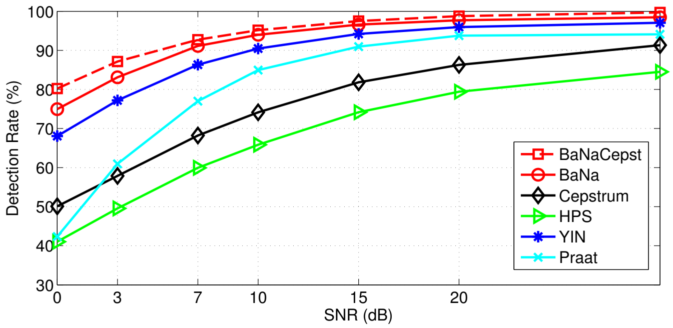

Fig. 2. Accuracy of different algorithms, averaged over all 8 types of noise. BaNa-NoCepst refers to the BaNa algorithm without Cepstrum as a pitch candidate. Inf represents the clean speech with no added noise.

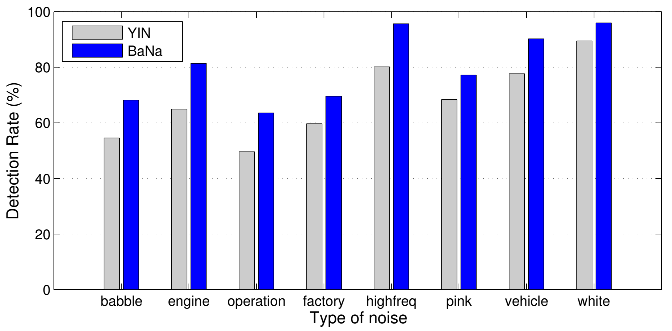

Fig. 3. Accuracy of BaNa and YIN for 8 types of noise at 0dB.

different types of real life background noise: speech babble, destroyer engine room noise, destroyer operations room noise, factory floor noise, vehicle interior noise, HF radio channel noise and 2 common types of noise: white noise and pink noise. To generate noisy speech with a certain SNR value, signal energy is calculated only on the voiced part, and the noise is amplified or attenuated to a certain level to achieve the target SNR value. The synthetic noisy speech data we generate as well as the source code we use to test the algorithms are available on our research group's website [14].

B. *Pitch detection performance*

Pitch detection accuracies of all the algorithms are evaluated as a function of SNR value, where the detection ratios are averaged over all types of noise for each SNR value. Fig. 2 depicts the results, which shows that the BaNa algorithm achieves the best accuracy among all algorithms in terms of detection ratio. It achieves the highest overall average detection ratio of 92.1%. The BaNa algorithm without the Cepstrum candidate is also shown in the results, and has an overall detection ratio of 89.8%, which is still higher than YIN's 85.6% and Praat's 75.4% accuracies. Similar to the BaNa algorithm, the HPS algorithm is also based on the ratios of the potential harmonics. However, in real speech, the harmonics are not integer multiples of the fundamental frequency, which may greatly affect the detection ratio. The Cepstrum method performs the worst and its performance is easily affected by the noise. From Fig. 2, we can see that the BaNa algorithm has a very high resiliency to noise, as it can correctly detect about 80% of pitch values accurately with 0dB SNR, which is 12% higher than YIN and at least 30% higher than the rest of the algorithms.

For a head to head comparison, we present the performances of the BaNa algorithm and the YIN algorithm under 8 different types of 0dB SNR noise in Fig. 3. We can see that BaNa has a better detection ratio for all 8 types of noise. Especially for the speech babble noise, which is a common background noise when a crowd of people are talking at the same time, the BaNa algorithm has 68% detection ratio over YIN's 55% even when the speech is only slightly audible by the human ear.

IV. Conclusions

In this paper, we presented BaNa, a noise resilient hybrid pitch detection algorithm. BaNa was designed to detect pitch in a noisy environment, for example on a mobile phone. This would enable the wide deployment of voice-based applications, such as the ones that use emotion detection. We were able to show that BaNa achieves the best detection rate among all the algorithms investigated from the literature, for different types of background noise, and under different SNR levels from 0dB to 20dB. Even for the very noisy scenario of 0dB SNR, BaNa can still correctly detect about 80% of the pitch values, outperforming the most competitive state-of-the-art reference algorithm YIN by 12%.

References

[1] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, pp. 1917–1930, 2002.

[2] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," in *Proceedings of the Institute of Phonetic Sciences 17*, 1993, pp. 97–110.

[3] A. M. Noll, "Cepstrum pitch determination," *Journal of the Acoustical Society of America*, vol. 41, pp. 293–309, 1967.

[4] L. R. Rabiner and R. W. Schafer, *Theory and Application of Digital Speech Processing*. Pearson, 2011.

[5] M. R. Schroeder, "Period histogram and product spectrum: New methods for fundamental frequency measurement," *Journal of the Acoustical Society of America*, vol. 43, pp. 829–834, 1968.

[6] C. Wang, "Prosodic modeling for improved speech recognition and understanding," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[7] J. P. Bello, G. Monti, and M. Sandler, "Techniques for automatic music transcription," in *International Symposium on Music Information Retrieval*, 2000, pp. 23–25.

[8] O. W. Kwon, K. Chan, J. Hao, and T. W. Lee, "Emotion recognition by speech signals," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[9] K. Chang, D. Fisher, and J. Canny, "AMMON: A Speech Analysis Library for Analyzing Affect, Stress, and Mental Health on Mobile Phones," in *2nd International Workshop on Sensing Applications on Mobile Phones*, 2011.

[10] Thomas O'Haver, Command-line findpeaks MATLAB function, http://terpconnect.umd.edu/~toh/spectrum.

[11] P. van Alphen and D. van Bergem, "Markov models and their application in speech recognition," in *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam*, 1989, pp. 1–26.

[12] "Emotional prosody speech and transcripts database," http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2002S28.

[13] "Noise database from Rice University," http://spib.ece.rice.edu/spib/data/signals/noise/.

[14] "Generated noisy speech data and BaNa source code, WCNG website," http://www.ece.rochester.edu/projects/wcng/project_bridge.html.