Pitch Extraction and Fundamental Frequency: History and Current Techniques

David Gerhard
Technical Report TR-CS 2003-06
November, 2003

Department of Computer Science
University of Regina
Regina, Saskatchewan, CANADA
S4S 0A2

ISSN 0828-3494
ISBN 0 7731 0455 0

Pitch Extraction and Fundamental Frequency: History and Current Techniques

David Gerhard

Abstract: Pitch extraction (also called fundamental frequency estimation) has been a popular topic in many fields of research since the age of computers. Yet in the course of some 50 years of study, current techniques are still not to a desired level of accuracy and robustness. When presented with a single clean pitched signal, most techniques do well, but when the signal is noisy, or when there are multiple pitch streams, many current pitch algorithms still fail to perform well. This report presents a discussion of the history of pitch detection techniques, as well as a survey of the current state of the art in pitch detection technology.

1 Introduction

Fundamental frequency (f_0) estimation, also referred to as pitch detection, has been a popular research topic for many years, and is still being investigated today. At the 2002 IEEE International Conference on Acoustics, Speech and Signal Processing, there was a full session on f_0 estimation. The basic problem is to extract the fundamental frequency (f_0) from a sound signal, which is usually the lowest frequency component, or *partial*, which relates well to most of the other partials. In a periodic waveform, most partials are harmonically related, meaning that the frequency of most of the partials are related to the frequency of the lowest partial by a small whole-number ratio. The frequency of this lowest partial is f_0 of the waveform.

Most research into this area goes under the name of pitch detection, although what is being done is actually f_0 estimation. Because the psychological relationship between f_0 and pitch is well known, it is not an important distinction to make, although a true pitch detector should take the perceptual models into account and produce a result on a pitch scale rather than a frequency scale.

Current speech recognition engines often discard the pitch information as irrelevant to the recognition task. While it is true that individual phonemes are recognizable regardless of the pitch of the driving function, or even in the absence of pitch as in whispered speech, this does not imply that pitch information is not useful. Much semantic information is passed on through pitch that is above the phonetic and lexical levels. In tonal languages, the relative pitch motion of an utterance contributes to the lexical information in a word. In this case, speech recognition algorithms must attend to the pitch or the context of the utterance to avoid ambiguity.

2 Theory of Pitch

The musical pitch of an audio signal is a perceptual feature, relevant only in the context of a human listening to that signal. The musical pitch scales that are used today were developed before people knew about frequency and spectral content, and was based on the similarity or dissimilarity of the note. Pitch is loosely related to the log of the frequency, perceived pitch increasing about an octave with every doubling in frequency. However, frequency doubling below 1000 Hz corresponds to a pitch interval slightly less than an octave, while pitch doubling above 5000 Hz corresponds to an interval slightly more than an octave [8, 14]. This relationship also changes with intensity. The perceived pitch of a sinusoid increases with intensity when the sinusoid is above 3000 Hz, and a sinusoid with frequency below 2000 Hz is perceived to drop in pitch as the intensity increases [2].

It is important to note that these measurements of the differences between frequency and the perception were made on isolated sinusoids. Real-world sounds have many harmonics above the fundamental frequency. The perception of pitch changes with this harmonic content as well. A richer spectrum seems to reinforce the sensation of the pitch, making the octave seem more "in-tune". The more sine-like a waveform is, the more distinct the notion of frequency, but the less distinct the perception of pitch [29]. This sensation also varies with the relationship between the partials. The more harmonically related the partials of a tone are, the more distinct the perception of pitch. Pitch perception also changes with intensity, duration and other physical features of the waveform.

There is some controversy as to how the human auditory system perceives pitch [1, 18, 30]. One group of people have traditionally used pure tone pitches to measure phenomena like critical bands, masking, and pitch perception. The other group of people use more complex tones to see how humans perceive groups of sounds and dissect the "scene" of sound around them. There are also important observations arising from the psychology, psychoacoustics and psychophysics being researched around the perception of tones and pitch, which provide insight into the problem of automatic f_0 estimation. For our purposes, it is less important to decide which general theory of audition is right, and more important to glean information about how humans perceive pitch from each group of researchers.

3 Automatic Fundamental Frequency Estimation

Fundamental frequency estimation has consistently been a difficult topic in audio signal processing. Many context-specific attempts have been made, and many of them work well in their specific context, but it has been difficult to develop a "context-free" f_0 estimator. f_0 estimators developed for a particular application, such as musical note detection or speech analysis, are well understood, but depend on the domain of the data: a detector designed for one domain is less accurate when applied to a different domain. The result is that there are many f_0 estimators currently on the market, but few that are appropriate to more than one domain.

Therefore, choosing a f_0 estimator for a speech/song discrimination is a difficult task because detectors that work well for music, and hence for song, work less well for speech, and vice versa. Three possible solutions to this problem are: find a detector that is reasonably good for both speech and song; build a detector that works very well for both speech and song; or use two f_0 estimators, one suited to speech and one suited to song, and compare the results. The latter generates two positive outcomes: the f_0 estimation is more reliable, and the differences between the f_0 estimations can be used as a classification feature between speech and song. For this report, f_0 estimators developed for speech and for instrumental music were found, but not specifically for vocal music. For this reason, it was decided to evaluate a set of f_0 estimators and choose one which was mostly accurate for both speech and song.

3.1 Evaluating Fundamental Frequency Estimators

It is difficult to empirically measure the performance of a f_0 estimator for several reasons. First, performance depends on domain, as discussed above. A f_0 estimator will almost certainly behave better in the context for which it was developed. Second, it is difficult to automatically rate the result of a f_0 estimator against expected outcomes, precisely because it is difficult to measure f_0 in the first place. We humans are good at it, and so we can listen to a file and judge the accuracy of a f_0 estimation engine, but to lend credibility to this measure, we must have many people, both expert and lay, judge the f_0 estimation result on a large number of sound files. Once a measure like this is taken, however, it can be used to evaluate the results of other f_0 estimation methods. Another way to evaluate f_0 estimators is to compare the results of multiple detectors on a common corpus. If, for a set of n detectors, $k \approx n$ of them agree, it is likely that the remaining $n - k$ are incorrect.

This third method of comparison is what will be used in this work. Section 8 presents an evaluation of three f_0 estimators by comparing their results. Errors in one f_0 estimator provide evidence that the other two are likely to be more accurate, and visual inspection of the f_0 tracks which are significantly different provide further insight into which f_0 track techniques may be better than others.

3.2 Measuring Frequency

There are a number of standard methods that researchers use to extract f_0, based on various mathematical principles. Since pitch is a perceptual quantity related to f_0 of a periodic or pseudo-periodic waveform, it should suffice to determine the period of such oscillation, the inverse of which is the frequency of oscillation. The problem comes when the waveform consists of more than a simple sinusoid. As harmonic components are added to a sinusoidal waveform, the appearance of pitch of the waveform becomes less clear and the concept of "fundamental frequency" or f_0 must be considered. The goal of a f_0 estimator is to find f_0 in the midst of the other harmonically related components of the sound.

The difficulty of finding the f_0 of a waveform depends on the waveform itself. If the waveform has few higher harmonics or the power of the higher harmonics is small, the f_0 is easier to detect, as in Figures 1 and 2. If the harmonics have more power than the f_0, then the period is harder to detect, as in Figures 3 and 4. Figure 4 is an example of the phenomenon of the missing fundamental.

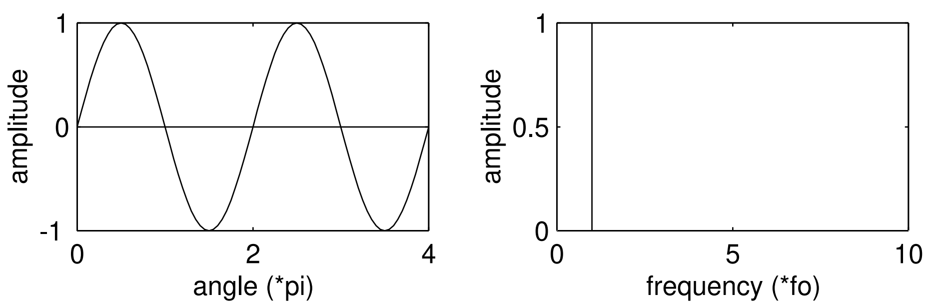

Figure 1: Waveform with no upper harmonics.

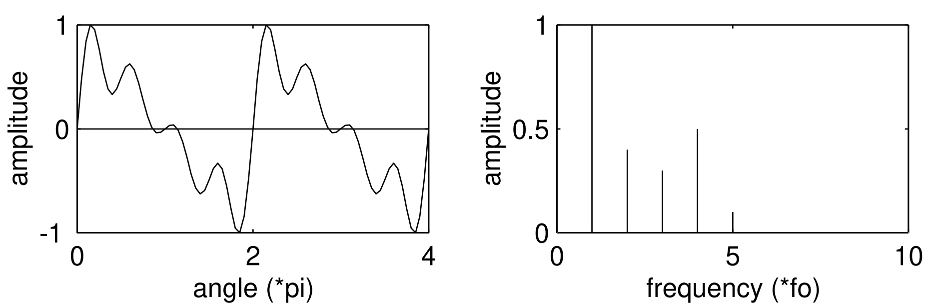

Figure 2: Waveform with lower power upper harmonics.

The next sections of this report discuss three general domains of f_0 estimation algorithms, organized by the type of input and the processing paradigm. Time domain methods are presented first, as they are

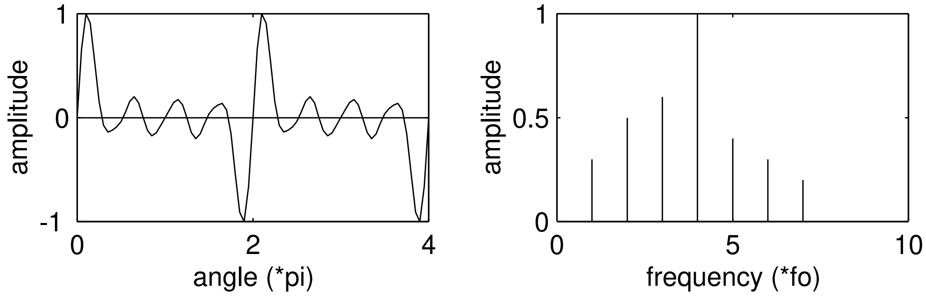

Figure 3: Waveform with higher power upper harmonics.

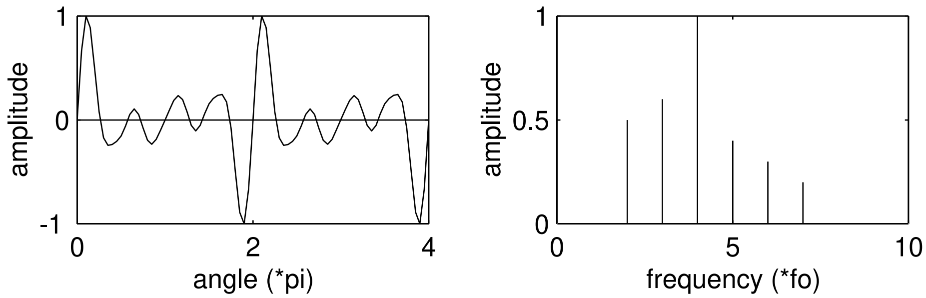

Figure 4: Waveform with high power upper harmonics and no fundamental.

usually computationally simple. Frequency domain methods, presented next, are usually more complex. Statistical methods use probability theory to aid in a decision. After this, Section 7 discusses improvements that can be applied to any f_0 estimation algorithm, and Section 8 presents a comparison and evaluation of some freely available algorithms. The report concludes with a discussion.

4 Time-Domain Methods

The most basic approach to the problem of f_0 estimation is to look at the waveform that represents the change in air pressure over time, and attempt to detect the f_0 from that waveform.

4.1 Time-Event Rate Detection

There is a family of related time-domain f_0 estimation methods which seek to discover how often the waveform fully repeats itself. The theory behind these methods is that if a waveform is periodic, then there are extractable time-repeating events that can be counted, and the number of these events that happen in a second is inversely related to the frequency. Each of these methods is useful for particular kinds of waveforms. If there is a specific time-event that is known to exist once per period in the waveform, such as a discontinuity in slope or amplitude, it may be identified and counted in the same way as the other methods.

Zero-crossing rate (ZCR). Since it was made popular in [15], the utility of the zero-crossing rate has often been in doubt, but lately it has been revived. Put simply, the ZCR is a measure of how often the waveform crosses zero per unit time. The idea is that the ZCR gives information about the spectral content of the waveform.

One of the first things that researchers used the ZCR for was f_0. The thought was that the ZCR should be directly related to the number of times the waveform repeated per unit time. It was soon made clear that there are problems with this measure of f_0 [22]. If the spectral power of the waveform is concentrated around f_0, then it will cross the zero line twice per cycle, as in Figure 5a. However, if the waveform contains higher-frequency spectral components, as in Figure 5b, then it might cross the zero line more than twice per cycle. A ZCR f_0 detector could be developed with initial filtering to remove the higher partials that contaminate the measurement, but the cutoff frequency needs to be chosen carefully so as not to remove the f_0 partial while removing as much high-frequency information as possible. Another possibility for the ZCR f_0 detector would be to detect *patterns* in the zero-crossings, and hypothesize a value for f_0 based on these patterns.

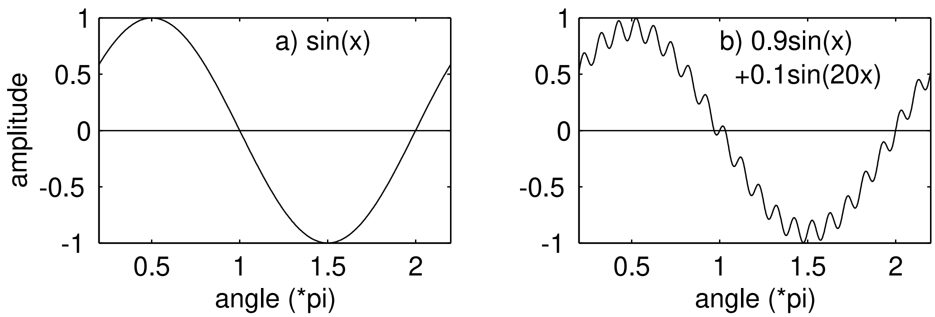

Figure 5: Influence of higher harmonics on zero crossing rate. (after [22])

It has since been shown that ZCR is an informative feature in and of itself, unrelated to how well it tracks f_0. Many researchers have examined statistical features of the ZCR. For example, [25] uses the ZCR as a correlate of the spectral centroid, or balance point, of the waveform, which, unless the spectrum is bimodal, is often the location of most of the power in the waveform. If the spectral centroid is of fairly high frequency, it could mean that the signal is a fricative, or an unvoiced human speech phoneme.

The ZCR has been used in the context of f_0 estimation as recently as [23], where the mean and the variance of the zero crossing rate were calculated to increase the robustness of a feature extractor. The feature is used to track the constancy of the f_0 across time frames. If the waveform is steady-state or slowly varying, as is the case in most pseudo-periodic musical signals, the mean and variance of the ZCR will be consistent over the course of a note, and thus this feature can be used to detect note boundaries, glissade and frequency modulation effects.

Peak rate. This method counts the number of positive peaks per second in the waveform. In theory, the waveform will have a maximum value and a minimum value each cycle, and one needs only to count these maximum values (or minimum values) to determine the frequency of the waveform. In practice, a local peak detector must be used to find where the waveform is locally largest, and the number of these local maxima in one second is the frequency of the waveform, unless each period of the waveform contains more than one local maximum. Similar alternatives are available for this method as are available for the zero-crossing rate detector—the distance between the local maxima gives the wavelength which is inversely proportional to the frequency.

Slope event rate. If a waveform is periodic, the slope of the waveform will also be periodic, and peaks or zeros in the slope can be extracted in the same way as the ZCR. In some cases, zeros or peaks in the slope might be more informative than zeros or peaks in the original waveform, or the detection of these events might be more robust, depending on the domain of the signal.

4.1.1 Discussion

The major difficulty with time-event rate detection methods is that spectrally complex waveforms rarely have just one event per cycle. Waveforms with rich harmonic spectra may cross zero many times or have many peaks in a cycle (Figure 5).

There are some positive aspects of time-event rate detection algorithms. These methods are exceedingly simple to understand and implement, and they take very little computing power to execute. If the nature of the signal is known, a method can be implemented which is tailored to the waveform, reducing the error. Peak counters have been the implementation of choice for hardware frequency-detectors for may years, because the circuit is very simple, and coupled with a simple low-pass filter, provides a fairly robust module.

4.2 Autocorrelation

The correlation between two waveforms is a measure of their similarity. The waveforms are compared at different time intervals, and their "sameness" is calculated at each interval. The result of a correlation is a measure of similarity as a function of time lag between the beginnings of the two waveforms. The *auto*correlation function is the correlation of a waveform with itself. One would expect exact similarity at a time lag of zero, with increasing dissimilarity as the time lag increases. The mathematical definition of the autocorrelation function is shown in Equation 1, for an infinite discrete function $x[n]$, and Equation 2 shows the mathematical definition of the autocorrelation of a finite discrete function $x'[n]$ of size N.

$$R_x(\nu) = \sum_{n=-\infty}^{\infty} x[n]x[n+\nu] \tag{1}$$

$$R_{x'}(\nu) = \sum_{n=0}^{N-1-\nu} x'[n]x'[n+\nu] \tag{2}$$

The cross-correlation between two functions $x[n]$ and $y[n]$ is calculated using Equation 3:

$$R_{xy}(\nu) = \sum_{n=-\infty}^{\infty} x[n]y[n+\nu]. \tag{3}$$

Periodic waveforms exhibit an interesting autocorrelation characteristic: the autocorrelation function itself is periodic. As the time lag increases to half of the period of the waveform, the correlation decreases to a minimum. This is because the waveform is out of phase with its time-delayed copy. As the time lag increases again to the length of one period, the autocorrelation again increases back to a maximum, because the waveform and its time-delayed copy are in phase. The first peak in the autocorrelation indicates the period of the waveform.

Problems with this method arise when the autocorrelation of a harmonically complex, *pseudo*periodic waveform is taken. One can imagine the output of an autocorrelation applied to the waveform in Figure 5b.

The first peak would not be at the period of the full waveform, but at the period of the 20th harmonic overtone. The first "large" peak would indeed occur at the fundamental period of the waveform, but it reduces the robustness and increases the computational complexity to have the algorithm try to distinguish between "large" and "small" peaks.

4.2.1 The YIN Estimator

The YIN f_0 estimator [3], developed by Alain de Cheveigné and Hideki Kawahara, is named after the oriental yin-yang philosophical principal of balance, representing this author's attempts to balance between autocorrelation and cancellation in the algorithm. The difficulty with autocorrelation techniques has been that peaks occur at sub-harmonics as well, and it is sometimes difficult to determine which peak is the fundamental frequency and which represent harmonics or partials. YIN attempts to solve these problems by in several ways.

YIN is based on the difference function, which, while similar to autocorrelation, attempts to *minimize the difference* between the waveform and its delayed duplicate instead of *maximizing the product* (for autocorrelation). The difference function is presented in Equation 4.

$$d_t(\tau) = \sum_{j=1}^{W} (x_j - x_{j+\tau})^2 \tag{4}$$

In order to reduce the occurrence of subharmonic errors, YIN employes a cumulative mean function which de-emphasizes higher-period dips in the difference function:

$$d'_t(\tau) = \begin{cases} 1, & \tau = 0 \\ \frac{d_t(\tau)}{\frac{1}{\tau}\sum_{j=1}^{\tau} d_t(j)} & \text{otherwise} \end{cases} \tag{5}$$

Other improvements in the YIN f_0 estimation system include a parabolic interpolation of the local minima, which has the effect of reducing the errors when the period estimation is not a factor of the window length used (in this case, 15 ms). For a more complete discussion of this method, including computational implementation and results, see the cited paper.

4.3 Phase Space

The phase space signal representation is a way of observing the short-time history of a waveform in a way that makes repetitive cycles clear. The basic phase space representation is to plot the value of the waveform at time t versus the slope of the waveform at the same point [12]. A periodic signal should produce a repeating cycle in phase space, returning to a point with the same value and slope. Higher dimension phase space representations plot the value and $n-1$ derivatives of the signal in n dimensions.

Pseudo-phase space, also called embedded representation, is a simpler form of phase space. The value of the incoming waveform is plotted against a time-delayed version of itself. The representation plots the points $(x, y) = (f(t), f(t-\tau))$, and in the n-dimensional case, $(x_0, x_1, \ldots, x_{n-1}) = (f(t), f(t-\tau_1), \ldots, f(t-\tau_{n-1}))$. Often, for simplicity, $\tau_k = k\tau_1$.

In the remainder of this discussion, "phase space" refers to the general class of representations that include multi-dimensional phase space and pseudo-phase space representations, unless otherwise stated. For a more detailed discussion of a theoretical phase space f_0 estimator, see [11, 27, 28].

4.3.1 Phase Space and Frequency

Any periodic signal forms a closed cycle in phase space, and the shape of the cyclic path depends on the harmonic composition of the signal. The f_0 of a signal is related to the speed with which the path completes the cycle in phase space. The task then becomes detecting the difference between new values in phase space crossing the old path, and new values intersecting and re-tracing the old path. The simplest solution would be to compare distances between points in phase space, and detect when the distance becomes minimal. An initial point would be selected, and the distance from that point would be traced as a function of time. When this distance became zero (or a minimum value) the waveform may have repeated.

This solution is akin to the problem of zero-crossing rate detection, with the associated problems. The phase space cycle might be retracing itself, or it might be crossing itself. It is clear that a simple distance measure will not be sufficient to measure the repetition rate. The distance in higher dimensions might yield a better result—it is conceivable that paths which overlap in two-dimensional space will not overlap in higher dimensions. The question to ask is how many dimensions are required to ensure that the only time the path of the signal intercepts itself is when it begins to repeat itself. The answer to this question will depend on the type of data being investigated, but for band-limited periodic signals, this dimension will be finite. A proof of this statement follows.

rmThis theorem and proof are new.

Theorem 4.1. *Given a band-limited periodic signal, a phase space representation can be constructed requiring a finite number of dimensions.*

Proof. A band-limited signal can be represented as a discrete time series sampled at twice the maximum frequency of the signal (shanon). Since the given signal is periodic, the corresponding time series can be represented by a finite number of samples, repeated infinitely many times. Consider a time series f of n samples. For this series, n difference measures d can be made for each sample, corresponding to the first n derivatives of the continuous signal. For $f(0)$, these are:

$$\begin{aligned} d_1(0) &= f(1) - f(0), \\ d_2(0) &= f(2) - f(0), \\ &\dots \\ d_n(0) &= f(n) - f(0). \end{aligned}$$

No further difference measures can be made since for the periodic signal, $f(n+1) = f(0)$, and $d_{n+1}(0) = f(n+1) - f(0) = f(0) - f(0) = 0$. In general, $d_{n+1}(k) = f(k+n+1) - f(k) = f(k) - f(k) = 0$, and differences above d_{n+1} cycle back to the values of the original differences.

Since the number of difference measures is finite, the number of dimensions required to define them is also finite, and the set of n differences represents a unique point in the n-dimensional space, which will be passed through only once per cycle. □

It is important to note that this proof amounts to a sufficient condition: It is possible to fully represent the phase space of all derivatives in a finite-dimensional hyperspace. It may not be necessary to use all of these dimensions to fully represent the waveform in a non-intersecting hyperspace path. If the signal is band-limited, fewer dimensions are necessary, and in the degenerate case of a sinusoid, only two dimensions are necessary to fully represent the cycle in a non-intersecting hyperspace path. If the amplitude and first derivative of a sinusoid are plotted against each other, the result is a circle.

While the number of dimensions may be finite, the window size must be kept small. Otherwise, the dimensionality of the fully represented phase space will be unwieldy. If the window size is smaller than a complete cycle of the periodic waveform, there would be insufficient information to determine the frequency.

4.3.2 Phase Space of Pseudo-Periodic Signals

A bigger problem with phase space f_0 estimation is how to deal with *pseudo*-periodic signals. In a phase-space representation, the path of a pseudo-periodic signal will never re-trace itself, although it will follow a closely parallel path.

A *Poincaré section* of a phase space plot is a lower-dimensional orthogonal slice through the plot which produces a cross-section of a path being considered. A Poincaré section of a periodic signal will be one or more discrete points, indicating the locations that the path intersects the section.

A pseudo-periodic signal will generate a cloud of points in a Poincaré section, localized in one or more clusters. If these clusters are separate, the mean location of each cluster can be treated as the intersection point for that cluster, and the period can be calculated by the time lag between successive points in the same cluster.

A problem arises when two clusters of points are close together, such that for some points it is not clear which cluster they should belong to. In this case, higher-dimension phase-space representations should be employed until the clusters are shown to be disjoint. There are many potential problems with this suggested method, but it may provide another alternative to the many f_0 estimation algorithms that are currently available.

5 Frequency-Domain Methods

There is much information in the frequency domain that can be related to the f_0 of the signal. Pitched signals tend to be composed of a series of harmonically related partials, which can be identified and used to extract the f_0. Many attempts have been made to extract and follow the f_0 of a signal in this manner.

5.1 Component Frequency Ratios

As early as 1979, Martin Piszczalski was working on a complete automatic music transcription system, the first step of which would be pitch detection [20, 21]. His system would extract the pitch of the signal (assuming that a single note was present at each point in time) and then find note boundaries, infer pitch key, and present a score.

Piszczalski's original procedure began with a spectral transform and identification of the partials in the signal, using peak detection. For each pair of these partials, the algorithm finds the "smallest harmonic numbers" that would correspond to a harmonic series with these two partials in it. As an example, if the two partials occurred at 435 Hz and 488 Hz, the smallest harmonic numbers (within a certain threshold) would be 6 and 7, respectively. Each of these harmonic number pairs are then used as a hypothesis for the fundamental frequency of the signal. In the previous example, the pair of partials would correspond to a hypothesis that the fundamental frequency of the signal is about 70 Hz. After all pairs of partials are considered in this way, the hypothesis most strongly suggested by the pairs of partials is chosen as the fundamental frequency. Some pairs of partials are weighted higher, meaning that their "vote" for the fundamental frequency of the signal counts for more than other pairs of partials. The weighing factor depends on the amplitude of the signals—higher amplitude pairs are counted more than lower amplitude pairs.

This method does not require that the fundamental frequency of the signal be present, and it works well with inharmonic partials and missing partials.

Dorken and Nawab presented an improvement to Piszczalski's method in [5]. They suggest "conditioning" the spectrum using a method they had previously used for principal decomposition analysis. This conditioning had the effect of identifying the frequency partials more accurately, and hence making the entire transform more accurate. Another improvement that they propose is to perform the entire transform in a constant-Q domain, making lower-frequency partials better defined, in an attempt to make the transform closer to human perception.

5.2 Filter-Based Methods

Filters are used for f_0 estimation by trying different filters with different centre frequencies, and comparing their output. When a spectral peak lines up with the passband of a filter, the result is a higher value in the output of the filter than when the passband does not line up.

5.2.1 Optimum Comb Filter

The optimum comb f_0 estimator [19] is a robust but computationally intensive algorithm. A comb filter has many equally spaced pass-bands. In the case of the optimum comb filter algorithm, the location of the passbands are based on the location of the first passband. For example, if the centre frequency of the first passband is 10 Hz, then there will be narrow pass-bands every 10 Hz after that, up to the shanon frequency.

In the algorithm, the input waveform is comb filtered based on many different frequencies. If a set of regularly spaced harmonics are present in the signal, then the output of the comb filter will be greatest when the passbands of the comb line up with the harmonics. If the signal has only one partial, the fundamental, then the method will fail because there will be many comb filters that will have the same output amplitude, wherever a passband of the comb filter lines up with that fundamental.

5.2.2 Tunable IIR Filter

A more recent filter-based f_0 estimator suggested in [16], this method consists of a narrow user-tunable band-pass filter, which is swept across the frequency spectrum. When the filter is in line with a strong frequency partial, a maximum output will be present in the output of the filter, and the f_0 can then be read off the centre frequency of the filter. The author suggests that an experienced user of this tunable filter will be able to recognize the difference between an evenly spaced spectrum, characteristic of a richly harmonic single note, and a spectrum containing more than one distinct pitch. The paper also presents suggestions for automating this search procedure, as a computer would be faster at scanning the frequency spectrum and more accurate at identifying the difference between a richly harmonic single note and multiple concurrent notes.

This f_0 estimation method is in some ways similar to the operation of the stroboscope, a tool used by piano tuners. The tool consists of a spinning disk with black and white marks, illuminated by a strobe light. The strobe is connected to a microphone, and emits a pulse of light as the input signal peaks, once per period. The spinning disk can be sped up or slowed down until the disk is illuminated once every rotation. This can be seen when the black and white marks on the disk appear stationary.

5.3 Cepstrum Analysis

Cepstrum analysis is a form of spectral analysis where the output is the Fourier transform of the log of the magnitude spectrum of the input waveform [9]. This procedure was developed in an attempt to make a non-linear system more linear. Naturally occurring partials in a frequency spectrum are often slightly inharmonic, and the cepstrum attempts to mediate this effect by using the log spectrum.

The name cepstrum comes from reversing the first four letters in the word "spectrum", indicating a modified spectrum. The independent variable related to the cepstrum transform has been called "quefrency", and since this variable is very closely related to time [22] it is acceptable to refer to this variable as time.

The theory behind this method relies on the fact that the Fourier transform of a pitched signal usually has a number of regularly spaced peaks, representing the harmonic spectrum of the signal. When the log magnitude of a spectrum is taken, these peaks are reduced, their amplitude brought into a usable scale, and the result is a periodic waveform in the frequency domain, the period of which (the distance between the peaks) is related to the fundamental frequency of the original signal. The Fourier transform of this waveform has a peak at the period of the original waveform.

Figure 6 shows the progress of the cepstrum algorithm. Figure 6b shows the standard spectral representation of a periodic harmonic signal (whistling at A_4). Figure 6c shows the log magnitude spectrum of the same signal. Note the periodicity of both spectra, and the re-scaled nature of the log magnitude spectrum.

The cepstrum method assumes that the signal has regularly-spaced frequency partials. If this is not the case, such as with the inharmonic spectrum of a bell or the single-partial spectrum of a sinusoid, the method will provide erroneous results. As with most other f_0 estimation methods, this method is well suited to specific types of signals. It was originally developed for use with speech signals, which are spectrally rich and have evenly spaced partials.

5.4 Multi-Resolution Methods

An improvement that can be applied to any spectral f_0 estimation method is to use multiple resolutions [10]. The idea is relatively simple: If the accuracy of a certain algorithm at a certain resolution is somewhat suspect, confirm or deny any f_0 estimator hypothesis by using the same algorithm at a higher or lower resolution. Thus, use a bigger or smaller time window to calculate the spectrum. If a frequency peak shows up in all or most of the windows, this can be considered a confirmation of the f_0 estimator hypothesis. However, each new analysis resolution means more computational expense, which is why multi-resolution Fourier analysis is slower than a dedicated multi-resolution transform such as the discrete wavelet transform.

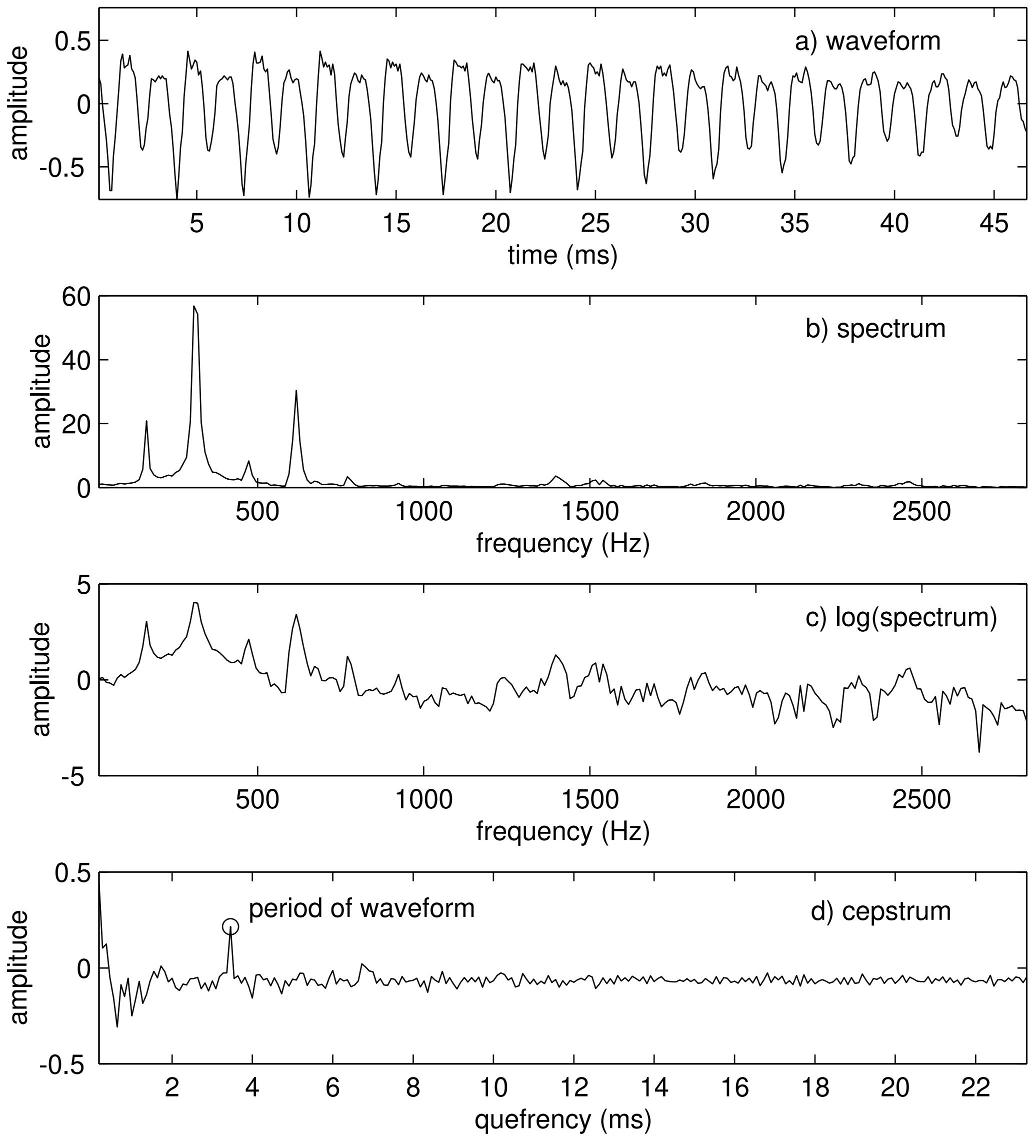

Figure 6: Stages in the cepstrum analysis algorithm.

6 Statistical Frequency Domain Methods

The problem of automatic f_0 estimation can be considered, in some ways, a statistical one. Each input frame is classified into one of a number of groups, representing the f_0 estimator of the signal. Many researchers have thought that modern statistical methods might be applied to the problem of f_0 estimation. Two such methods are presented here.

6.1 Neural Networks

Connectionist models, of which neural nets are an example, are self-organizing pattern matchers, providing a classification output for messy or fuzzy input. Logically, they consist of a collection of nodes, connected by links with associated weights. At each node, signals from all incoming links are summed according to the weights of these links, and if the sum satisfies a certain transfer function, an impulse is sent to other nodes through output links. In the training stage, input is presented to the network along with a suggested output, and the weights of the links are altered to produce the desired output. In the operation stage, the network is presented with input and provides output based on the weights of the connections.

The choice of the dimensionality and domain of the input set is crucial to the success of any connectionist model. A common example of a poor choice of input set and test data is the Pentagon's foray into the field of object recognition. This story is probably apocryphal and many different versions exist on-line, but the story describes a true difficulty with neural nets. As the story goes, a network was set up with the input being the pixels in a picture, and the output was a single bit, yes or no, for the existence of an enemy tank hidden somewhere in the picture. When the training was complete, the network performed beautifully, but when applied to new data, it failed miserably. The problem was that in the test data, all of the pictures that had tanks in them were taken on cloudy days, and all of the pictures without tanks were taken on sunny days. The neural net was identifying the existence or non-existence of sunshine, not tanks.

A connectionist model for the recognition of pitch might take as input a set of spectral partials, or the time-domain waveform, or the phase space representation of the signal. It would likely output a frequency hypothesis, which could then be translated to pitch.

Another approach to using connectionist models for f_0 estimation is the modeling of the human auditory system, as in [24], where the authors present a neural network model based on the cochlear mechanisms of the human ear. Other neural network models could be based on the functioning of the neural pathways (although a good model of this activity has not yet been developed) or could be based on the psychological reaction to pitch. Whatever the case, for a connectionist model, input domain and training data must be chosen carefully.

Another problem with connectionist models is that even if a good model is found, it does not provide any understanding of how the problem is solved. All of the algorithmic information in the model is stored in the weights of the connections, and in large models with thousands or millions of connections, it is prohibitively complicated to translate these weights into a description or algorithm. One must be happy with the "black box" doing what it does without knowing why or how.

6.2 Maximum Likelihood Estimators

Boris Doval and Xavier Rodet have presented a series of papers on f_0 estimation using maximum likelihood estimators [6, 7]. This statistical technique compares different variable value hypotheses based on the likelihood of their being correct in context with the past values of these variables. The intent is to recognize and deal with the slight inharmonicity of naturally occurring frequency partials in a pitched signal.

The model they present is set up as follows: an observation O consists of a set of partials in a short-time Fourier transform representation of a sound. Each observation is assumed to have been produced by a sound with a particular fundamental frequency f_0, and each spectrum contains other information including inharmonic and non-sinusoidal partials (noise). This model is a simplification of the general sound model, assuming that a sound consists primarily of harmonic partials at integer multiples of f_0, with a minority of inharmonic partials and noise.

For a set of candidate fundamental frequencies, the algorithm computes the probability (likelihood) that a given observation was generated from each f_0 in the set, and finds the maximum. The choice of the set of fundamental frequencies is important, because theoretically, the observation could originate from *any* f_0.

7 General Improvements

Most of the models described can be improved by pre-processing the input, reducing the input domain, or by increasing the frequency or time resolution of the input depending on whether the input data is time or frequency information. There are two more major improvements that can be employed by most of these methods, and these are described below.

7.1 Human Auditory Modeling

Because pitch detection (and hence f_0 estimation) is, by its nature, a perceptual problem, any algorithm designed specifically for pitch should be able to be improved by adding some characteristics of the human auditory system. A simple improvement that can be added to any frequency-domain method is to use a constant-Q spectral transform instead of a basic Fourier spectrum. A constant-Q transform is more computationally demanding, but is more faithful to the human auditory perceptual system.

Two factors must be considered when deciding whether or not to use human auditory modeling. First, the application for which the detector be used. If the goal is simply to detect the fundamental frequency of the signal without consideration of the pitch, human perceptual factors are probably not very important. However, if the goal is to detect the pitch for a transcription application, human factors are more relevant. The second factor is computational complexity. Human auditory modeling often results in a significant increase in the computation time required for the application. If computation time is a domain constraint, it may be necessary to forego auditory modeling in favor of a method which is faster but less physiologically accurate.

If properties of the human auditory system are to be used in any application, including f_0 estimation, we must first understand the human perceptual system much better than we currently do. Presently, the most we can do is make the computer system provide the same type of results that the human system does, and hope that these improvements will make the system more accurate and robust.

7.2 Frequency Estimator Tracking

An improvement that several researchers have implemented, applicable to any f_0 estimation algorithm, is tracking [7]. A f_0 estimation based on a single spectral window, no matter how high the resolution of the spectral representation or how robust the algorithm, is the f_0 estimation of a single frame of time. The human system tracks the pitch of an incoming waveform, allowing us to identify such phenomena as *glissandi* (a smooth transition from one pitch to another) and pitch intervals. A time window containing a definite pitch of a small number of cycles is often very difficult for a human to identify [22], but when many time windows are played one after another, a sensation of pitch becomes apparent.

A simple modification to a f_0 estimation algorithm which can improve performance without increasing the computational burden is to give preference to f_0 hypotheses that are close to the f_0 hypothesis of the last time frame. Storing the f_0 hypothesis of the n previous time frames requires only n more memory words, and the comparison to the present hypothesis is a simple operation for each past time frame considered.

A more involved comparison method is the use of hidden Markov models (HMMs), statistical models which track variables through time [4]. These models have been used for linguistics and circuit theory as well as f_0 estimation. HMMs are state machines, with a hypothesis available for the output variable at each state. At each time frame, the HMM moves from the current state to the most likely next state, based on the input to the model and the state history which is represented in the current state. The state transition properties of HMMs are calculated using input-output pairs, consisting of (in the case of f_0 estimation) a set of spectral windows (or a set of spectral partials) and the corresponding best f_0 hypothesis.

8 Evaluation of Implementations

Because there has been much f_0 estimation research lately, many researchers have designed and implemented their own f_0 estimators, and some have made these available to the wider research community. Using an off-the-shelf f_0 estimator is a good place to start because the algorithm is already implemented, and the researcher can begin immediately by analyzing results of the algorithm and designing add-on or sub-feature analysis components. One drawback is that the algorithm has been designed for a particular research problem and might not be appropriate for the problem at hand, although the algorithm could me modified to apply more closely if needed.

8.1 Common Problems with Fundamental Frequency Estimators

When a signal is pseudo-periodic with a low-power fundamental, it is possible to mistake an upper harmonic for the fundamental. Humans do this as well, and it is a result more of the signal itself than of the recognition algorithm. A period-k signal can become a period-$2k$ signal through a process called period doubling [26, 13]. At the transition point, it is unclear whether it is appropriate to count the period as k or $2k$. This transition point is unstable, so it is uncommon to hear signals of ambiguous pitch in nature. However, it does indicate that period doubling errors may be a difficult problem to overcome.

Subharmonic errors can lead to misleading results because they often occur within the context of a single pitch event, causing the f_0 estimation to jump back and forth between two (or more) subharmonics of the "true" fundamental frequency. The challenge then is to improve the f_0 estimation algorithm to deal with these problems. The YIN improvements attempt to rectify subharmonic errors, and have some success over less computationally complex algorithms.

8.2 Off-the-Shelf Estimators

For this report, four off-the-shelf f_0 estimators are evaluated and compared. The first two f_0 estimators are part of a speech analysis software package called Colea, developed by Philip Loizou [17] for the MATLAB programming environment. This package contains tools for analyzing speech using f_0 estimation, formants, and spectral content. There are two f_0 estimators built into this package, one based on autocorrelation and one based on the cepstrum. The third off-the-shelf f_0 estimator is the YIN algorithm described in Section 4.2.1.

The fourth is Terez's implementation of an embedded representation f_0 estimator described in Section 4.3. Terez's embedded representation was available only in compiled format, so the testing is not as exhaustive, but it is sufficient to negate the claims he makes that the method completely solves the pitch detection problem. While it may be true that Terez's method has particular advantages for particular classes of signals, the algorithm is similar in many ways to traditional autocorrelation techniques and so is susceptable to the same types of errors. Figures 7 and 8 show the output of Terez's pitch estimator for two typical sounds from the signing portion of the corpus. No further testing is done on this estimator.

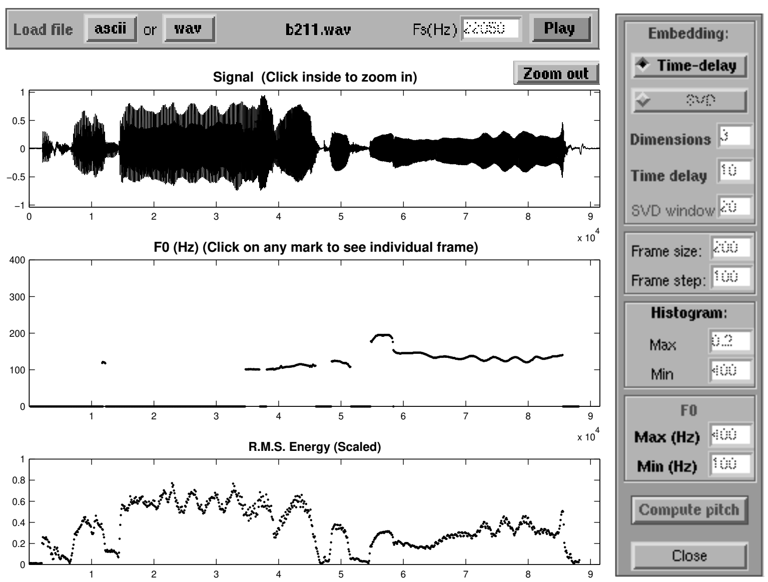

Figure 7: Terez's algorithm with existence errors (singing).

8.3 Evaluation

The three remaining f_0 estimators were tested on the speech/song corpus and the f_0 estimations were compared. Since the f_0 estimations were based on different frame rates, the first task was to match the f_0 estimations on a normalized time scale by interpolating between the frame measurements of the f_0 estimations to match the highest frame rate. Figure 9 shows an example of the three f_0 estimation techniques compared on a common scale. Figure 10 shows an example of a situation where the three f_0 estimation techniques did not agree.

These three f_0 estimators were compared using two criteria:

- Consistency between detectors
- Visual inspection of results

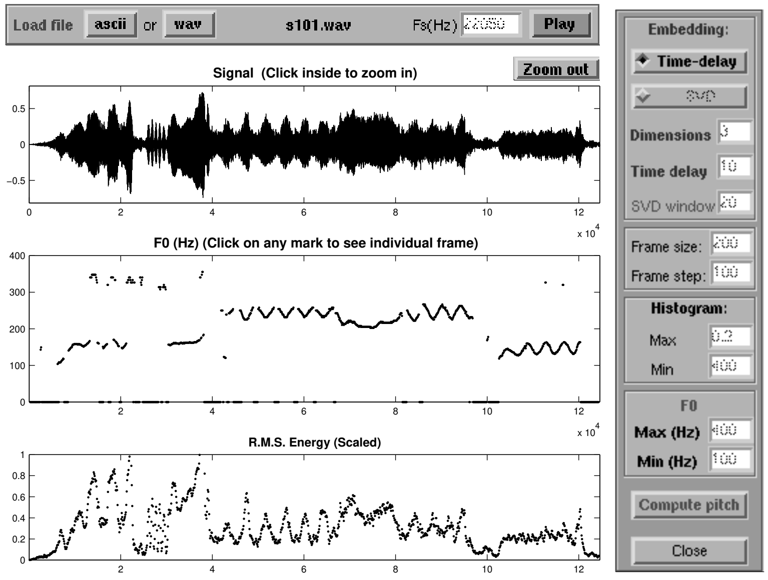

Figure 8: Terez's algorithm with subharmonic errors (singing).

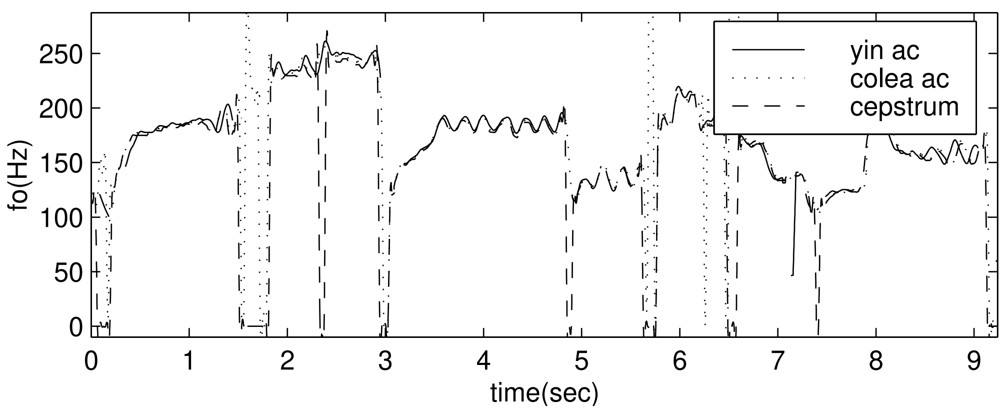

Figure 9: Comparison of three f_0 estimation methods - all methods near agreement (file b226).

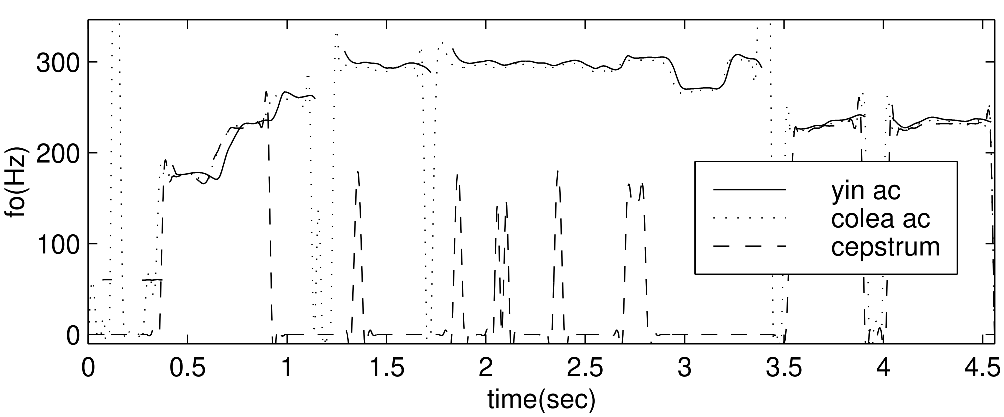

Figure 10: Comparison of three f_0 estimation methods showing differences among methods(file b212).

It should be noted that consistency between detectors, as an evaluation technique in isolation, is not particularly rigorous. It is not unreasonable to expect two detectors to agree on an erroneous f_0 estimation. This evaluation method becomes acceptable when combined with visual inspection. Files with one method in disagreement with the other two were inspected visually, and the f_0 estimations were compared to the perceived pitch track. In instances where two methods agreed, in the majority of cases, visual inspection showed that the agreeing methods were correct and the method not in agreement was in error.

A further comparison method could be to generate a manual (and presumably accurate) f_0 track for each file and compare these tracks to the results generated by each method. This evaluation technique was considered too labour-intensive for this work, and the results gained from the three presented criteria are sufficient for a comparison among the methods. If f_0 track accuracy were of paramount importance in comparing the methods (as in a transcription project), annotated corpora currently exist with Electroglottogram (EGG) f_0 track targets which could be used to evaluate the f_0 accuracy of the method. Synthetic signals have often been used to test f_0 detectors, although care must be taken to use synthetic signals that closely resemble the real-world signals that the estimator is likely to encounter.

Relative accuracy is a sufficient measure for this work because this provides an evaluation of the kinds of errors we are interested in, being subharmonic errors and existence errors. Subharmonic errors are described in Section 8.1. Existence errors are generated when a pitched frame in a sound is considered by the detector to not have a pitch, or when a f_0 hypothesis is presented for an un-pitched frame. The three criteria used here are sufficient for evaluations based on these measures.

The first criterion is measured by finding the difference between each pair of f_0 tracks. For each file, the difference between the three f_0 estimations are calculated according to Equation 6:

$$D = \frac{1}{N} \sum_{P_1^v, P_2^v} |P_1 - P_2|, \tag{6}$$

where N is the length of the f_0 track and P^v is a notation for the valid portions of a f_0 track P. The mean difference over the set of files is calculated, and the results for the entire corpus as well as the talking files only and the singing files only are presented in Table 1.

Table 1: Mean f_0 estimation difference between three f_0 estimators.

	YIN / Colea AC	YIN / Colea Cepstrum	Colea AC / Cepstrum
All files	13.33 Hz	19.68 Hz	41.22 Hz
Singing files	11.11 Hz	17.80 Hz	31.83 Hz
Talking files	14.00 Hz	20.27 Hz	43.75 Hz

It can be seen from these results that the two f_0 estimation techniques based on autocorrelation had more similar results than the f_0 estimator based on cepstrum. This is perhaps to be expected, since the base algorithm is the same. This measure is enough to support the hypothesis that the Colea cepstrum f_0 estimator is not as accurate as the two autocorrelation f_0 detectors.

It is also important here to look at how the f_0 estimators compared in specific tasks. Since f_0 estimators are usually designed for a specific task, one would expect the f_0 estimator to perform better for that task (e.g. the estimation of the pitch of speech) than another task (e.g. the estimation of the pitch of song). Table 1 shows that the difference between the autocorrelation methods is lower for singing files than for talking files, but the difference between the cepstrum method and the two autocorrelation methods is higher for singing files than for talking files.

The second evaluative criterion is visual inspection of the three f_0 tracks. Files were selected with low and high relative error rates, and these were visually inspected for consistency errors. An example of a file with high difference is in Figure 10, where the cepstrum f_0 estimator failed to detect the f_0 of the signal through the time range of about 1 second to 3.5 seconds. The two autocorrelation methods agree well on this sample. It is important to notice the slight delay between the two autocorrelation f_0 tracks. This is because of the extra processing steps in the YIN detector, which seem to introduce a slight delay in the measured f_0 track. This can be corrected by re-aligning the f_0 track with the time scale of the original utterance, but this again is another computational step.

The visual inspection provides no rigorous results, although a count could be made of the files in which subharmonic and existence errors occurred, and the detector responsible for the error. The results that were provided by the visual inspection are that in most cases, differences between the f_0 tracks are due to errors in the cepstrum f_0 track, especially in singing utterances. Subharmonic errors show up between the autocorrelation f_0 estimators, and these errors are more or less equally distributed among the YIN, Colea and/or both.

It should be noted that the Colea f_0 estimators provided no measure of the confidence of the f_0 estimation. When the utterance is non-periodic, the f_0 estimation becomes erratic, jumping to zero or to a higher value out of range. These jumps, combined with the application of a power threshold, can be used to detect the presence or absence of f_0 which is an important feature in the speech/song comparison. The confidence metric of the YIN estimation means that this extra post-processing is not required.

9 Discussion

f_0 estimation algorithms tend to be based on a number of fairly strict assumptions:

1. The input waveform consists of a single pitched signal, segmented into frames, and the waveform is homogeneous throughout the time frame being considered.
2. The input is limited to a specific audio domain, for which the algorithm is designed.
3. f_0 estimation is the same thing as pitch detection.

These assumptions are acceptable for initial development, and many successful algorithms have been developed using these assumptions. Indeed, without severely limiting the domain at the beginning of research, it would be impossible to achieve anything at all. Many researchers who accept that assumption 3 is theoretically incorrect continue to cite their work as pitch detectors rather than f_0 estimators. Given the slightly non-logarithmic transfer function from frequency to pitch, and also given some considerations about the base frequency used to create the music (e.g. $A_4 = 440$ Hz), a simple transformation can be developed to accurately map the frequency of a signal to its musical pitch.

Assumption 2 is another necessity for the introductory design of an algorithm. As the algorithms become more robust and more accurate, the domain for which the algorithm is useful will expand until assumption 2 can perhaps be relaxed. It is equally possible, however, that the nature of audio signals is such that certain algorithms are good for certain input and not others, and there is no “silver bullet” algorithm that will handle every periodic input without error. It is even conceivable that the human perceptual system uses more than one analysis method for deducing pitch from the vibrations of the eardrum.

This leaves assumption 1. f_0 estimation of *multiple* auditory streams is not difficult for the human auditory system, although it can be difficult to concentrate on more than one stream at a time. Work on auditory stream separation is proceeding, but it would perhaps be more fruitful if the f_0 estimation community would work with the stream separation community, and vice versa. Clearly, each has much to learn from the other.

References

[1] Albert Bregman. *Auditory Scene Analysis*. MIT Press, Cambridge, 1990.

[2] Stanley Coren, Lawrence M. Ward, and James T. Enns. *Sensation and Perception*. Harcourt Brace College Publishers, Toronto, 1994.

[3] Alain de Cheveigné and Hideki Kawahara. Yin, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, 111(4), 2002.

[4] Phillipe Depalle, Guillermo García, and Xavier Rodet. Tracking of partials for additive sound synthesis using Hidden Markov Models. In *International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 225–228. IEEE, 1993.

[5] Erkan Dorken and S. Hamid Nawab. Improved musical pitch tracking using principal decomposition analysis. In *International Conference on Acoustics, Speech and Signal Processing*, volume II, pages 217–220. IEEE, 1994.

[6] Boris Doval and Xavier Rodet. Estimation of fundamental frequency of musical sound signals. In *International Conference on Acoustics, Speech and Signal Processing*, pages 3657–3660. IEEE, 1991.

[7] Boris Doval and Xavier Rodet. Fundamental frequency estimation and tracking using maximum likelihood harmonic matching and HMMs. In *International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 221–224. IEEE, 1993.

[8] John M. Eargle. *Music, Sound and Technology*. Van Nostrand Reinhold, Toronto, 1995.

[9] James L. Flanagan. *Speech Analysis, Synthesis and Perception*. Springer-Verlag, New York, 1965.

[10] Edouard Geoffriois. The multi-lag-window method for robust extended-range f_0 determination. In *Fourth International Conference on Spoken Language Processing*, volume 4, pages 2239–2243, 1996.

[11] David Gerhard. Audio visualization in phase space. In *Bridges: Mathematical Connections in Art, Music and Science*, pages 137–144, August 1999.

[12] Vincent Gibiat. Phase space representations of acoustical musical signals. *Journal of Sound and Vibration*, 123(3):537–572, 1988.

[13] James Glieck. *Chaos: Making a New Science*. Penguin, New York, 1987.

[14] Stephen Handel. *Listening*. MIT Press, Cambridge, 1989.

[15] Benjamin Kedem. Spectral analysis and discrimination by zero-crossings. *Proceedings of the IEEE*, 74(11):1477–1493, November 1986.

[16] John E. Lane. Pitch detection using a tunable IIR filter. *Computer Music Journal*, 14(3):46–57, Fall 1990.

[17] Philip Loizou. Colea: A matlab software tool for speech analysis. [Online] Retrieved March 18, 2003, from `http://www.utdallas.edu/~loizou/speech/colea.htm`

[18] Brian C. M. Moore, editor. *Hearing*. Academic Press, Toronto, 1995.

[19] James A. Moorer. On the transcription of musical sound by computer. *Computer Music Journal*, pages 32–38, November 1977.

[20] Martin Piszczalski. *A Computational Model for Music Transcription.* PhD thesis, University of Stanford, 1986.

[21] Martin Piszczalski and Bernard A. Galler. Predicting musical pitch from component frequency ratios. *Journal of the Acoustical Society of America*, 66(3):710–720, September 1979.

[22] Curtis Roads. *The Computer Music Tutorial.* MIT Press, Cambridge, 1996.

[23] Stéphane Rossignol, Xavier Rodet, Jöel Soumagne, Jean-Luc Collette, and Philippe Depalle. Features extraction and temporal segmentation of acoustic signals. In *International Computer Music Conference*, pages 199–202, 1998.

[24] Hajime Sano and B. Keith Jenkins. A neural network model for pitch perception. *Computer Music Journal*, 13(3):41–48, Fall 1989.

[25] Eric Scheirer and Malcolm Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. In *International Conference on Acoustics, Speech and Signal Processing*, volume II, pages 1331–1334. IEEE, 1997.

[26] Manfred R. Schroeder. *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise.* W.H.Freeman, New York, 1991.

[27] Dmitry Terez. Fundamental frequency estimation using signal embedding in state space. *Journal of the Acoustical Society of America*, 112(5):2279, November 2002.

[28] Dmitry Terez. Robust pitch determination using nonlinear state-space embedding. In *International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 345–348, 2002.

[29] Barry Truax, editor. *Handbook for Acoustic Ecology.* A.R.C. Publications, Vancouver, 1978.

[30] William A. Yost, Arthur N. Popper, and Richard R. Fay. *Human Psychophysics.* Springer-Verlag, New York, 1993.