A sawtooth waveform inspired pitch estimator for speech and music

Arturo Camacho, and John G. Harris

Citation: The Journal of the Acoustical Society of America **124**, 1638 (2008); doi: 10.1121/1.2951592
View online: https://doi.org/10.1121/1.2951592
View Table of Contents: https://asa.scitation.org/toc/jas/124/3
Published by the Acoustical Society of America

A sawtooth waveform inspired pitch estimator for speech and music

Arturo Camacho and John G. Harris
Computational NeuroEngineering Laboratory, University of Florida, Gainesville, Florida 32611

(Received 5 December 2007; revised 30 May 2008; accepted 2 June 2008)

A sawtooth waveform inspired pitch estimator (SWIPE) has been developed for speech and music. SWIPE estimates the pitch as the fundamental frequency of the sawtooth waveform whose spectrum best matches the spectrum of the input signal. The comparison of the spectra is done by computing a normalized inner product between the spectrum of the signal and a modified cosine. The size of the analysis window is chosen appropriately to make the width of the main lobes of the spectrum match the width of the positive lobes of the cosine. SWIPE′, a variation of SWIPE, utilizes only the first and prime harmonics of the signal, which significantly reduces subharmonic errors commonly found in other pitch estimation algorithms. The authors' tests indicate that SWIPE and SWIPE′ performed better on two spoken speech and one disordered voice database and one musical instrument database consisting of single notes performed at a variety of pitches.
 [DOI: 10.1121/1.2951592]

PACS number(s): 43.66.Hg [EJS] Pages: 1638–1652

I. INTRODUCTION

Pitch is an attribute of sound that gives important information about its source. In speech, it helps us to identify the gender of the speaker (females tend to have higher pitch than males) and gives additional meaning to words (a set of words may be interpreted as an affirmation or a question depending on the intonation). In music, it determines the names of the notes.

Several definitions of pitch have been proposed. One of them is "pitch is that attribute of auditory sensation in terms of which sounds may be ordered on a musical scale" (American Standards Association, 1960). In practice, however, it is convenient not only to order sounds by their pitch, but also to assign a number to them. The standard method for doing that is to ask a group of listeners to adjust the frequency of a pure tone until its pitch matches the pitch of the target sound, and then define the pitch of the target sound as the frequency of the pure tone that best matched its pitch. Sometimes, complex tones made up of several harmonics are preferred to pure tones for matching.

Pitch estimation has applications in many areas that involve processing of sound. In music, it is used for automatic music transcription (Klapuri, 2004) and query by humming (e.g., Dannenberg *et al.*, 2004). In communications, it is used for speech coding (Spanias, 1994). In speech pathology, it is used to detect voice disorders (e.g., Yumoto *et al.* 1982). In linguistics, it is used to facilitate second language acquisition through the display of intonation patterns (de Bot, 1983).

Pitch estimation has a long history. An extensive review is given by Hess (1983). More recently, in a paper in which we presented a pitch estimator based on a smooth harmonic average peak-to-valley envelope (SHAPE) (Camacho and Harris, 2007), we included a review of some pitch estimators and illustrated some of their problems. Specifically, we showed that (i) algorithms that use the logarithm of the spectrum [e.g., cepstrum (Noll, 1967) and harmonic product spectrum (Schroeder, 1968)] are prone to fail when there are missing harmonics; (ii) algorithms that use the square of the spectrum [e.g., autocorrelation (Sondhi, 1968; Rabiner, 1977)] are prone to fail when there are salient harmonics; (iii) algorithms that give the same weight to all the harmonics (e.g., cepstrum, unbiased autocorrelation, and harmonic product spectrum) are prone to estimate the pitch as one of its subharmonics; (iv) algorithms that analyze the spectrum only at harmonic frequencies are prone to fail for inharmonic signals [e.g., harmonic product spectrum and subharmonic summation (Hermes, 1988)]; and (v) harmonic sieves (Duifhuis *et al.*, 1982), in which a component is accepted as harmonic if it is located within a certain range of a harmonic, are not completely satisfactory, since a slight shift of the component may put it in or out of the range, potentially changing the estimated pitch drastically.

The work presented here is an extension of SHAPE. The only new features are the use of window sizes that are proportional to the pitch period of the candidates, and the multiplication of the first and last negative lobes of the kernel by $1/2$ to avoid a bias that existed in SHAPE. It will be shown that the types of signals for which the algorithm is optimized are periodic signals whose spectral envelope decays inversely proportional to frequency. An example of such a signal is a sawtooth waveform. This type of signal is the one that motivated the name of the algorithm: sawtooth waveform inspired pitch estimator (SWIPE).

We conclude this section with a description of the scope of our work. Our goal is to estimate the pitch of speech and musical instruments, but not the pitch of some synthetic sounds that are traditionally used to test pitch perception models, such as sinusoidally amplitude-modulated noise (Meddis and Hewitt, 1991) and periodic signals with alternating phase harmonics (Meddis and O'Mard, 1997). Furthermore, we do not attempt to explain how the auditory system determines pitch but simply to create a black box that attempts to reproduce human pitch percepts. It is also worth

mentioning that our goal is to determine pitch but not fundamental frequency (defined as the maximum common divisor of its spectral components). In many cases, these two attributes coincide, but that is not always the case. For example, a periodic signal formed by the 13th, 25th, and 29th harmonics of 50 Hz (i.e., 650, 950, and 1250 Hz) is perceived as having a pitch of 334 or 650 Hz (Patel and Balaban, 2001) but not 50 Hz.

II. METHOD

A. Main idea

The main idea of the algorithm is the same underlying idea present in several pitch estimators (e.g., Sun, 2000; Rabiner, 1977; Sondhi, 1968; Noll, 1967): the measurement of the average peak-to-valley distance (APVD) at harmonic locations.[1] The APVD at the kth harmonic of f is defined as

$$\begin{aligned} d_k(f) &= \tfrac{1}{2}[|X(kf)| - |X((k-1/2)f)|] \\ &\quad + \tfrac{1}{2}[|X(kf)| - |X((k+1/2)f)|] \\ &= |X(kf)| - \tfrac{1}{2}[|X((k-1/2)f)| + |X((k+1/2)f)|], \end{aligned} \quad (1)$$

where $X(f)$ is the Fourier transform (FT) of the signal. Averaging over the first n peaks, the global APVD is

$$\begin{aligned} D_n(f) &= \frac{1}{n}\sum_{k=1}^{n} d_k(f) \\ &= \frac{1}{n}\left[\frac{1}{2}|X(f/2)| - \frac{1}{2}|X((n+1/2)f)| \right. \\ &\quad \left. + \sum_{k=1}^{n} |X(kf)| - |X((k-1/2)f)|\right]. \end{aligned} \quad (2)$$

As a first approach, we estimate the pitch as the frequency that maximizes the global APVD. This can be expressed using an integral transform as

$$p = \arg\max_f \int_0^\infty |X(f')| K_n(f,f') df', \quad (3)$$

where

$$\begin{aligned} K_n(f,f') &= \frac{1}{2}\delta(f' - f/2) - \frac{1}{2}\delta(f' - (n+1/2)f) \\ &\quad + \sum_{k=1}^{n} \delta(f' - kf) - \delta(f' - (k-1/2)f). \end{aligned} \quad (4)$$

Notice that the $1/n$ factor was obviated because the argument that maximizes the integral is invariant to scaling factors.

The kernel corresponding to a pitch candidate of 190 Hz and $n=9$ is shown in Fig. 1. The figure also shows the spectrum of a signal having the same pitch. The signal consists of the vowel /u/ passed through a bandpass filter that removed the frequencies outside the range 300–3400 Hz, mimicking telephone speech. Its spectrum exhibits a strong second harmonic that was presumably boosted by a formant close to 380 Hz.

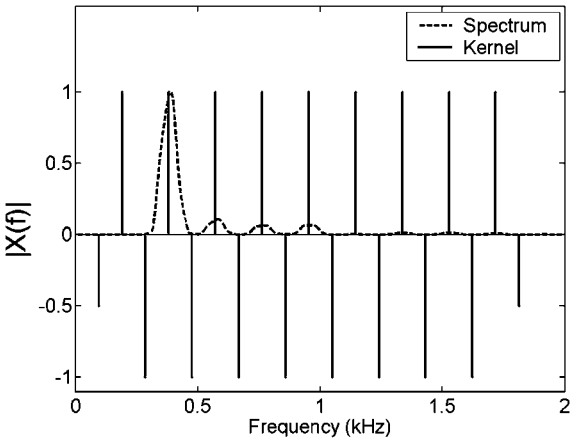

FIG. 1. Average peak-to-valley-distance (APVD) kernel. The APVD kernel has positive pulses at multiples of the fundamental and negative pulses in between. The first and last negative pulses have half the height of the others.

In the next sections, we will refine this first approach, trying to avoid the problem-causing features cited in the Introduction. Specifically, we will warp the spectrum, introduce a decaying weighting factor to the harmonics amplitudes, and replace the pulses with a smoother function.

B. Warping of the spectrum[2]

When we developed SHAPE, we found that using the square root of the spectrum produced better results than using the logarithm, square, or raw spectrum. Consequently, we will use it for SWIPE as well. The reason we believe the square root of the spectrum produces better results than the other functions will be postponed until the next section.

C. Weighting of the amplitude of the harmonics

Giving the same weight to all the harmonics amplitudes may lead to subharmonics of the pitch obtain the same score as the pitch. For example, if the signal consists of a pure tone of frequency f Hz and the same weight is applied to all the harmonic amplitudes, each of the subharmonic pitch candidates $f/2$, $f/3$,…, and f/n Hz, would have the same score as f Hz. To avoid this, we explored the use of the exponentially and harmonically decaying weights shown in Fig. 2. For exponential decays, a weight of r^{k-1} was applied to the kth harmonic amplitude ($k=1,2,\ldots,n$, and $r=0.5$, 0.7, and 0.9), and for harmonic decays, a weight of $1/k^p$ was applied to the kth harmonic amplitude ($k=1,2,\ldots,n$, and $p=1/2$, 1 and 2). In informal tests, the best results were obtained when using harmonic decays with $p=1/2$. Notice that this decay matches the decay of the square root of the average spectrum of vowels (Fant, 1970), which was the spectral warping shown to work best in the previous section. In other words, better pitch estimates were obtained when computing the inner product between the square root of the spectrum and a kernel whose envelope decays as $1/\sqrt{f}$, than when computing the inner products between the spectrum and a kernel whose envelope decays as $1/f$, for example.

The benefit of using a weighting of the harmonics amplitudes of the form $1/\sqrt{k}$ and the square root of the spectrum probably comes from the fact that when the signal has

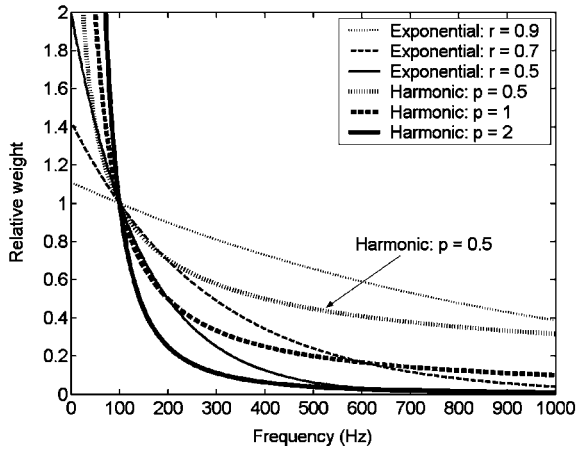

FIG. 2. Weighting of the harmonics amplitudes. Exponentially and harmonically decaying weighting factors of the form r^{k-1} (r=0.5, 0.7, and 0.9) and $1/k^p$ (p=1/2, 1 and 2) were utilized to weight the kth harmonic ($k=1,2,\ldots,n$). The highlighted curve corresponds to the one that produced the best results.

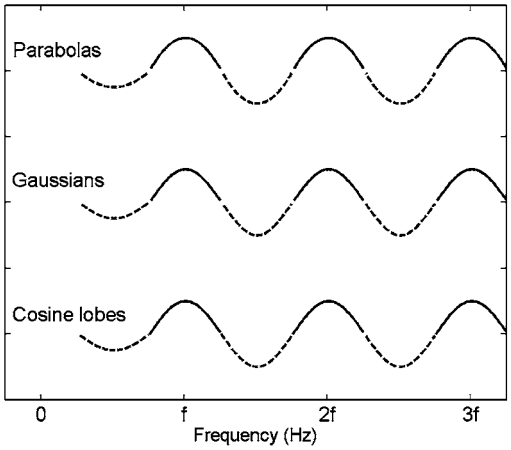

FIG. 3. Explored kernels. Kernels built from concatenations of truncated positive (continuous lines) and negative (dashed lines) parabolas, Gaussians, and cosine lobes were explored. (See the text for details about the selection of the truncation point.)

the expected shape of a vowel (i.e., the amplitudes of the harmonics decay as 1, 1/2, 1/3, etc.), each harmonic contributes to the inner product with a value proportional to its amplitude. If the spectrum has the aforementioned shape, the square root of the harmonics amplitudes will decay as 1, $1/\sqrt{2}$, and $1/\sqrt{3}$, etc., just like the weights of the harmonics amplitudes. This will make the terms in the sum of the inner product decay as 1, 1/2, 1/3, etc., and the relative contribution of each harmonic will be proportional to its amplitude. Conversely, if we compute the inner product over the raw spectrum using a weighting of the form $1/k$, the terms of the sum will be 1, 1/4, 1/9, etc., which are not proportional to the amplitude of the harmonics but to their square. This would make the contribution of the strongest harmonics too large and the contribution of the weakest too small.

D. Blurring of the harmonics

Analyzing the spectrum only at harmonic locations is inconvenient because it does not allow recognizing the pitch of inharmonic signals. To recognize the pitch of these signals, we propose the use of smooth weighting functions that take into account the spectrum not only at the harmonics, but also in their neighborhood.

In our previous paper (Camacho and Harris, 2007), it was shown that the local maxima of the kernel must be strictly concave (i.e., second derivative strictly positive). Consequently, concatenations of positive and negative truncated parabolas, Gaussians, and cosine lobes as shown in Fig. 3 (positive components with continuous lines and negative components with dashed lines) were proposed as kernels.

The criterion used to select the truncation point was the maximization of the smoothness of the concatenation by making as many derivatives continuous as possible. Even though smoothness sounds attractive, the main reason for using this criterion was the uniqueness of the solution for the Gaussian and the cosine: The truncation point has to be the inflection point. For the Gaussian, it occurs at one standard deviation and it warranties continuous first and second derivatives for the concatenation. For the cosine, it occurs at the zero crossing and it warranties all-order continuous derivatives (the concatenation of cosine lobes truncated in that way forms a cosine). In the case of the parabola, it is not possible to make a concatenation of positive and negative parabolas having continuous second and higher order derivatives, so arbitrarily we truncated it at its fixed point $(\pm 1,1)$.

As it can be suspected from the similarity of the kernels in Fig. 3, there were no significant differences in performance between them. However, we preferred the cosine kernel because of its smoothness and simplicity (it can be expressed without using concatenations).

E. Number of harmonics

With respect to the number of harmonics, there are typically two types of algorithms: those that use a fixed number of harmonics and those that use as many harmonics as possible, up to a certain frequency, usually the Nyquist frequency. We explored both possibilities and found that the best results were obtained when using as many harmonics as possible, although going beyond 3.3 kHz for speech and 5 kHz for musical instruments did not increase the performance significantly. Hence, we will use all available harmonics on SWIPE.

F. Warping of frequency

For the purpose of computing the integral of a function, we can think of warping the scale as the process of sampling the function more finely in some regions than others, effectively giving more emphasis to the more finely sampled regions. Since we are computing an inner product to estimate pitch, it makes sense to sample the spectrum more finely in the regions that contribute the most to the determination of pitch. It seems reasonable to assume that these regions are the ones with the most harmonic energy. In the case of speech, and assuming that the amplitude of the harmonics decays as $1/f$, it seems reasonable to sample the spectrum more finely in the neighborhood of the fundamental, and

decrease the granularity as we increase frequency, following the expected $1/f$ pattern. A decrease in granularity should also be performed below the fundamental because no harmonic energy is expected in that region. Unfortunately, the determination of the location of the fundamental is ill-posed, since that is precisely what we want to determine.

We explored frequency scale transformations of the form

$$\phi(f) = C \log(1 + f/\sigma), \tag{5}$$

where the constant factor C is irrelevant and can be set arbitrarily. The explored values of σ were 229, 700, and 0 (the latter in an asymptotical sense). These transformations correspond to the equivalent rectangular bandwidth (ERB) (Glasberg and Moore, 1990), mel, and logarithmic scales, respectively. (Notice also that the Hertz scale corresponds to $\sigma=\infty$.) We also explored the Bark scale given by the formula (Traunmüller, 1990)

$$z(f) = [26.81/(1 + 1960/f)] - 0.53. \tag{6}$$

To compute the inner product between the spectrum and the kernel, we sample both of them uniformly in the transformed scale before computing the inner product. The scale that on average produced the best results on speech was the ERB scale, which is expressed in a base-10 logarithmic scale as

$$\mathrm{ERBs}(f) = 21.4 \log_{10}(1 + f/229). \tag{7}$$

This scale has several desirable characteristics: It approaches a logarithmic behavior as f increases, tends towards a constant (zero) as f decreases, and the frequency at which the transition occurs (229 Hz) is close to the mean fundamental frequency of speech, especially for females (Bagshaw, 1994; Wang and Lin, 2004; Schwartz and Purves, 2004).

For musical instruments, the Hertz scale was the one that produced the best results. An explanation of why this may be the case will be given in Sec. III D.

G. Window type and size

Most common methods of pitch estimation use a fixed window size. This makes the width of the main lobe of each of the spectral components to be fixed as well. This has the disadvantage of making high pitches more likely to obtain high scores than low pitches. The reason for this is illustrated in Fig. 4. Figure 4(a) shows the spectrum of a signal with a pitch of 500 Hz and the kernel corresponding to that pitch, and Fig. 4(b) shows the spectrum of a signal with a pitch of 125 Hz and the kernel corresponding to that pitch. The width of the cosine lobes increases with pitch, but not the width of the spectral lobes. This causes the overall weight given to the main spectral lobes to increase as the pitch increases, and to decrease as the pitch decreases. Actually, below a certain pitch (125 Hz in this example), the weight given to the sides of the main spectral lobes becomes negative (not shown in the figure).

One way to solve this problem is to make the spectral lobes as narrow as possible, but this requires making the window infinitely large, which is undesirable given the changing nature of pitch (signals with constant pitch for eternity do not occur in real life). Another way to solve the problem is to try to make the square root of the main spectral lobes match the cosine lobes for each of the pitch candidates, and then compute the normalized inner product between them, defined as

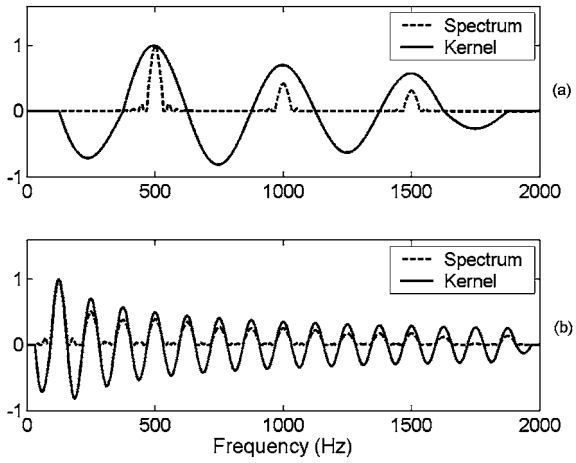

FIG. 4. Similarity between cosine lobes and square root of main spectral lobes. (a) High pitch, low similarity. (b) Low pitch, high similarity.

$$\frac{\int_R \xi(f)\psi(f)df}{(\int_R \xi^2(f)df)^{1/2}(\int_R \psi^2(f)df)^{1/2}}, \tag{8}$$

for any two functions $\xi(f)$ and $\psi(f)$, over a region R. Since the spectrum is non-negative, but not the cosine, the cosine kernel must be normalized using only its positive part in order to obtain a normalized inner product close to 1.[3]

A type of window whose square-root spectrum has a large similarity with a cosine is the Hann window. A Hann window of size T (in seconds) is defined as

$$h_T(t) = \frac{1}{T}\left[1 + \cos\left(\frac{2\pi t}{T}\right)\right] \tag{9}$$

for $|t| < 1/2$, and 0 otherwise. Its FT is

$$H_T(f) = \mathrm{sinc}(Tf) + \tfrac{1}{2}\mathrm{sinc}(Tf - 1) + \tfrac{1}{2}\mathrm{sinc}(Tf + 1), \tag{10}$$

where the sinc function is defined as

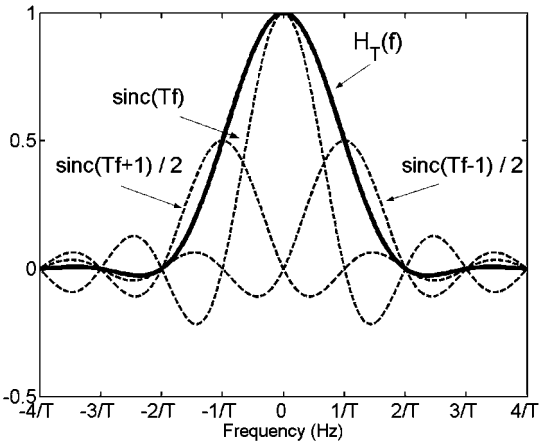

FIG. 5. FT of the Hann window. The FT of the Hann window is a sum of three sinc functions.

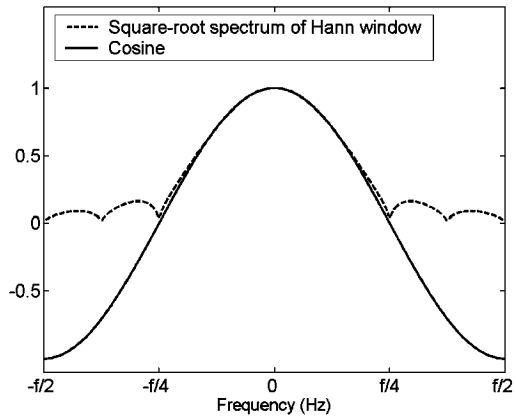

FIG. 6. Similarity between cosine lobe and square root of the spectrum of the Hann window.

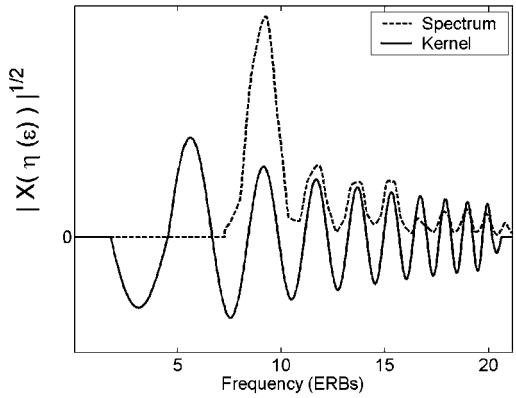

FIG. 7. Normalized SWIPE kernel. The SWIPE kernel consists of a truncated decaying cosine with halved first and last valleys. The kernel is normalized using its positive part.

$$\mathrm{sinc}(\varphi) = \frac{\sin(\pi\varphi)}{\pi\varphi}. \tag{11}$$

This FT is illustrated in Fig. 5. The width of its main lobe is $4/T$. If we match this width to the width of the cosine lobe, $f/2$, where f is the pitch of the candidate (in hertz), and solve for T, we will find that the window size must be $T=8/f$.

Figure 6 shows the square root of the spectrum of a Hann window of size $T=8/f$ and a cosine with period f. The similarity between the main lobe of the spectrum and the positive lobe of the cosine is remarkable. They match at five frequencies: 0, $\pm f/8$, and $\pm f/4$, with values $\cos(0)=1$, $\cos(\pi/4)=1/\sqrt{2}$, and $\cos(\pi/2)=0$, respectively. The normalized inner product between the main lobe of the spectrum and the positive part of the cosine sampled at 128 equidistant points is 0.9996, and the normalized inner product computed over the whole period of the cosine sampled at 256 equidistant points is 0.8896. This reduction in normalized inner product is caused by the side lobes.

The normalized inner product between the cosine lobe and the main lobe of the square root of the spectrum for other types of windows is shown in Table I. The width of the main lobes of the spectrum of these windows is $2k/T$, where k depends on the window type (see Oppenheim *et al.*, 1999), and is given in the second column of the table.

The window type that produces the largest normalized inner product between the square root of the main lobe of the spectrum and the cosine lobe is the Hann window (0.9996). When computed over the whole period of the cosine, there are other window types that produce a larger normalized inner product. However, most of these windows are larger than the Hann window, and therefore require more computation. The only window that has the same value of k as the Hann window but a larger normalized inner product over the whole period of the cosine is the Hamming window. However, we prefer the Hann window because of its simpler formula.

H. SWIPE

Putting all pieces together, we can express the SWIPE estimate of the pitch at time t as

$$p(t) = \arg\max_{f} \frac{\int_0^{\mathrm{ERBs}(f_{\max})} 1/\eta(\varepsilon)^{1/2} K(f,\eta(\varepsilon)) |X(t,f,\eta(\varepsilon))|^{1/2} d\varepsilon}{\left(\int_0^{\mathrm{ERBs}(f_{\max})} 1/\eta(\varepsilon)[K^{+}(f,\eta(\varepsilon))]^2 d\varepsilon\right)^{1/2} \left(\int_0^{\mathrm{ERBs}(f_{\max})} |X(t,f,\eta(\varepsilon))| d\varepsilon\right)^{1/2}}, \tag{12}$$

where

$$K(f,f') = \begin{cases} \cos(2\pi f'/f) & \text{if } 3/4 < f'/f < n(f) + 1/4 \\ \frac{1}{2}\cos(2\pi f'/f) & \text{if } 1/4 < f'/f < 3/4 \text{ or } n(f) + 1/4 < f'/f < n(f) + 3/4 \\ 0 \quad \text{otherwise,} & \end{cases} \tag{13}$$

$$X(t,f,f') = \int_{t-4/f}^{t+4/f} [1 + \cos(\pi f(t'-t)/4)] x(t') e^{-j2\pi f' t'} dt', \tag{14}$$

ε is frequency in ERBs, $\eta(\cdot)$ converts frequency from ERBs into hertz, ERBs$(\cdot)$ converts frequency from hertz into ERBs, $K^{+}(\cdot)$ is the positive part of $K(\cdot)$ {i.e., $\max[0,K(\cdot)]$}, $f_{\max}$ is the maximum frequency to be used (typically the Nyquist frequency, although 5 kHz is enough for most applications), $n(f)=\lfloor f_{\max}/f-3/4 \rfloor$, and $j^2=-1$.

The normalized kernel corresponding to a candidate with frequency 190 Hz (5.6 ERBs) is shown in Fig. 7. The

TABLE I. Normalized inner products between the kernel and the square root of the spectrum of several window types,[a] computed over the main spectral lobe and over one period of the cosine around zero. The parameter k determines the window size T required to produce those normalized inner products, based on the formula $T=4k/f$, where f is the frequency of the cosine.

		Normalized inner product	
Window type	k	Main spectral lobe	Whole cosine period
Bartlett	2	0.9984	0.7959
Bartlett–Hann	2	0.9995	0.8820
Blackman	3	0.9899	0.9570
Blackman–Harris	4	0.9738	0.9689
Bohman	3	0.9926	0.9474
Flattop	5	0.9896	0.9726
Gauss	3.14	0.9633	0.8744
Hamming	2	0.9993	0.9265
Hann	2	0.9996	0.8896
Nuttall	4	0.9718	0.9682
Parzen	4	0.9627	0.9257
Rectangular	1	0.9925	0.5236
Triangular	2	0.9980	0.8820

[a]The normalized inner products were computed using 128 equidistant samples for the main spectral lobe and 256 equidistant samples for the whole cosine period.

figure also shows the normalized spectrum of a signal with the same pitch. It is easy to show that the type of signal for which the function maximized in Eq. (12) achieves its maximum is periodic signals whose spectral envelope decays as $1/f$. An example of such type of signal is a sawtooth waveform, which is the one that inspired the name of the algorithm. Another type of signal with that property (on average) is a vowel (Fant, 1970).

I. SWIPE′

One of the most common mistakes of SWIPE (and other algorithms) is misestimating the pitch as one of its subharmonics. Figure 8 illustrates why this error is common. It shows the spectrum of a signal whose component frequencies are harmonics of 100 Hz and are of equal amplitude. It

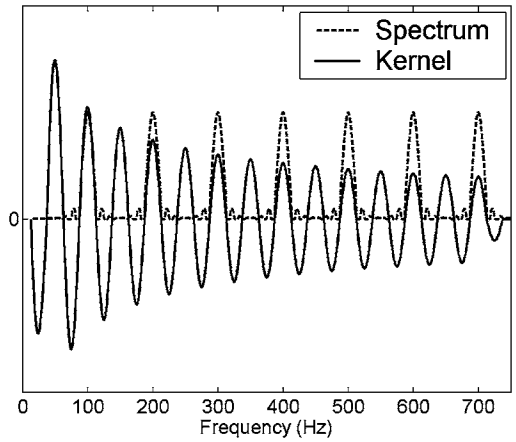

FIG. 8. Kernel corresponding to a subharmonic of the pitch. The figure shows a signal formed by equal-amplitude harmonics of 100 Hz, and the kernel corresponding to the 50 Hz candidate. This kernel produces a high score based only on its even components.

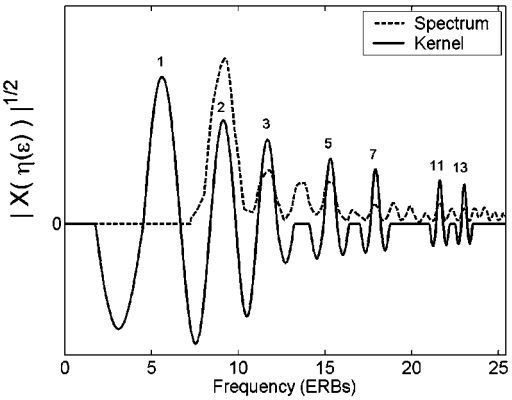

FIG. 9. Normalized SWIPE′ kernel. The SWIPE′ kernel has peaks only on its first and prime harmonics.

also shows the kernel corresponding to a candidate of 50 Hz. Since every multiple of 100 is also a multiple of 50, this candidate will receive significant credit based only on its even harmonics. Something similar would occur with the kernel at 33 Hz (not shown), but based on its multiples-of-3 harmonics. The phenomenon extends to any subharmonic of the pitch.

To avoid such situations, we propose to remove from the kernel its nonprime harmonics, except the first one. We do this by redefining the kernel as

$$K(f,f') = \sum_{i \in \{1\} \cup P} K_i(f,f'), \tag{15}$$

where P is the set of prime numbers, and

$$K_i(f,f') = \begin{cases} \cos(2\pi f'/f) & \text{if } |f'/f - i| < 1/4 \\ \frac{1}{2}\cos(2\pi f'/f) & \text{if } 1/4 < |f'/f - i| < 3/4 \\ 0 \quad \text{otherwise.} \end{cases} \tag{16}$$

[Notice that the SWIPE kernel can also be written as in Eqs. (15) and (16) by including all the harmonics (not only the first and the primes) in the sum of Eq. (15).]

This variation of SWIPE in which only the first and prime harmonics are used is named SWIPE′. The kernel corresponding to a pitch candidate of 190 Hz (5.6 ERBs) is shown in Fig. 9. The numbers on top of the peaks of the kernel indicate the corresponding harmonic number. With this approach, no candidate below the pitch will get credit from more than one of the harmonics of the signal.

J. Reduction of computational cost

Equations (12)–(14) [or Eqs. (12) and (14)–(16)] form a complete description of the algorithm to estimate pitch in continuous time. However, in practice, we use digital computers, which require discrete-time and discrete-frequency versions of these equations. Also, computational power is usually limited, which implies that arbitrarily large levels of resolution are unattainable. However, most signals we deal with are relatively slowly varying, which means that after a certain point, increments in resolution are unnecessary because they only provide redundant information. For many

applications, computations can be done at a relatively low resolution, and if higher resolution is required, interpolation can be used to fill in the gaps. This idea will be exploited in the following sections.

1. Reduction of the number of Fourier transforms

One of the most costly operations of SWIPE and SWIPE′ is the computation of the (short term) FT in Eq. (14). Hence, to reduce computational cost, it is important to reduce the number of FTs. There are two strategies to achieve this: reducing the window overlap and sharing FTs among candidates.

a. Reduction of window overlap. The most common windows used in signal processing are attenuated towards zero at the edges. Therefore, it is possible to overlook short events if they are located at these edges. To avoid this situation, it is common to use overlapping windows, which increase the coverage of the signal at the cost of an increase in computation. However, after a certain point, overlapping windows produce redundant data, making it unnecessary to go to the limit of sample by sample window shifting.

A study by Doughty and Garner (1947) showed that, depending on frequency, at least two to four cycles are required to perceive the pitch of a pure tone. To avoid this interaction between number of cycles and frequency, we will assume that a minimum of four cycles are required to detect the pitch of a pure tone, regardless of its frequency. Furthermore, we extrapolate this result to the type of signals for which SWIPE produces maximum output: sawtooth waveforms. Finally, we assume that the pitched/unpitched decision threshold is no larger than half the score obtained by a full length sawtooth waveform (~0.89, according to Table I). It can be shown that, under these assumptions, four cycles of a sawtooth waveform will obtain a score higher than the threshold as long as the window overlap is at least 50%.

The worst case scenario occurs when the four cycles of the sawtooth are centered at the middle point between two consecutive windows. In this scenario, the signal spans over one-half of each of the windows and the score obtained on each window is half the score that would be obtained if the signal would cover the whole window. Shifting the signal to the left/right necessarily increases the score on the left/right window, making the detection of the pitch more likely.

b. Use of exclusively power-of-2 window sizes. The use of the "optimal" window size proposed in Sec. II G requires that each pitch candidate uses a different window size, which means that a FT must be computed for each candidate. Furthermore, since the most efficient way to compute a FT is using a fast Fourier transform (FFT) and this algorithm is more efficient when using window sizes that are powers of 2 (in samples), the optimal window sizes may not be appropriate to use a FFT.

To alleviate this problem, we propose to substitute the optimal window size with the two closest power-of-2 window sizes, and then linearly combine the scores obtained using these windows into a single score, based on the distance between the size of the windows and the optimal window size, in a logarithmic scale. More precisely, if the optimal window size in samples is $N^*=2^{L+\lambda}$, where L is an integer and $0\leq\lambda<1$, the two closest power-of-2 window sizes are $N_0=2^L$ and $N_1=2^{L+1}$. Using these sizes, we compute scores $S_0(f)$ and $S_1(f)$ as in the function maximized in Eq. (12), modifying the integration region in Eq. (14) to have a length corresponding to those sizes. Finally, we combine these two scores into the single score

$$S(f)=(1-\lambda)S_0(f)+\lambda S_1(f). \tag{17}$$

The intuition behind this formula is that if the optimal window size is closer to N_0 than to N_1, $S_0(f)$ will have a larger contribution towards $S(f)$ than $S_1(f)$, and vice versa.

Alternatively, we could have used only the score produced by the closest power-of-2 window size (in a logarithmic scale), but this would have introduced undesirable discontinuities in $S(f)$, as shown in Fig. 10(a). This figure shows scores produced by two different window sizes. The pitch was chosen to match the point where the change of

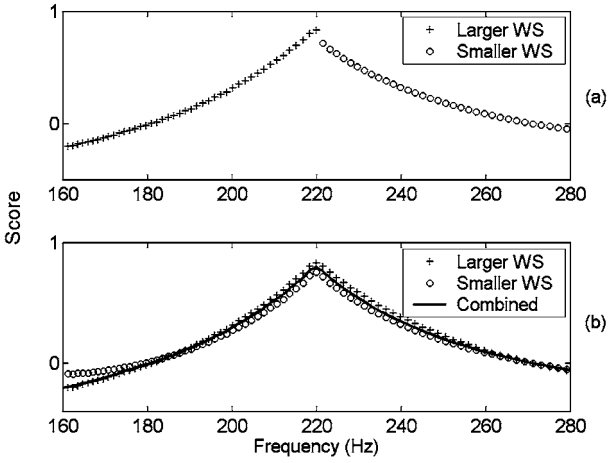

FIG. 10. Individual and combined scores of candidates between 160 and 280 Hz for a sawtooth waveform with a pitch of 220 Hz. The scores in (a) were produced using the power-of-2 window size closest to the optimal window size for each of the candidates. The scores in (b) (crosses and circles) were produced using the *two* power-of-2 window sizes closest to the optimal window size for each of the candidates. The continuous line in (b) consists of the combination of the scores produced by the two power-of-2 window sizes closest to the optimal window size for each of the candidates, as determined in Eq. (17).

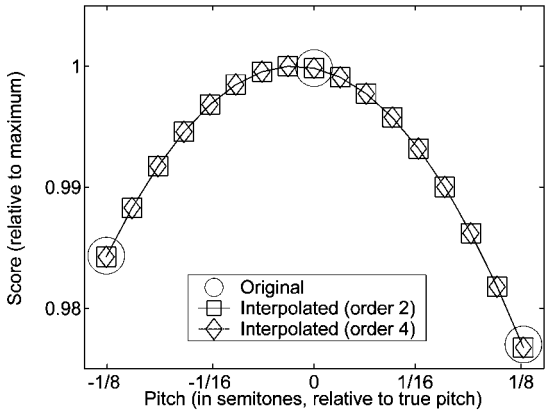

FIG. 11. Interpolated scores. Circles show scores of candidates separated at a distance of 1/8 semitone, squares show interpolated scores produced by an order-2 polynomial, and diamonds show interpolated scores produced by an order-4 polynomial.

window size occurs. Since the larger window tends to produce larger scores, the estimated pitch may be shifted slightly to the left in this case. In Fig. 10(b), the scores produced by both windows were combined using Eq. (17) to produce the continuous solid curve in between.

Besides using a convenient window size for the FFT computation, the approximation of the optimal window size using the two closest power-of-2 window sizes has another advantage that is probably more important: The same FFT can be shared by several pitch candidates, more precisely, by all the candidates within an octave of the optimal pitch for that FFT.

Using this approach, and translating the algorithm to a discrete-time domain (necessary to compute a FFT), we can express the SWIPE estimate of the pitch at the discrete-time index τ as

$$p[\tau] = \arg\max_f (1 - \lambda(f)) S_{L(f)}(\tau, f) + \lambda(f) S_{L(f)+1}(\tau, f), \tag{18}$$

where

$$\lambda(f) = L^*(f) - L(f), \tag{19}$$

$$L(f) = \lfloor L^*(f) \rfloor, \tag{20}$$

$$L^*(f) = \log_2(8 f_s / f), \tag{21}$$

$$S_L(\tau, f) = \frac{\sum_{m=0}^{\lfloor \mathrm{ERBs}(f_{\max})/\Delta\varepsilon \rfloor} 1/\eta(m\Delta\varepsilon)^{1/2} K(f, \eta(m\Delta\varepsilon)) |\hat{X}_{2^L}[\tau, \eta(m\Delta\varepsilon)]|^{1/2}}{\left(\sum_{m=0}^{\lfloor \mathrm{ERBs}(f_{\max})/\Delta\varepsilon \rfloor} 1/\eta(m\Delta\varepsilon) [K^+(f, \eta(m\Delta\varepsilon))]^2\right)^{1/2} \left(\sum_{m=0}^{\lfloor \mathrm{ERBs}(f_{\max})/\Delta\varepsilon \rfloor} |\hat{X}_{2^L}[\tau, \eta(m\Delta\varepsilon)]|\right)^{1/2}}, \tag{22}$$

$$\hat{X}_N[\tau, f'] = I(\{0, \ldots, N-1\}, X_N[\tau, \{0, \ldots, N-1\}], f' N / f_s), \tag{23}$$

$$X_N[\tau, \varphi] = \sum_{\tau'=-N/2}^{N/2-1} [1 + \cos(2\pi(\tau' - \tau)/N)] x[\tau'] e^{-j 2\pi \varphi \tau' / N}, \tag{24}$$

f_s is the sampling frequency, $\Delta\varepsilon$ is the ERB scale step size (0.1 was used in our tests), and $I(\Phi, \Xi, \phi)$ is an interpolating function that uses the functional relations $\Xi_k = F(\Phi_k)$ to predict the value of $F(\phi)$. The other variables, constants, and functions are defined as before (see Sec. II H).

2. Reducing the number of spectral integral transforms

Since the pitch resolution of SWIPE depends on the granularity of the pitch candidates, to achieve high pitch resolution, a large number of pitch candidates are required. To avoid excessive computational cost, we propose to compute the score only for certain candidates, and then use interpolation to estimate the score of the others. Since the autocorrelation function of a signal is the FT of its power spectrum, it consists of a sum of cosines that can be approximated around zero by using a Taylor series expansion with even powers (de Cheveigné, 2002). If the signal is periodic, its autocorrelation function is also periodic, and the shape of the curve around the pitch period is the same as the shape around zero. This means that the autocorrelation function around the pitch period can be approximated by the Taylor series expansion around zero after shifting it to the pitch period. If the width of the spectral lobes is narrow and the energy of the high frequency components is small, the terms of order 4 and higher in the series vanish as the independent variable approaches the pitch period, and the series can be approximated using a parabola.

Since SWIPE multiplies a compressed version of the spectrum by a cosine-based kernel, it could be expected that a similar argument applies to SWIPE as well. However, there are two complications: First, the widths of the spectral lobes produced by SWIPE are not narrow, in fact, they are as wide as the positive lobes of the cosine; and second, the use of the square root of the spectrum rather than its energy could make the contribution of the high frequency components large, violating the principle of low contribution of high frequency components.

Nevertheless, parabolic interpolation produces a good fit to SWIPE's score in the neighborhood of the maxima, as shown in Fig. 11. This figure shows in circles the scores (relative to the maximum) corresponding to the pitch and two pitch candidates located at a distance of 1/8 semitones from the pitch, for a signal consisting of a sawtooth waveform. Since the function looks almost symmetric around its maximum, its Taylor series expansion must contain mostly even terms. The figure also shows polynomial expansions of order 2 (squares) and order 4 (diamonds), which look identical. Expansions of higher order are not shown, but they look the same as these two. Therefore, a parabola is good enough to satisfactorily interpolate the scores in the neighborhood of the maximum for the chosen resolution.

III. EVALUATION

SWIPE and SWIPE′ were compared against other algorithms in terms of performance using three speech databases and a musical instruments database.

A. Algorithms

The algorithms against which SWIPE and SWIPE′ were compared are the following:

AC-P. This algorithm (Boersma, 1993) computes the autocorrelation of the signal and divides it by the autocorrelation of the window used to analyze the signal. It uses postprocessing to reduce discontinuities in the pitch trace. It is available with the Praat System at http://www.fon.hum.uva.nl/praat. The name of the function is *ac*.

AC-S. This algorithm uses the autocorrelation of the cubed signal. It is available with the Speech Filing System at http://www.phon.ucl.ac.uk/resource/sfs. The name of the function is *fxac*.

ANAL. This algorithm (Secrest and Doddington, 1983) uses autocorrelation to estimate the pitch, and dynamic programming to remove discontinuities in the pitch trace. It is available with the Speech Filing System at http://www.phon.ucl.ac.uk/resource/sfs. The name of the function is *fxanal*.

CATE. This algorithm uses a quasiautocorrelation function of the speech excitation signal to estimate the pitch. We implemented it based on its original description (Di Martino and Laprie, 1999). The dynamic programming component used to remove discontinuities in the pitch trace was not implemented.

CC. This algorithm uses cross correlation to estimate the pitch and postprocessing to remove discontinuities in the pitch trace. It is available with the Praat System at http://www.fon.hum.uva.nl/praat. The name of the function is *cc*.

CEP. This algorithm (Noll, 1967) uses the *cepstrum* of the signal and is available with the Speech Filing System at http://www.phon.ucl.ac.uk/resource/sfs. The name of the function is *fxcep*.

ESRPD. This algorithm (Bagshaw, 1993; Medan *et al.*, 1991) uses a normalized cross correlation to estimate the pitch, and postprocessing to remove discontinuities in the pitch trace. It is available with the Festival Speech Filing System at http://www.cstr.ed.ac.uk/projects/festival. The name of the function is *pda*.

RAPT. This algorithm (Secrest and Doddington, 1983) uses a normalized cross correlation to estimate the pitch, and dynamic programming to remove discontinuities in the pitch trace. It is available with the Speech Filing System at http://www.phon.ucl.ac.uk/resource/sfs. The name of the function is *fxrapt*.

SHS. This algorithm (Hermes, 1988) uses subharmonic summation. It is available with the Praat System at http://www.fon.hum.uva.nl/praat. The name of the function is *shs*.

SHR. This algorithm (Sun, 2000) uses the subharmonic-to-harmonic ratio. It is available at Matlab Central (http://www.mathworks.com/matlabcentral) under the title "Pitch Determination Algorithm." The name of the function is *shrp*.

TEMPO. This algorithm (Kawahara *et al.*, 1999) uses the instantaneous frequency of the outputs of a filterbank. It is available with the STRAIGHT System at its author web page (http://www.wakayama-u.ac.jp/~kawahara). The name of the function is *exstraightsource*.

YIN. This algorithm (de Cheveigné and Kawahara, 2002) uses a modified version of the average squared difference function. It was made available by his author. The name of the function is *yin*.

B. Databases

The databases used to test the algorithms were the following.

DVD: disordered voice database. This database contains 657 samples of the sustained vowel "ah" produced by persons with disordered voice. Most of the files have a sampling rate of 25 kHz, although a few of them (77) have a sampling rate of 50 kHz. It can be bought from Kay Pentax (http://www.kayelemetrics.com).

KPD: Keele pitch database. This speech database was collected by Plante *et al.* (1995) at Keele University with the purpose of evaluating pitch estimation algorithms. It contains about 8 min of speech spoken by five males and five females, sampled at 20 kHz. Laryngograph data were recorded simultaneously with speech and were used to produce estimates of the fundamental frequency.

MIS: musical instruments samples. This database contains more than 150 min of sound produced by 20 different musical instruments sampled at 44.1 kHz.[4] It was collected at the University of Iowa Electronic Music Studios, directed by Lawrence Fritts, and is publicly available at http://theremin.music.uiowa.edu.

PBD: Paul Bagshaw's database for evaluating pitch determination algorithms. This database contains about 8 min of speech spoken by one male and one female, sampled at 20 kHz. Laryngograph data were recorded simultaneously with speech and were used to produce estimates of the fundamental frequency. It was collected by Paul Bagshaw at the University of Edinburg (Bagshaw *et al.*, 1993; Bagshaw, 1994) and is publicly available at http://www.cstr.ed.ac.uk/research/projects/fda.

C. Methodology

Whenever possible, the algorithms were asked to produce pitch estimates at every millisecond. The search range was set to 40–800 Hz for speech and 30–1666 Hz for musical instruments, and the algorithms were given the freedom to decide if the sound was pitched or not. To compute the statistics, we rounded to the closest millisecond the times associated with each ground truth value and the times associated with each pitch estimate produced by the algorithms that were able to give an output every millisecond, and considered only the rounded times at which all these algorithms and the ground truth agreed that the sound was pitched.

Special care was taken to account for misalignments. Specifically, the pitch estimates were associated with the time corresponding to the center of their respective analysis windows, and when the ground truth pitch varied over time (i.e., for PBD and KPD), the estimated pitch time series were shifted in steps of 1 ms within the range ± 100 ms to find the best alignment with the ground truth.

The performance measure used to compare the algorithms was the gross error rate (GER). A gross error occurs when the estimated pitch is off from the reference pitch by

TABLE II. GERs obtained for speech and voice databases (PBD, KPD, and DVD).[a]

Algorithm	Gross error (%)			
	PBD	KPD	DVD	Average
SWIPE′	0.13	0.83	0.63	0.53
SHS	0.15	1.00	1.10	0.75
SWIPE	0.15	0.87	1.70	0.91
RAPT	0.75	1.00	2.40	1.40
TEMPO	0.32	1.90	2.00	1.40
YIN	0.33	1.40	4.50	2.10
SHR	0.69	1.50	5.10	3.50
ESRPD	1.40	3.90	4.60	5.00
CEP	6.10	4.20	14.00	5.90
AC-P	0.73	2.90	16.00	6.70
CATE	2.60	10.00	7.20	6.60
CC	0.48	3.60	5.00	2.40
ANAL	0.83	2.00	35.00	13.00
AC-S	8.80	7.00	40.00	19.00
Average	1.70	3.00	9.90	4.90

[a]Values computed using two significant digits.

TABLE III. GERs by gender for speech databases (PVD and KPD).[a]

Algorithm	Gross error (%)		
	Male	Female	Average
SWIPE′	0.36	2.40	1.4
SHS	0.55	2.50	1.5
SWIPE	0.49	2.70	1.6
RAPT	0.42	2.90	1.7
TEMPO	0.67	3.10	1.9
SHR	0.61	3.60	2.1
YIN	1.10	3.20	2.2
AC-P	2.10	3.60	2.9
CEP	1.80	4.20	3.0
CC	2.40	4.50	3.5
ESRPD	3.10	3.90	3.5
ANAL	1.30	5.90	3.6
AC-S	3.20	10.00	6.6
CATE	11.00	4.20	7.6
Average	2.10	4.00	3.1

[a]Values computed using two significant digits.

more than 20%. At first glance, this margin of error may seem too large, but considering that most of the error pitch estimation algorithms produced are octave errors (i.e., halving or doubling the pitch), this is a reasonable metric. On the other hand, this tolerance gives room for dealing with misalignments. The GER measure has been used previously to test pitch estimators by other researchers (e.g., Bagshaw *et al.*, 1993; Di Martino and Laprie, 1999; de Cheveigné and Kawahara, 2002).

D. Results

Table II shows the GERs for each of the algorithms over each of the speech databases. The rows and the columns are sorted by average GER (the best algorithms are at the top and the more difficult databases are at the right). The best algorithm overall is SWIPE′, followed by SHS and SWIPE. On average, SHS performed better than SWIPE; however, SHS beat SWIPE only on the disordered voice database, but not in the normal speech databases, which suggests that SWIPE works better than SHS on normal speech.

Table III shows the pitch estimation performance as a function of gender for the two databases for which we had access to this information: PVD and KPD. On average, error rates are larger for females than for males.

Table IV shows the GERs for the musical instruments database. Some of the algorithms were not evaluated on this database because they did not provide a mechanism to set the search range, and the range they covered was smaller that the pitch range spanned by the database. The two algorithms that performed the best were SWIPE′ and SWIPE.

Table V shows the GERs by instrument family. The two best algorithms are SWIPE′ and SWIPE. SWIPE′ tends to perform better than SWIPE, except for the piano, for which SWIPE produces almost no error. SWIPE′ performance on piano is relatively bad, especially when compared against correlation based algorithms. In general, the family for which fewer errors were obtained was the brass family (many algorithms achieved almost perfect performance for this family). The family for which more errors were produced was the strings family playing *pizzicato*, i.e., by plucking the strings. Indeed, pizzicato sounds were the ones for which the performers produced more errors and the ones that were hardest for us to label (see Appendix).

Table VI shows the GERs as a function of octave. The best performance on average was achieved by SWIPE′ and SWIPE.

As a final test, we wanted to validate the choices we made in Sec. II, i.e., shape of the kernel, warping of the spectrum, weighting of the harmonics, warping of the frequency scale, and selection of window type and size. For this purpose, we evaluated SWIPE′ replacing every one of its features with most[5] of the alternative features described in Secs. II B–II G. We varied each of these features, one at a time, and obtained the results shown in Table VII. The step sizes used for each of the alternative frequency scales and the actual number of steps is shown in Table VIII. The step sizes

TABLE IV. GERs for musical instruments (MIS database).[a]

Algorithm	Gross error (%)		
	Underestimates	Overestimates	Total
SWIPE′	1.00	0.10	1.10
SWIPE	1.30	0.02	1.30
SHS	0.88	1.00	1.90
TEMPO	0.29	1.70	2.00
YIN	1.60	0.83	2.40
AC-P	3.20	0.00	3.20
CC	3.60	0.00	3.60
ESRPD	5.30	1.50	6.80
SHR	15.00	5.30	20.00
Average	3.60	1.20	4.70

[a]Values computed using two significant digits.

TABLE V. GERs by instrument family (MIS database).[a]

	Gross error (%)					
Algorithm	Brass[b]	Bowed strings[c]	Woodwinds[d]	Piano	Plucked strings[e]	Average
SWIPE′	0.01	0.19	0.14	2.20	8.80	2.30
SWIPE	0.00	0.22	0.23	0.02	11.00	2.30
TEMPO	0.00	2.60	1.40	7.30	4.00	3.10
YIN	0.03	1.10	1.50	0.36	14.00	3.40
SHS	0.02	1.50	0.72	12.00	8.10	4.50
AC-P	0.03	0.56	0.80	0.36	26.00	5.60
CC	0.07	0.83	1.00	0.36	28.00	6.00
ESRPD	4.00	6.90	7.10	6.00	11.00	7.00
SHR	22.00	25.00	38.00	26.00	15.00	25.00
Average	2.90	4.30	5.60	6.10	14.00	6.60

[a]Values computed using two significant digits.
[b]French horn, bass/tenor trombones, trumpet, and tuba.
[c]Double bass, cello, viola, and violin.
[d]Flute, bass/alto flutes, bass/Bb/Eb clarinets, and alto/soprano saxophones.
[e]Double bass and violin.

were empirically chosen in a pilot test by decreasing their magnitude until no significant improvement in performance was observed. In all cases, the number of steps was forced to be at least as large as the number of steps used on the ERB scale.

Overall, no alternative feature made SWIPE′ improve neither consistently nor on average over all the databases, although some of the alternative features performed as well or almost as well as the proposed features, specifically, the parabolic and Gaussian kernels, and all window types. However, there were some features that made SWIPE′ improve significantly (by more than 10% relative to the GER) on the musical instruments database, specifically, the flat kernel envelope, and the Hertz, Barks, and mel scales. Inspection of the spectrum of the signals on which SWIPE′ improved showed that the spectrum was far from having the expected $1/f$ envelope. Instead, the spectrum tended to increase with harmonic number at low order harmonics, and then decreased after a relatively high order harmonic. This is consistent with the improvement produced by the flat envelope, which gives relatively more weight to high order harmonics than the $1/f$ envelope does. It is also consistent with the improvement produced by the Hertz and mel scales, which give a relatively better sampling of the high frequencies compared to the ERB scale [i.e., the transition from linear to logarithmic behavior in Eq. (5) occurs at larger values of σ]. Such statement is hard to make for the Bark scale since Eqs. (5) and (6) are not directly comparable. However, the GERs of Table VII suggest that the Bark scale behaves as being between the mel and the Hertz scales, probably because its frequency scaling factor (1960) is between the frequency scaling factors of the mel and Hertz scales (700 and infinity, respectively). Finally, the last row of Table VII shows the results of combining the flat envelope and the Hertz scale.[6] The combination of these features made SWIPE′ improve even more on the musical instruments database, but also worsen on the speech databases.

The discussion in the previous paragraph suggests that

TABLE VI. GERs for musical instruments by octave (MIS database).[a]

	Gross error (%)						
Algorithm	46.2 Hz +/−1/2 oct.	92.5 Hz +/−1/2 oct.	185 Hz +/−1/2 oct.	370 Hz +/−1/2 oct.	740 Hz +/−1/2 oct.	1480 Hz +/−1/2 oct.	Average
SWIPE′	1.20	1.00	2.30	0.89	0.13	0.29	0.97
SWIPE	0.08	1.20	3.00	1.00	0.25	0.38	0.99
YIN	3.20	0.95	5.30	1.80	0.69	0.96	2.20
AC-P	0.24	2.00	7.80	2.50	0.71	0.30	2.30
SHS	7.80	2.60	3.20	1.20	0.23	0.14	2.50
CC	0.26	2.60	8.20	2.70	0.93	0.40	2.50
TEMPO	15.00	2.80	2.00	1.10	0.52	0.31	3.60
ESRPD	7.90	2.60	4.80	4.20	12.00	32.00	11.00
SHR	37.00	0.60	1.80	27.00	70.00	81.00	36.00
Average	8.10	1.80	4.30	4.70	9.50	13.00	6.90

[a]Values computed using two significant digits.

TABLE VII. GERs of variations of SWIPE′ on PBD, KPD, DVD, and MIS databases.[a]

Variation	Gross error (%)				
	PBD	KPD	DVD	MIS	Average
Original	0.13	0.83	0.63	1.10	0.67
Flat envelope	0.16	1.00	1.40	0.60	0.79
Raw spectrum and $1/f$ envelope[b]	0.30	2.20	1.70	10.00	3.60
Squared spectrum and $1/f^2$ envelope[b]	12.00	14.00	9.70	22.00	14.00
Pulsed kernel	0.21	0.84	3.00	2.60	1.70
Parabolic kernel	0.13	0.83	0.62	1.10	0.67
Gaussian kernel	0.13	0.83	0.63	1.10	0.67
Hertz scale[c]	0.23	1.70	1.40	0.37	0.93
Bark scale[c]	0.18	1.00	0.90	0.61	0.67
Mel scale[c]	0.16	0.97	0.88	0.67	0.67
Logarithmic scale[c]	0.14	1.10	0.89	2.30	1.10
Fixed window size[d]	0.15	0.77	1.70	9.10	2.90
Hamming window ($k=2$)[e]	0.13	0.80	0.64	1.20	0.69
Blackman window ($k=3$)[e]	0.14	0.80	0.74	1.20	0.72
Gaussian window ($k=3.14$)[e]	0.13	0.83	0.63	1.10	0.67
Blackman-Harris window ($k=4$)[e]	0.15	0.76	0.77	1.20	0.72
Flat envelope and Hertz scale[f]	0.49	5.00	2.30	0.17	2.00

[a]Values computed using two significant digits.
[b]The use of the raw or squared spectrum implies the use of a kernel whose envelope decays as $1/f$ or $1/f^2$, respectively, to match the spectral envelope of a sawtooth waveform.
[c]Spectrum was computed using FFTs and was inter/extrapolated to equidistant steps in the specified scale (see Table VIII for step size).
[d]The power-of-2 window size whose optimal pitch was closest to the geometric mean pitch of the database was used in each case (1024 for the speech databases and 256 for the musical instruments database).
[e]Window type was selected among all the window types with that value of k for having the spectrum whose square root produced the largest normalized inner product between its main lobe and one period of the cosine around zero (see Table I).
[f]The spectral analysis was limited to the range 0–5 kHz (see text for details).

SWIPE and SWIPE′ would benefit from preprocessing the signal with an auditory model, as in the work by Klapuri (2008). Klapuri (2008) showed that preprocessing the signal with a gammatonelike filterbank, followed by a half-wave rectifier, a compressive function, and a low pass filter, tend to produce low order harmonics in each of the outputs of the system, even in the outputs that came from filters that responded only to high order harmonics. This means that such preprocessing would boost (or make appear) weak (or missing) low order harmonics, making the signal more suitable for analysis with SWIPE or SWIPE′. However, this claim needs further validation.

IV. SUMMARY AND CONCLUSIONS

SWIPE estimates the pitch as the fundamental frequency of the sawtooth waveform whose spectrum best matches the spectrum of the input signal. The schematic description of the algorithm is the following.

(1) For each pitch candidate f within a pitch range $f_{\min}-f_{\max}$, compute its score as follows:
   (a) Compute the square root of the spectrum of the signal.
   (b) Normalize the square root of the spectrum and apply an integral transform using a normalized cosine kernel whose envelope decays as $1/\sqrt{f}$.

TABLE VIII. Step sizes and number of steps used to sample the spectrum for each of the frequency scales and databases. The range used to sample the spectrum was lower bounded by one-quarter of the lowest expected pitch and upper bounded by the frequency specified in parenthesis (the Nyquist frequency on each case). (DVD has two columns because it contains files sampled at two different sampling rates.)

Scale	Step size	Number of steps			
		MIS (5 kHz)	PBD & KPD (10 kHz)	DVD (12.5 kHz)	DVD (25 kHz)
ERBs	1/10	288	350	370	434
Bark	1/20	384	446	461	495
Mel	5	471	612	659	809
Logarithmic	1/96 octave	901	957	988	1084
Hertz	5	1000	2000	2500	5000

(2) Estimate the pitch as the highest score candidate.

An implicit objective of the algorithm is to find the frequency for which the average peak-to-valley distance at its harmonics is maximized. To achieve this, the kernel is set to zero below the first negative lobe and above the last negative lobe, and to avoid bias, the magnitude of these two lobes is halved. To make the contribution of each harmonic of the sawtooth waveform proportional to its amplitude, the square root of the spectrum is taken before applying the integral transform. To make the kernel match the normalized square-root spectrum of the sawtooth waveform, a $1/\sqrt{f}$ envelope is applied to the kernel, which is normalized using only its positive part, and the spectrum is computed using a window size that makes the width of the main spectral lobes match the width of the lobes of the kernel.

Several techniques are applied to reduce computational cost. First, the optimal window size is replaced with the two closest power-of-2 window sizes, for which it is more efficient to compute a FFT, and the scores are appropriately combined to produce a single score. This approach has the extra advantage of allowing a FFT to be shared by several pitch candidates. Second, the scores are computed using a coarse resolution and then fine-tuned using parabolic interpolation. Third, the window overlap is minimized while allowing the pitch of a signal as short as four cycles to be recognized. Last, the inner product between the kernel and the square root of the spectrum is computed on the ERB frequency scale, since this scale emphasizes the regions where most of the spectral energy is concentrated. This last technique not only reduces the computational cost but also improves the performance (as shown in the tests).

SWIPE′, a variation of SWIPE, uses only the first and prime harmonics of the signal, which largely decreases the scores of subharmonics of the pitch, significantly reducing the chances of estimating the pitch as one of its subharmonics.

SWIPE and SWIPE′ were tested using speech and musical instruments databases and their performance was compared against 12 other algorithms, which have been cited in literature and for which free implementations exist. SWIPE′ was shown to outperform all the algorithms on all the databases. SWIPE was ranked second in the normal speech and musical instruments databases, and was ranked third in the disordered speech database.

APPENDIX: EXTRA DETAILS OF THE EVALUATION

1. Disordered voice database

The fundamental frequency estimates included in the disordered voice database were not used as ground truth. The reason was that we wanted to estimate pitch, and fundamental frequency does not always correspond to pitch, as mentioned in the Introduction.

Since this database consisted of sustained vowels and most of them had a relatively stable pitch, we used a subject with vast experience in music transcription (the first author) to label the samples with their pitch, by matching them to the closest note playing sawtooth waveforms on an electronic keyboard. Assuming that he correctly chose one of the two closest notes, this procedure should introduce an error no larger than 6% (one semitone), which is smaller than the 20% necessary to produce a gross error (see its definition in Sec. III C).

There are some samples in this database for which the pitch spans a perfect fourth or more (i.e., the highest pitch is more than 33% higher than the lowest pitch). Since this range is relatively large compare to the permissible error ($\pm 20\%$), these samples were excluded. Only samples for which the range did not span more than a major third (i.e., the highest pitch does not exceed the lowest pitch by more than 26%) were preserved, and they were assigned the pitch of the note corresponding to the median of the range. If the median was between two notes, the pitch of any of them was assigned to the sample. This should introduce an error no larger than two semitones (12%), which is about half the maximum permissible error of 20%. There were 30 samples for which the subject could not confidently perceive pitch; hence, those samples were also excluded.

The GERs on this database were first computed per sample (vowel), and then averaged over the samples. Since the ground truth data were based on the perception of only one listener, it could be argued that these data have low validity. To alleviate this, we excluded the samples for which the minimum GER was larger than 50%.

2. Musical instruments database

The files were downsampled from 44.1 to 10 kHz in order to reduce computational cost. No noticeable change of pitch was perceived by doing this, even for the highest pitch sounds.

The files were labeled assuming that their content was a chromatic scale. However, some of the intervals were imprecise in some of the files, leading to accumulated errors that exceeded one semitone, and consequently to wrong labels. This situation was especially common among string instruments, especially when playing *pizzicato*. To correct this situation, the first author listened to the files and relabeled them using an electronic keyboard as reference. This procedure introduced repeated file names, which were removed by keeping only the sounds whose pitch was closest to the target. When the conflicting files had notes whose pitches were equally close to the target, the file with the best sound quality was preserved. This removal of files was done to avoid the overhead of having to add extra symbols to the file names to allow for repetitions, which would have complicated the generation of scripts to test the algorithms. Since the process of manually correcting the names of the notes was very tedious, especially for pizzicato sounds, only the labels of bass and violin pizzicato files were fixed, and the cello and viola pizzicato sounds were excluded from the evaluation.

The commands issued for each of the algorithms were the following.

AC-P. To pitch (ac)…0.001 30 15 no 0.03 0.45 0.01 0.35 0.14 1666.

CC. To Pitch (cc)…0.001 30 15 no 0.03 0.45 0.01 0.35 0.14 1666.

ESRPD. pda input_file -o output_file -P -d 1 -shift 0.001 -length 0.0384 -fmax 1666 -fmin 30 -n 0 -m 0.

SHS. To pitch (shs)… 0.001 30 15 5000 15 0.84 1666 48.

SHR. [t,p]=shrp(x,fs,[30 1666],40,1,0.4,5000,0,0).

SWIPE′. [p,t]=swipep(x,fs, [30 1666], 0.001, 1/96, 0.1,-Inf).

YIN. p.minf0=30; p.maxf0=1666; p.hop=10; p.sr=10000; r=yin(x,p).

Since the range 30–1666 Hz was too large for the Speech Filing System algorithms (AC-S, ANAL, CEP, and RAPT), they were not evaluated on this database.

The GERs on this database were first computed per sample (i.e., note) and then averaged over the samples. However, there were some samples for which agreement on existence of pitch among the algorithms existed only at very few instants of time (only one in some cases). To avoid giving too much emphasis to these few instants, only samples in which the algorithms agreed that pitch existed in more than 50% of the time were used in the statistics.

3. Speech databases (KPD and PBD)

The commands issued for each of the algorithms were the following[7]

AC-P. To pitch (ac)…0.001 40 15 no 0.03 0.45 0.01 0.35 0.14 800.

AC-S. fxac input_file.

ANAL. fxanal input_file.

CC. To pitch (cc)…0.001 40 15 no 0.03 0.45 0.01 0.35 0.14 800.

CEP. fxcep input_file.

ESRPD. pda input_file -o output_file -L -d 1 -shift 0.001 -length 0.0384 -fmax 800 -fmin 40 -lpfilter 600.

RAPT. fxrapt input_file.

SHS. To pitch (shs)…0.001 40 15 1250 15 0.84 800 48.

SHR. [t,p]=shrp(x,fs,[40 800],40,1,0.4,1250,0,0).

SWIPE. [p,t]=swipe(x,fs,[40 800], 0.001, 1/96, 0.1,-Inf).

SWIPE′. [p,t]=swipep(x,fs,[40 800], 0.001, 1/96, 0.1, -Inf).

TEMPO. f0raw=exstraightsource(x,fs).

YIN. p.minf0=40; p.maxf0=800; p.hop=20; p.sr=fs; r=yin(x,p).

[1]This assertion can be inferred from the analysis by Camacho and Harris (2007).

[2]In this work, we will make use of the magnitude of the spectrum but not its phase. Hence, to abbreviate, the words *magnitude of* will be omitted, and whenever the word *spectrum* is found, it should be understood as *magnitude of the spectrum*.

[3]A normalized inner product equal to 1 is unattainable since side lobes will inevitably appear in the same region as the negative part of the cosine.

[4]It was downsampled to 10 kHz in our evaluation to save computational cost (see Appendix for details).

[5]The logarithm of the spectrum was not evaluated because it can be negative, which would make the normalization in Eq. (12) nonsense. Also, only the window types with the largest normalized inner product between the kernel and the square root of the spectrum evaluated over the whole period of the cosine for each value of k in Table I were evaluated. The flattop window was left out because its high value of k made its computational cost too high for our evaluation.

[6]For this test, the spectral analysis was limited to 5 kHz in *all* databases. This was done to discard the possibility that the use of a higher upper limit of frequency for speech (10–25 kHz) than for musical instruments (5 kHz) may have had an effect on the performance divergence (i.e., increase of performance on musical instruments but decrease of performance on speech) occurred in the tests where the flat envelope and the Hertz scales were evaluated separately.

[7]The command for CATE is not reported because we used our own implementation of the algorithm.

American Standards Association (**1960**). "Acoustical Terminology SI 1-1960," American Standards Association, New York.

Bagshaw, P. C., Hiller, S. M., and Jack, M. A. (**1993**). "Enhanced pitch tracking and the processing of F0 contours for computer aided intonation teaching," *Proceedings of the Third European Conference on Speech Communications and Technology*, pp. 1003–1006.

Bagshaw, P. C. (**1994**). "Automatic prosodic analysis for computer aided pronunciation teaching," Ph.D. thesis, University of Edinburgh, Edinburgh.

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

Camacho, A., and Harris, J. G. (**2007**). "A pitch estimation algorithm based on the smooth harmonic average peak-to-valley envelope," Proceedings of the International Symposium on Circuits and Systems, pp. 3940–3943.

Dannenberg, R. B., Birmingham, W. P., Tzanetakis, G. P., Meek, C. P., Hu, N. P., and Pardo, B. P. (**2004**). "The MUSART testbed for query-by-humming evaluation," Comput. Music J. **28**, 34–48.

De Bot, K. (**1983**). "Visual feedback of intonation I: Effectiveness and induced practice behavior," Lang Speech **26**, 331–350.

de Cheveigné, A., and Kawahara, H. (**2002**). "YIN, a fundamental frequency estimator for speech and music," J. Acoust. Soc. Am. **111**, 1917–1930.

Di Martino, J., and Laprie, Y. (**1999**). "An efficient *F*0 determination algorithm based on the implicit calculation of the autocorrelation of the temporal excitation signal," Proceedings of the Sixth European Conference on Speech Communication and Technology, pp. 2773–2776.

Doughty, J., and Garner, W. (**1947**). "Pitch characteristics of short tones. I. Two kinds of pitch threshold," J. Exp. Psychol. **37**, 351–365.

Duifhuis, H., Willems, L. F., and Sluyter, R. J. (**1982**). "Measurement of pitch in speech: An implementation of Goldstein's theory of pitch perception," J. Acoust. Soc. Am. **71**, 1568–1580.

Fant, G. (**1970**). *Acoustic Theory of Speech Production, With Calculations Based on X-Ray Studies of Russian Articulations* (Mouton De Gruyter, The Hague).

Glasberg, B. R., and Moore, B. C. J. (**1990**). "Derivation of auditory filter shapes from notched-noise data," Hear. Res. **47**, 103–138.

Hermes, D. J. (**1988**). "Measurement of pitch by subharmonic summation," J. Acoust. Soc. Am. **83**, 257–264.

Hess, W. (**1983**). *Pitch Determination of Speech Signals* (Springer-Verlag, Berlin).

Kawahara, H., Katayose, H., de Cheveigné, A., and Patterson, R. D. (**1999**). "Fixed point analysis of frequency to instantaneous frequency mapping for accurate estimation of F0 and periodicity," Proceedings of the Sixth European Conference on Speech Communication and Technology, pp. 2781–2784.

Klapuri, A. (**2004**). "Automatic music transcription as we know it today," J. New Music Res. **33**, 269–282.

Klapuri, A. (**2008**). "Multipitch analysis of polyphonic music and speech signals using an auditory model," IEEE Trans. Audio, Speech, Lang. Process. **16**, 255–266.

Medan, Y., Yair, E., and Chazan, D. (**1991**). "Super resolution pitch determination of speech signals," IEEE Trans. Signal Process. **39**, 40–48.

Meddis, R., and Hewitt, M. J. (**1991**). "Virtual pitch and phase sensitivity of a computer model of the auditory periphery. I: Pitch identification," J. Acoust. Soc. Am. **89**, 2866–2882.

Meddis, R., and O'Mard, L. (**1997**). "A unitary model of pitch perception," J. Acoust. Soc. Am. **102**, 1811–1820.

Noll, A. M. (**1967**). "Cepstrum pitch determination," J. Acoust. Soc. Am. **41**, 293–309.

Oppenheim, A. V., Schafer, R. W., and Buck, J. R. (**1999**). *Discrete-Time Signal Processing* (Prentice-Hall, Englewood Cliffs, NJ).

Patel, A. D., and Balaban, E. (**2001**). "Human pitch perception is reflected in the timing of stimulus-related cortical activity," Nat. Neurosci. **4**, 839–844.

Plante, F., Meyer, G., and Ainsworth, W. A. (**1995**). "A pitch extraction

reference database," Proceedings of the Fourth European Conference on Speech Communication and Technology, pp. 837–840.
Rabiner, L. R. (**1977**). "On the Use of Autocorrelation Analysis for Pitch Detection," IEEE Trans. Acoust., Speech, Signal Process. **25**, 24–33.
Schroeder, M. R. (**1968**). "Period histogram and product spectrum: New methods for fundamental frequency measurement," J. Acoust. Soc. Am. **43**, 829–834.
Schwartz, D. A., and Purves, D. (**2004**). "Pitch is determined by naturally occurring periodic sources," Hear. Res. **194**, 31–46.
Secrest, B., and Doddington, G. (**1983**). "An integrated pitch tracking algorithm for speech systems," Proceedings of ICASSP-83, pp. 1352–1355.
Sondhi, M. M. (**1968**). "New methods of pitch extraction," IEEE Trans. Audio Electroacoust. **AU-16**, 262–266.
Spanias, A. S. (**1994**). "Speech coding: A tutorial review," Proc. IEEE **82**, 1541–1582.
Sun, X. (**2000**). "A pitch determination algorithm based on subharmonic-to-harmonic ratio," Proceedings of the International Conference on Spoken Language Processing, Vol. **4**, pp. 676–679.
Traunmüller, H. (**1990**). "Analytical expressions for the tonotopic sensory scale," J. Acoust. Soc. Am. **88**, 97–100.
Wang, M., and Lin, M. (**2004**). "An analysis of pitch in Chinese spontaneous speech," International Symposium on Tonal Aspects of Tone Languages, Beijing, China.
Yumoto, E., Gould, W. J., and Baer, T. (**1982**). "Harmonics-to-noise ratio as an index of the degree of hoarseness," J. Acoust. Soc. Am. **71**, 1544–1549.