The SIFT Algorithm for Fundamental Frequency Estimation

JOHN D. MARKEL

Abstract—In this paper a new method for estimating F_0, the fundamental frequency of voiced speech versus time, is presented. The algorithm is based upon a simplified version of a general technique for fundamental frequency extraction using digital inverse filtering. It is demonstrated that the simplified inverse filter tracking algorithm (hereafter referred to as the SIFT algorithm) encompasses the desirable properties of both autocorrelation and cepstral pitch analysis techniques. In addition, the SIFT algorithm is composed of only a relatively small number of elementary arithmetic operations. In machine language, SIFT should run in several times real time while with special-purpose hardware it could easily be realized in real time.

I. Autocorrelation, Cepstral, and Inverse Filter Analysis

One of the oldest digital methods for estimating the fundamental frequency (F_0) of voiced speech is autocorrelation analysis. A window or frame of data of N samples, encompassing several pitch periods, is used to calculate the short-term autocorrelation sequence specified by

$$\rho_j = \sum_{n=0}^{N-1-|j|} s_n s_{n+|j|},$$

where $\{s_n\} = \{s_0, s_1, \cdots, s_{N-1}\}$ defines the input sequence obtained from the continuous-speech signal by sampling above the Nyquist frequency and $j=0,1, \cdots, N-1$. The N-length sequence $\{\rho_j\}$ can be efficiently calculated for large N if N is an integer power of two by applying a fast Fourier transform (FFT) in the following manner. Define $\{s'_n\}$ as the $N'=2N$-length sequence obtained by appending N zeros to the end of the sequence $\{s_n\}$. Then: 1) calculate $\{S'_k\} =$ FFT $\{s'_n\}$, 2) replace $\{S'_k\}$ by $\{|S'_k|^2\}$, and 3) calculate $\{\rho'_j\} =$ FFT $\{S'_k\}/N'$. Finally, $\rho_j = \rho'_j$, $j=0, 1, \cdots, N-1$. (Note that $\rho'_j - \rho'_{N'-j} = 0, 1, \cdots, N'/2$.) Since $\{s_n\}$ is real, the computation time can be further reduced by one half. F_0 is defined over the sequence of N samples as the reciprocal of the estimated pitch period P, where P is the location in time of the maximum ρ_j within some specified interval. An important property is that if P is known to lie within a specified interval, it is only necessary to search this interval of the autocorrelation sequence. Furthermore, the dynamic range of the pitch period peaks in the autocorrelation sequence is usually less than 10 dB. In contrast, pitch extraction from the acoustic wave itself, a common approach to tracking with analog devices, requires the detection of peaks over a dynamic range sometimes exceeding 30 dB.

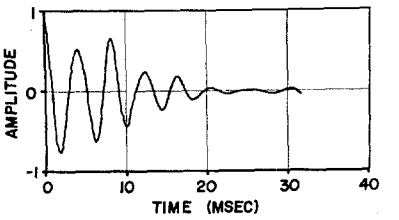

Fig. 1. Normalized autocorrelation results from a test segment of the vowel /i/ spoken in the phrase: "We were."

The acoustic speech waveform can be modeled as the convolution of terms that includes a periodic component due to the glottal waveform and a term representing the vocal tract impulse response. The resonances or formants of the vocal tract have narrow enough bandwidths (50–80) Hz for the autocorrelation sequence to frequently have several high-amplitude oscillations that interact with the component due to the pitch period. Fig. 1 shows the autocorrelation sequence obtained from a 32 ms segment of the vowel/i/spoken in the context: "We were." This segment has been analyzed under different conditions for comparative purposes and will hereafter be referred to as the test segment. The zero-time sample is always the largest amplitude term and therefore, an autocorrelation sequence can always be normalized to unity at the origin. The pitch period can be seen by the slight increase in the third positive peak. In general, the pitch peak detection is nontrivial, and in addition, the estimate will be somewhat in error due to the interaction of the glottal wave component and the damped sinusoidal term due predominantly to the first formant [2].

The single property $\log ab = \log a + \log b$ leads to what has been termed cepstral analysis, a suggested solution to the fundamental frequency extraction problem. The cepstrum is calculated identically as the autocorrelation previously described using two FFT's except that in step 2), instead of replacing $\{S'_k\}$ by $\{|S'_k|^2\}$, $\{S'_k\}$ is replaced by $\{\log |S'_k|^2\}$. With this trivial modification, dramatically different results are obtained. Fig. 2 shows the cepstrum for the test segment. Fig. 2(a) shows the cepstrum normalized to unity at the origin while Fig. 2(b) shows the cepstrum normalized to the pitch peak after the first two milliseconds were zeroed out. The sharp peak at 8.3 ms is due to the pitch period. To the right of the first few samples, the pitch period in a voiced segment can usually be uniquely defined by the largest peak in the cepstrum. The effects of the vocal tract impulse response are contained largely within the first few milliseconds of the origin [3]. Thus

Manuscript received May 8, 1972. This work was supported by the Office of Naval Research under Contract N00014-67-C-0118.

The author is with Speech Communications Research Laboratory, Inc. (SCRL), Santa Barbara, Calif. 93101.

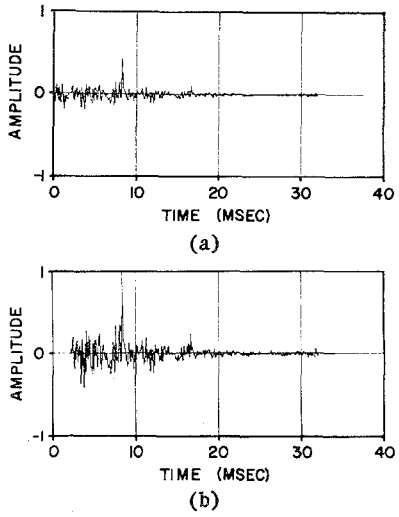

Fig. 2. Cepstral analysis results. (a) Normalized to unity at origin. (b) Normalized to unity at peak where first two milliseconds have been zeroed out.

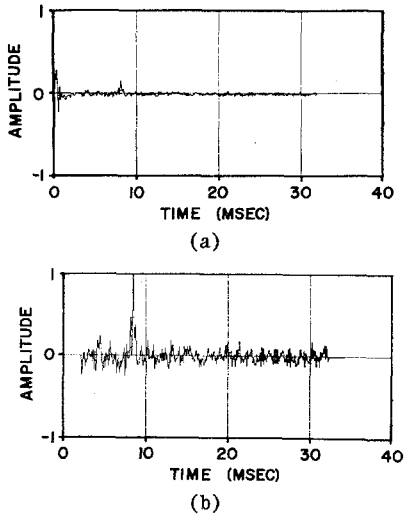

Fig. 3. Inverse filter analysis results. (a) Normalized to unity at the origin. (b) Normalized to unity at peak where first two milliseconds have been zeroed out.

the problem of interaction between the formants and fundamental frequency has been largely solved. Unfortunately, because of the nonlinear logarithmic operation, two undesirable results are obtained: 1) the peak at the origin can no longer be used as a reference for normalization, and 2) the actual amplitude of the spike is a function not only of the number of pitch periods within the window, but also of the spectral shape. The zero-time value of the peak is dependent only upon the mean value of the log magnitude spectrum. Spectral shaping is dependent largely upon the formant values and to a first order approximation can be considered independent of the fundamental frequency.

If F_0 tracking is accomplished manually or if the speech segment is completely voiced, these criticisms are irrelevant. Generally, however, it is desirable to automatically determine whether the segment is voiced (in which case F_0 is to be calculated) or unvoiced (in which case F_0 does not exist). It has been stated that voicing is detected by a sharp spike at the pitch period while unvoicing is detected by the absence of a sharp spike [4]. Although this statement when used as an algorithm works extremely well for manual calculation, automatic implementation is not trivial for the two reasons stated above.

What is proposed in this paper is a simplified analysis technique, based upon an inverse filter formulation [1] which retains the advantages of both the autocorrelation and cepstral analysis techniques. The results from an inverse filter analysis of the test segment are shown in Fig. 3. Fig. 3(a) shows the output normalized to unity at the origin, while Fig. 3(b) shows the output normalized to the pitch peak after the first 2 ms were zeroed out. This output sequence is defined as the autocorrelation of the inverse filter output and thus can be normalized so that the units on the ordinate correspond to correlation values from -1.0 to 1.0. There is a sharp peak corresponding to a correlation value of 0.43 at 8.3 ms. The pitch period is defined as the location of this peak, and generally it will have the largest correlation over all samples except at the origin. Since it is always possible to normalize the output, and since the data values have the physical interpretation of correlation, it should be possible to define a simple voiced-unvoiced decision based upon a fixed threshold value.

In Figs. 2(b) and 3(b), the waveforms are normalized to the pitch peak so that the peak signal-to-noise ratios can be compared over the interval (2, 32) ms. Since negative correlation values can never be possible candidates for pitch estimators, the peak undesired noise amplitude is 0.30 in the autocorrelation sequence. For the cepstrum in the interval (2, 32) ms, the maximum undesired peak is 0.25. Note that the first few milliseconds of the cepstrum can have very large positive or negative terms. One additional comment pertaining to the cepstrum is in order. Squaring of the cepstral samples has been suggested in the literature [5]. By squaring the samples, certainly a much more attractive representation is obtained. Unfortunately, this does not accomplish any desirable goal, since the dynamic range over which detection must be accomplished is also squared. The results for the SIFT algorithm to be developed in this paper are shown in Fig. 4 for the test segment. The properties are quite similar to the inverse filter analysis shown in Fig. 3. The major differences are: 1) the peak is broadened slightly; 2) the higher frequency terms in the undesired portion of the output are suppressed; 3) the harmonics of the pitch period are more apparent since a Hamming window was not applied to the input data; and 4) the desired signal-peak-to-undesired-noise-peak ratio is increased.

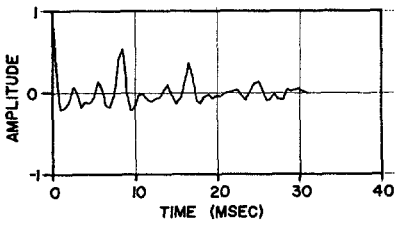

Fig. 4. SIFT algorithm results for the analysis of the test segment.

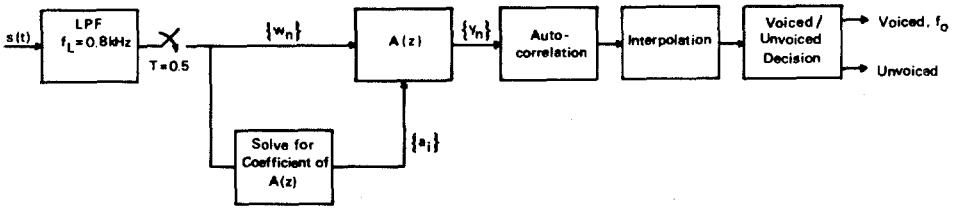

Fig. 5. Block diagram of the SIFT algorithm.

II. The SIFT Algorithm

A Block Diagram Description

A block diagram of the SIFT analysis system is shown in Fig. 5. The speech waveform $s(t)$ is first prefiltered by a low-pass filter with a cutoff at 0.8 kHz. After sampling the filter output at a 2-kHz rate, the first five terms $\rho_j, j=0, 1, \cdots, 4$, of the short-term autocorrelation sequence are calculated for an appropriate input length (as a representative number $N=64$ was chosen corresponding to a 32-ms window). The set of linear equations $\sum_{i=1}^{4} a_i\rho_{i-j} = -\rho_j$, $j=1, 2, \cdots, 4$, is then solved for the inverse filter coefficients $\{a_i\}$, where the inverse filter is defined by $A(z)=1+\sum_{i=1}^{4} a_i z^{-1}$. Knowing $\{a_i\}$, the inverse filter output $\{y_n\}$ can be calculated. The output autocorrelation sequence $\{r_n\}$ from which F_0 is estimated is then calculated as the autocorrelation sequence of $\{y_n\}$. After $\{r_n\}$ is obtained, the largest peak within specified limits is found. Interpolation is applied in the region of the peak and then a voiced-unvoiced decision is made based upon the interpolated peak. If the segment is voiced, the reciprocal of the location of the interpolated peak defines F_0. Each of these operations, along with computational considerations, will be discussed in detail.

Prefiltering

A sampling frequency of 10 kHz is often used in digital speech analysis to insure that all significant frequency components of voiced speech are accurately represented. This sampling rate also insures adequate time scale resolution (0.1 ms) for accurate estimation of P. By deriving a sampling theorem that corresponds to a form of trignometric interpolation, accurate estimation of P will be shown possible even with a 2-kHz sampling rate chosen for the analysis. By using this low sampling frequency, the total number of necessary operations is greatly reduced.

To insure against folding over of frequency components (aliasing) into the (0, 1) kHz range, it is necessary that the input signal be bandlimited to 1 kHz. Aliasing problems are minimal for voiced speech since the spectrum of a voiced sample will always have a maximum peak in the range (0, 1) kHz with the largest peak outside the range, generally 5–10 dB below the first peak. For unvoiced speech, however, such a situation will not usually exist. For example, the peak during an /s/ may be located at 5 kHz with an amplitude 30 dB above the low-frequency components. Unless rather elaborate (and time-consuming) digital filtering is employed, the filter cutoff must be chosen as somewhat less than 1 kHz. A cutoff at 0.8 kHz is a reasonable choice for including most of the low-frequency range while providing sufficient attenuation at 1.0 kHz. To demonstrate the fact that extremely sharp cutoff filters are not necessary and that phase and group delay characteristics are not critical, a Chebyshev 3-pole 2-dB ripple filter specification has been used.

Actually, for the simulation a digital version was implemented, since a 10-kHz sampling rate was most readily available. The digital filter is specified by $u_n = a_1 s_n + a_2 u_{n-1}$ and $x_n = a_3 u_n + a_4 x_{n-1} + a_5 x_{n-2}$ where

$$a_1 = 1 - e^{-\alpha_1 T}$$
$$a_2 = e^{-\alpha_1 T}$$
$$a_3 = 1 - 2e^{-\alpha_2 T}\cos\beta_2 T + e^{-2\alpha_2 T}$$
$$a_4 = 2e^{-\alpha_2 T}\cos\beta_2 T$$
$$a_5 = -e^{-2\alpha_2 T}$$
$$\alpha_1 = (0.3572)2\pi f_c$$
$$\alpha_2 = (0.1786)\pi f_c$$

$$\beta_2 = (0.8938)\pi f_c$$
$$u_n = 0, \qquad n < 0$$
$$x_n = 0, \qquad n < 0$$

and $\{s_n\}$ and $\{x_n\}$ are the input and output sequences, respectively, $f_c = 0.8$ kHz, and $T = 0.1$ ms. To convert the samples to a 2-kHz rate, assuming the input is sampled at 10 kHz, a new sequence $\{w_n\}$, made up of every fifth sample of $\{x_n\}$, is defined.

Fig. 6 shows the discrete frequency response of the digital filter. Some aliasing obviously occurs since the response is down only about 10 dB at the folding frequency. If a sharper cutoff filter is applied, less aliasing will occur and slightly better results will be obtained. Nonetheless, it is demonstrated that accurate simulation results are obtained even with this filter.

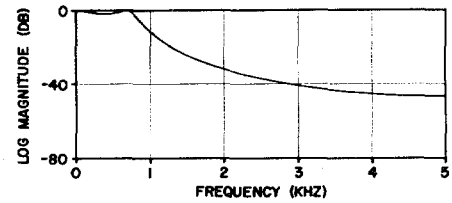

Fig. 6. Frequency response of 3-pole 2-dB ripple Chebyshev filter used in SIFT.

Determination of the Inverse Filter

The general form of the inverse filter $A(z) = 1 + \sum_{i=1}^{\overline{M}} a_i z^{-i}$ is defined by determining the coefficients $\{a_i\}$ such that the difference between a constant and the filter output, $Y(z)$, is minimized in the least squares sense. Using this criterion the inverse filter will attempt to transform the input spectrum into a white noise, or a constant, spectrum. To within an irrelevant gain constant, this criterion is equivalent to minimizing the energy output of the filter $A(z)$. Thus the coefficients can be determined from

$$\frac{\partial}{\partial a_k} \sum_{n=1}^{L_1} y_n^2 = \frac{\partial}{\partial a_k} \sum_{n=0}^{L_1} \left(x_n + \sum_{i=1}^{\overline{M}} a_i x_{n-i} \right)^2 = 0,$$
$$k = 1, 2, \cdots, \overline{M}$$

with the solution given by the autocorrelation equations

$$\sum_{i=1}^{\overline{M}} a_i p_{i-j} = -p_j, \qquad i = 1, 2, \cdots, \overline{M}$$

where the autocorrelation coefficients p_j are calculated from $\{w_n\}$ by $p_j = \sum_{n=0}^{N-1-j} w_n w_{n+j}$, $j = 0, 1, \cdots, \overline{M}$, and $L_1 = N + \overline{M} - 1$. The success of the method is strongly dependent upon a proper choice of $\overline{M}$ the number of undetermined filter coefficients and N the number of data samples. If $\overline{M}$ is too small, very poor estimation of the resonance structure within the range $(0, F_s/2)$ is obtained, where F_s is the sampling frequency. If $\overline{M}$ is too large, then the fine-grain structure (due to the pitch periods) is estimated along with the resonance structure (in the limit, if $\overline{M}$ is large enough, the input will be transformed into a Kronecker delta function at the origin). What is desired is to obtain a close estimation only of the resonance structure, and to ignore the finegrain structure. N should be large enough to include several pitch periods but small enough to insure that significant pitch period variations do not occur.

For this study $\overline{M} = 4$ and $N = 64$ have been chosen. With $\overline{M} = 4$, either zero, one, or two resonances can be accurately represented. The minimum number of resonances possible within (0, 1) kHz is zero (due to unvoiced speech) while the maximum is two (due to voiced speech with close first and second formants). With $N = 64$, a maximum interval of 32 ms can be represented, corresponding to approximately three pitch periods of a normal male voice. Additional considerations in the choice of N and $\overline{M}$ are presented elsewhere [6]. The set of autocorrelation equations is most efficiently solved for general $\overline{M}$ by use of Levinson's method [7]. With four coefficients, however, it is possible to quite easily obtain a closed form solution for the $\{a_i\}$. The set of linear equations to be solved is

$$a_1 p_0 + a_2 p_1 + a_3 p_2 + a_4 p_3 = -p_1$$
$$a_1 p_1 + a_2 p_0 + a_3 p_1 + a_4 p_2 = -p_2$$
$$a_1 p_2 + a_2 p_1 + a_3 p_0 + a_4 p_1 = -p_3$$
$$a_1 p_3 + a_2 p_2 + a_3 p_1 + a_4 p_0 = -p_4.$$

By adding the first equation to the last, and the second equation to the next to last,

$$\alpha_{14}(p_0 + p_3) + \alpha_{23}(p_1 + p_2) = -(p_1 + p_4)$$
$$\alpha_{14}(p_1 + p_2) + \alpha_{23}(p_0 + p_1) = -(p_2 + p_3)$$

where $\alpha_{14} = a_1 + a_4$ and $\alpha_{23} = a_2 + a_3$. If instead of adding, the corresponding equations are subtracted,

$$\beta_{14}(p_0 - p_3) + \beta_{23}(p_1 - p_2) = -(p_1 - p_4)$$
$$\beta_{14}(p_1 - p_2) + \beta_{23}(p_0 - p_1) = -(p_2 - p_3)$$

where $\beta_{14} = a_1 - a_4$ and $\beta_{23} = a_2 - a_3$.

Solving each of the two sets of second-order equations gives the $\{a_i\}$ as

$$a_1 = (\alpha_{14} + \beta_{14})/2$$
$$a_2 = (\alpha_{23} + \beta_{23})/2$$
$$a_3 = (\alpha_{23} - \beta_{23})/2$$
$$a_4 = (\alpha_{14} - \beta_{14})/2.$$

Since $|A(e^{j\omega T})|^2 = |1 + \sum_{i=1}^{\overline{M}} a_i e^{-j\omega T}|^2$ defines the spectrum of the inverse filter, $|D(e^{j\omega T})|^2 = |1/A(e^{j\omega T})|^2$ defines the estimate of the resonance behavior of the inverse filter input spectrum. The reciprocal of the inverse filter spectrum, $|D(e^{j\omega T})|^2$, is shown with the input spectrum on an expanded logarithmic scale for the test segment in Fig. 7. The first formant peak is clearly predicted. The primary peaks every 120 Hz under the smooth envelope are due to the periodicity of the waveform, while the secondary peaks every 31 Hz are due to the 32-ms length of data.

After the inverse filter is specified for a particular

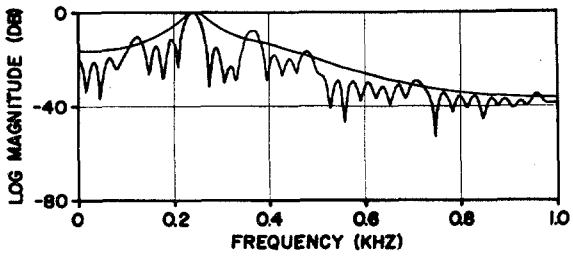

Fig. 7. Spectrum of SIFT inverse filter input and reciprocal of inverse filter.

frame, the output $\{y_n\}$ corresponding to the error or deviation from a white noise sequence is calculated as

$$y_n = w_n + \sum_{i=1}^{4} a_i w_{n-i} \quad n = 0, 1, \cdots, 63,$$

where $w_n = 0, n < 0.$

Although $\{y_n\}$ will have nonzero values out to $n = N + \overline{M}$, they will not contribute to the pitch estimates and are thus ignored. Fig. 8 shows the spectrum of the inverse filter output for the test segment. The major resonance behavior has been completely removed leaving only the fundamental frequency information superimposed upon a constant. Thus, the inverse filter can be considered as a prewhitening filter which attempts to whiten the input spectrum by eliminating the trend characteristics or spectral shaping (due predominantly to the vocal tract resonances or formants) while retaining the fine structure due to the glottal pulses. Note that since the length of the inverse filter is constrained to a small value, it would probably be more correct to say the inverse filter acts as a pseudo-prewhitening filter since the output obviously does not have a purely constant spectrum. Actually, the glottal pulses also have trend characteristics which are removed by the inverse filter. Thus if the input waveform were synthesized according to Fant's [8] model, the acoustic speech wave at the output to the inverse filter would be transformed into Kronecker delta functions at the initiation of each pitch period. Fig. 9 illustrates results for the SIFT algorithm as seen at the output of the inverse filter, compared with the input test segment. It is seen that the formant structure (the damped sinusoidal characteristic) has been removed and fairly sharp pulses at the initiation of each pitch period are obtained at the inverse filter output. (It is easily shown that the width of the pitch period spike will be approximately $1/f_c$ ms where f_c is the filter cutoff in kilohertz.) Although an attempt could be made to estimate the pitch period directly from the inverse filter output, it is not recommended for the same reasons that direct pitch estimation from the acoustic waveform is not recommended, as discussed earlier.

The pitch period estimates are finally obtained by a standard autocorrelation method, only the input signal now has the resonance structure or formant information eliminated. The autocorrelation sequence $\{r_n\}$ calculated at the output of the inverse filter is of length $M = 2N$ and is given by

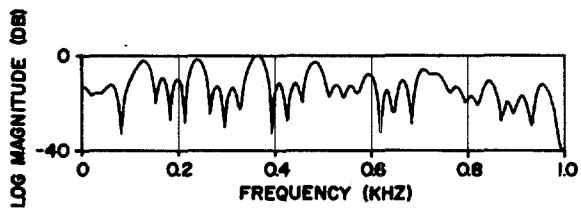

Fig. 8. Spectrum of SIFT inverse filter output.

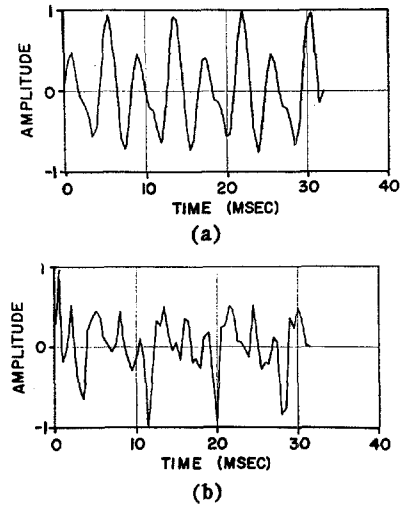

Fig. 9. Inverse filter waveforms from SIFT. (a) Input waveform. (b) Output waveform.

$$r_n = \begin{cases} \sum_{j=0}^{N-1-n} y_j y_{j+n}, & n = 0, 1, \cdots, M/2-1 \\ 0, & n = M/2 \\ r_{M-n}, & n = M/2+1, M/2+2, \cdots, M-1. \end{cases}$$

It is assumed that the initial pitch period estimate is known within $\pm$ 4 ms. Thus for each frame, a total of 16 autocorrelation samples are necessary (including r_0, used for normalization). When a peak $\hat{r}$ is obtained at $n = \hat{N}$, r_n is evaluated for $n = \hat{N}-7, \hat{N}-6, \cdots, \hat{N}, \cdots, \hat{N}+6, \hat{N}+7$, and $n = 0$ in the following frame.

If an unvoiced decision is made, $\hat{N}$ is reset to the initial value chosen. For most speech, an 8 ms range is quite sufficient. For a male speaker with an average 8 ms pitch period, calculation of 15 autocorrelation values encompasses a range of 83–250 Hz. Because the pitch samples are tracked from frame to frame, a much larger range of F_0 is effectively obtained. If, however, a greater range of uncertainty exists on F_0, it is only necessary to calculate additional autocorrelation terms from the inverse filter output.

Interpolation

Accurate measurement of fundamental frequency requires a time scale resolution of approximately 0.1 to 0.15 ms. If T had been chosen as 0.125 ms, and the true pitch P were 6 ms for example, the maximum

quantization error would be approximately $T/2P^2 = 1.74$ Hz. For the chosen value of $T = 0.5$ ms, the maximum quantization error of 7.0 Hz is large enough to be quite noticeable in synthetic speech. The straightforward approach to this problem of decreasing the sampling period greatly increases the computation time, and also then makes it necessary to eliminate the resonance effects of the higher formants by designing a considerably larger filter. This approach is discussed elsewhere [1]. A simplified solution to this problem can be obtained by deriving a trigonometric interpolation function for $\{r_n\}$, the autocorrelation sequence obtained from the inverse filter. For simplicity it is assumed that M is a power of 2. Then the $M = 2N$ length sequence $\{r_n\}$ has a discrete Fourier transform (DFT) given by

$$R_k = \sum_{n=0}^{M-1} r_n e^{-j2\pi nk/M}, \qquad k = 0, 1, \cdots, M-1.$$

But since $\{r_n\}$ is an autocorrelation sequence, it must be real symmetric in the sense that $r_n = r_{M-n}$, $n = 0, 1, \cdots, M/2$, and thus, $\{R_k\}$ must be real and symmetric, resulting in

$$R_k = 2 \sum_{l=0}^{M/2-1} u_l \cos \frac{2\pi lk}{M}, \quad k = 0, 1, \cdots, M-1 \tag{1}$$

where

$$u_l = \begin{cases} \frac{1}{2} r_0, & l = 0 \\ r_l, & l = 1, 2, \cdots, M/2 - 1. \end{cases} \tag{2}$$

The inverse relationship is

$$r_n = \frac{1}{M} \sum_{k=0}^{M-1} R_k e^{j2\pi nk/M}$$

$$= \frac{1}{M}\left[R_0 + R_{M/2} e^{j\pi n} + 2 \sum_{k=1}^{M/2-1} R_k \cos(2\pi nk/M)\right]. \tag{3}$$

Now, if a new sequence $\{\hat{R}_k\}$ of length $M' = 2^I M$ is defined as

$$\hat{R}_k = \begin{cases} R_k, & k = 0, 1, \cdots, M/2 \\ 0, & k = M/2 + 1, \cdots, M' - M/2 \\ R_{M'-k}, & k = M' - M/2 + 1, \cdots, M' - 1 \end{cases} \tag{4}$$

where I is a positive integer, then $\hat{r}_n$ is given by

$$\hat{r}_n = \frac{1}{M'} \sum_{k=0}^{M'-1} \hat{R}_k \exp\left(\frac{j2\pi kn}{M'}\right), \quad n = 0, 1, \cdots, M' - 1$$

$$\frac{1}{M'}\left[R_0 + R_{M/2} \exp\left(\frac{j\pi nM}{M'}\right) + 2 \sum_{k=1}^{M/2-1} R_k \cdot \cos\left(\frac{2\pi k}{M} \frac{nM}{M'}\right)\right]$$

$$= \frac{M}{M'} r_{\frac{nM}{M'}}, \qquad n = 0, 1, \cdots, M' - 1. \tag{5}$$

Thus $\{r_n\}$ can be interpolated efficiently with the use of two FFT's. The interpolated sequence $\{\hat{r}_n\}$ can also be derived explicitly in terms of $\{r_n\}$. Substituting (1) into (5),

$$r_a = \frac{1}{M}\left[2 \sum_{l=0}^{M/2-1} u_l + 2e^{j\pi a} \sum_{l=0}^{M/2-1} u_l (-1)^l + 4 \sum_{k=1}^{M/2-1} \sum_{l=0}^{M/2-1} \mu_l \cos \frac{2\pi lk}{M} \cos \frac{2\pi ak}{M}\right]$$

where $a = nM/M'$. It is reasonable to assume that the folding frequency term $R_{M/2}$ does not substantially contribute to the final results if $M \gg 1$. By interchanging the summations on k and l and writing the cosines in exponential form, geometric progressions are obtained which result in

$$r_a = \frac{1}{M}\left\{2 \sum_{l=0}^{M/2-1} \mu_l + \sum_{l=0}^{M/2-1} \mu_l \cdot \left[\frac{\sin \alpha(M-1)/2 - \sin \alpha/2}{\sin \alpha/2} + \frac{\sin \beta(M-1)/2 - \sin \beta/2}{\sin \beta/2}\right]\right.$$

This equation reduces very neatly into the interpolation formula

$$r_a = \frac{1}{M} \sum_{l=0}^{M/2-1} \mu_l \left[\frac{\sin \alpha(M-1)/2}{\sin \alpha/2} + \frac{\sin \beta(M-1)/2}{\sin \beta/2}\right] \tag{6}$$

where

$$\alpha = \frac{2\pi}{M}(l + a)$$

$$\beta = \frac{2\pi}{M}(l - a)$$

$$a = \frac{nM}{M'}, \qquad n = 0, 1, \cdots, M' - 1.$$

$M = 2N$ and u_l is defined by (2). If interpolation over the full range of the term a is desired, it is much faster to use the FFT twice, as indicated by (1)–(5). For this application, however, the characteristics of the resulting autocorrelation sequence allow a much faster solution by the direct application of (6). For voiced speech, the largest peak away from the origin at $n = \hat{N}$ will generally define the pitch period within ± 1 sample. Since the peak is usually about three samples wide (assuming $f_c = 0.8$ kHz and $T = 0.5$ ms), a very reasonable approximation to the interpolated values between $\hat{N} - 1$ and $\hat{N} + 1$ can be accomplished by using only $r_{\hat{N}-1}$, $r_{\hat{N}}$, and $r_{\hat{N}+1}$ and considering all other terms as zero. This is possible because of the rather rapid decay of the interpolation function (assuming $\hat{N} > 3$ or 4 which will be the case except for extremely high F_0). Fig. 10 is a graph of r_a versus a for $r_n = \delta_{n,16}$ on a normalized time scale where $a = n/4$, $n = 0, 1, 2, \cdots, N$. The distance between zero crossings, away from the main peak, precisely defines

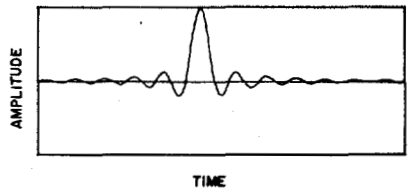

Fig. 10. Representation of interpolation function. Zero crossings occur at sampling interval.

the sampling period. It is seen from (6) that the exact values of the interpolation function are dependent upon l. In the region 2–16 ms, however, the differences are relatively small. By referring $\hat{N}$ to some fixed point and defining the interpolation ratio M'/M, it is possible to obtain a considerably simplified set of algebraic equations. If a 4 to 1 interpolation ratio is used, discrete samples will be calculated at intervals of 0.125 ms. By defining $n=16$ in (6) as the reference point (corresponding to 8 ms), and noting symmetry relationships, the simplified interpolation equations can by written in matrix form as

$$[\gamma_{\pm 3/4} \quad \gamma_{\pm 1/2} \quad \gamma_{\pm 1/4}]^T = A[\gamma_{\pm 1} \quad \gamma_0 \quad \gamma_{\mp 1}]^T \qquad (7)$$

where

$$A = \begin{bmatrix} 0.879124 & 0.321662 & -0.150534 \\ 0.637643 & 0.636110 & -0.212208 \\ 0.322745 & 0.878039 & -0.158147 \end{bmatrix}$$

and $\gamma_a = r_{\hat{N}+a}/r_0$.

An example of 4/1 interpolation using (7) is shown in Fig. 11 on a normalized time scale. The discrete samples with large dots indicate the uninterpolated pitch peak estimate and adjacent terms from the autocorrelation calculation $\{r_n\}$, normalized by r_0. The interpolated samples are indicated by the small dots, while the solid line indicates the continuous curve obtained by evaluating the general interpolation equation (6). The peak value over the seven samples $\gamma_{i/4}$, $i=0, \pm 1, \pm 2, \pm 3$, is defined as $\hat{\gamma}$.

In practice $\gamma_{\pm 3/4}$ will never be the peak value $\hat{\gamma}$ and thus need not be calculated. The slope measured from γ_0 will indicate the direction of the peak and thus only three terms must be calculated per frame ($\gamma_{\pm 1/4}$ and $\gamma_{1/2}$ if $\gamma_{1/4} > \gamma_{-1/4}$ or $\gamma_{\pm 1/4}$ and $\gamma_{-1/2}$ if $\gamma_{1/4} < \gamma_{-1/4}$). The interpolated F_0 measurement is finally given by $F_0(\text{kHz}) = 1/P$ where $P = (\hat{N}+\hat{a})/2$ and $\hat{a}$ is the index corresponding to $\hat{\gamma}$.

Decision Criteria

If the pitch period were constant over a very large number of input samples, pitch detection would be trivial since the autocorrelation calculation would average out the undesired terms (defined as noise) to essentially zero value. Assuming the noise to be Gaussian in nature, it is possible to predict a threshold value to assure that the probability of any noise sample in the autocorrelation sequence is less than some specified value, as a function of the number of samples N [1]. The result for $N=64$, assuming an error probability of 0.001, is 0.378 for the threshold. It is shown in the section on experimental results that a threshold setting of 0.378 to 0.400 is quite realistic.

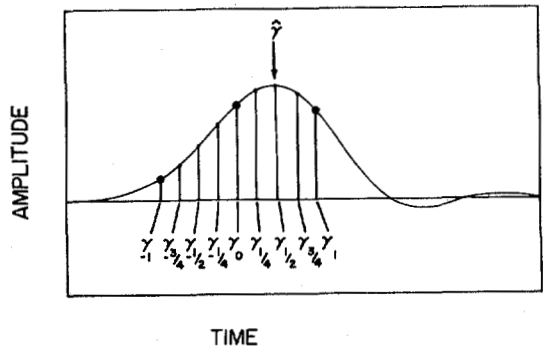

Fig. 11. Illustration showing 4/1 interpolation about the points γ_{-1}, γ_0, γ_1, $\hat{\gamma}$ indicates the interpolated peak estimate.

If the inverse filter output is modeled as a periodic Kronecker delta train, the normalized autocorrelation sequence is described by a linear function of the period with a decreasing slope. For the parameters suggested, we have found that the estimated pitch period peak $\hat{\gamma}$ for voiced speech can be reasonably well described by $\hat{\gamma} = -0.03P + 0.9$ in many cases, where $2 \leq P \leq 16$ ms. This equation assumes, of course, that within the 64-point window, pitch period variations are small. With a threshold setting of 0.4, pitch period estimates of up to 16 ms can therefore be obtained. (In the actual implementation P is estimated from $(\hat{N}+\hat{a})/2$.)

Under most conditions, a voiced-unvoiced decison can be made by simply testing to see if $\hat{\gamma}$ is greater than some predefined threshold. If so, the segment is defined as voiced. Otherwise, it is defined as unvoiced. However, anomalies can occur, such as a peak in a voiced segment being slightly below the threshold, whereas the preceding and following segments cross the threshold.

A simple decision algorithm for determining whether a particular frame is voiced or unvoiced is shown in Fig. 12. Whenever a peak exceeds the threshold value of 0.4, frame k is defined as voiced. Occasionally it is possible for a voiced frame to be incorrectly defined as unvoiced due to either considerable variation in the pitch period values or phonetic variations within the window. If this isolated condition is detected, frame $k-1$ is redefined as voiced with a pitch period equal to the mean of frame k and frame $k-2$.

If the peak in frame k does not cross the threshold, it is tested to see if the previous two frames are unvoiced. If so, frame k is unvoiced. If the previous two frames are voiced, then the threshold is lowered by 25 percent. At the end of a voiced phrase, part of the segment may be voiced and part unvoiced. This decision algorithm favors the voiced portion. In addition, fairly rapid changes in the pitch period values within a window can cause the peak correlation value to be decreased

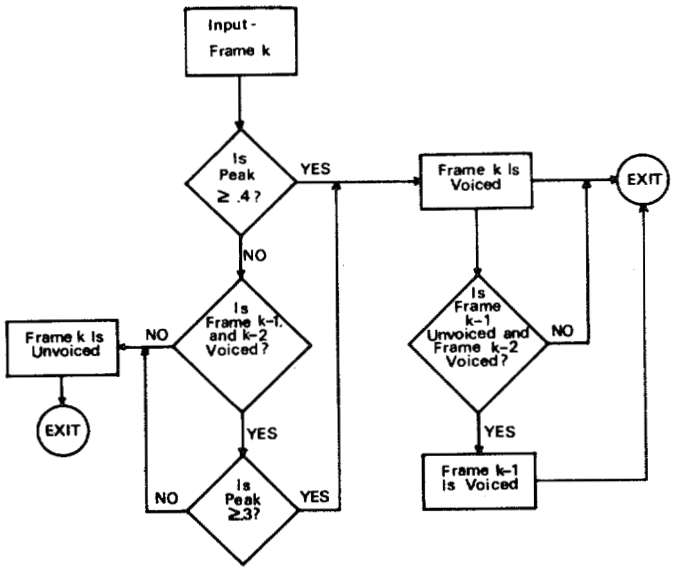

Fig. 12. Decision algorithm for voiced-unvoiced decision.

by 15–20 percent. If the peak still does not cross the threshold, frame k is defined as unvoiced. Otherwise it is defined as voiced.

Calculation of Necessary Operations

Let t_a and t_m denote the computer add and multiply times, respectively, in microseconds. It is reasonable to assume for simplicity that a subtract operation is equivalent to an add and a divide is roughly equivalent to a multiply. Calculation of the five correlation coefficients from the 64-point input sequence requires 320 (t_a+t_m) μs. Calculation of the inverse filter coefficients requires $20t_a+16t_m$ μs if the terms are efficiently grouped. Performing the inverse filtering of the input signal requires $256t_m+320t_a$ μs. The autocorrelation calculation of the inverse filter output requires $1024(t_a+t_m)$ μs. Finally, the interpolation and peak picking require roughly $69t_a+9t_m$ μs (for simplicity, a comparison is assumed equivalent to an addition). Thus, for real time total arithmetic computation $1750t_m+1625t_a \leq 1000P_f$, where P_f (ms) is the time interval between frames of data. If $P_f=15$ ms, $t_m=5$ μs and $t_a=3$ μs, real-time processing can be accomplished (assuming that overhead functions such as fetching and storing are included in the add and multiply times). Thus, it should be possible to operate in close to a real time environment even with the popular minicomputers if programming is done in assembler language and inefficient software floating point subroutines can be avoided. With the design of special-purpose hardware, it should be possible to easily attain a real time digital fundamental frequency tracker using the SIFT algorithm. As a further advantage of the method, the algebra involved is quite simple and requires only the four basic arithmetic operations without any table lookups.

In contrast, the cepstral analysis method as described by Schafer and Rabiner [9] uses two 1024-point real FFT's that require at least $(3t_a+2t_m)10^4$ μs. Thus, the SIFT algorithm is greater than an order of magnitude faster than the cepstral analysis pitch extraction method for the assumed conditions of $t_m=5$ μs and $t_a=3$ μs. With both algorithms programmed in Fortran on an IBM 1800, the SIFT algorithm was found to be approximately 20 times faster, even with efficient real FFT computation.

III. Experimental Results

To illustrate the capability of the SIFT algorithm for voiced speech, the exclamatory "oh" with a sharply rising and falling F_0 was spoken and analyzed. The data was also cepstrally analyzed for comparison purposes. The input data was analyzed at 16 ms intervals using a 32 ms sliding window.

The cepstral estimates were obtained by multiplying the data by a 32-ms Hamming window. A sampling rate of 10 kHz was used, to allow a 0.1 ms time scale resolution. Since the utterance is all voiced, the cepstral estimate for F_0 was defined by $F_0=1/P$ where P is the location of the cepstral peak in the range (2, 16) ms. The SIFT algorithm estimates for this example were also made by simply searching the output sequence for a peak in the range (2, 16) ms.

Results are shown in Fig. 13 where the triangles indicate cepstral estimates and the dots indicate SIFT estimates. The points are shown at 32 ms increments. The analysis covers an extremely wide range of fundamental frequency, from approximately 70–330 Hz. This range encompasses the vast majority of fundamental frequency range from adult male speech to children's speech. With the exception of a few frames of data, the results are seen to be quite close. There is one gross error due to the cepstral analysis at frame zero and one due to the SIFT analysis at frame 20, where the second harmonic of the pitch period had slightly higher amplitude than the fundamental. The average deviation between the two curves (excluding the two gross errors) is approximately 3 Hz. These results are obtained in spite of the fact that the second formant frequency F_2 varies from slightly above to slightly below the folding frequency in the SIFT algorithm. For several frames of data, namely frames 12–21, a component due to F_2 occurs in the autocorrelation of the inverse filter output. If sharper input filtering is applied, the results at the high fundamental frequency will compare favorably with the cepstral results.

To demonstrate typical results obtainable from SIFT where both voiced and unvoiced sounds occur, the phrase "put she can," spoken with each word emphasized, was analyzed. Again, for comparative purposes, cepstral analysis was performed on the same frames of input data. Within any range, say (2, 16) ms, a peak can be found for cepstral output. So that no difficulties would be encountered in the voiced-unvoiced decision using cepstral analysis, the decision was made manu-

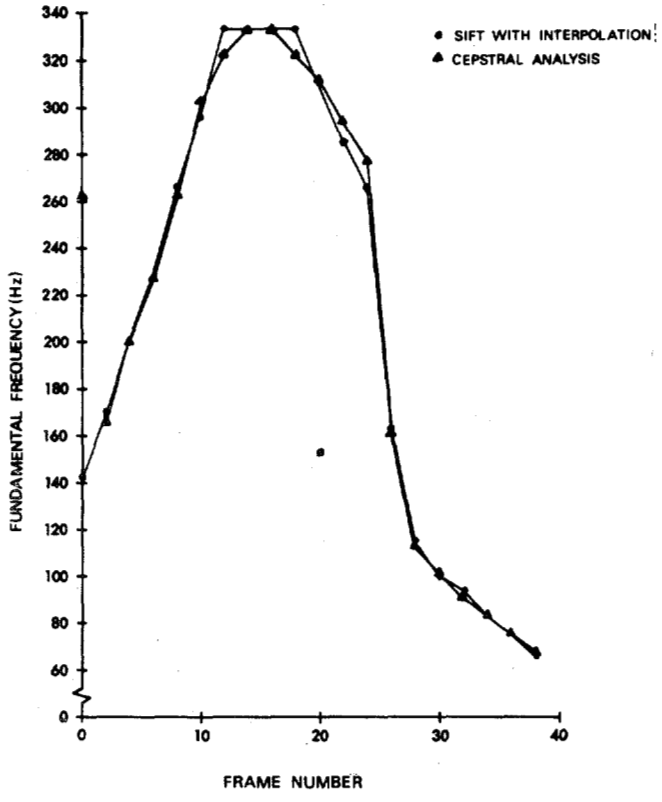

Fig. 13. F_0 analysis for the utterance "oh."

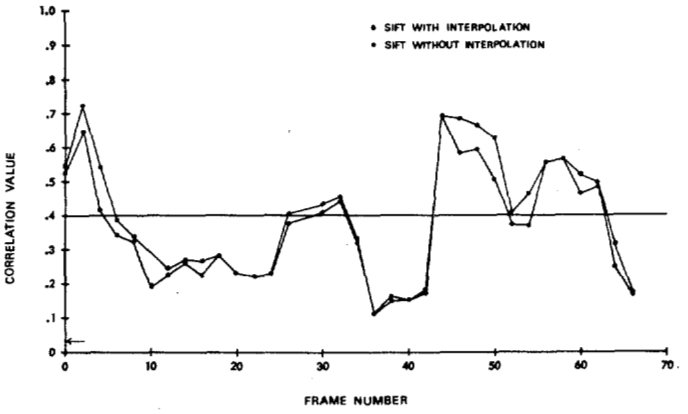

Fig. 14. Peak correlation values measured from SIFT for the phrase "put she can."

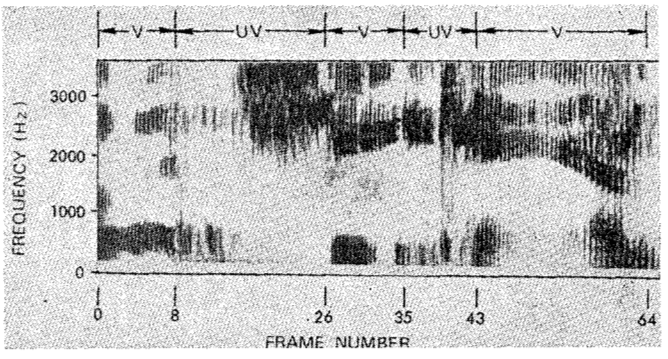

Fig. 15. Results of voiced-unvoiced decision from SIFT algorithm shown on spectrogram of utterance "put she can."

ally by inspection of the cepstral output data for each frame. If a reasonably sharp peak occurred with respect to the noise level, the segment was defined as voiced with $F_0 = 1/P$, where P is the location of the maximum peak in the range (2, 16) ms. Otherwise, the segment was defined as unvoiced and P was not used.

The fundamental frequency and voiced-unvoiced decision with the SIFT algorithm were obtained automatically. First, it is of interest to observe the character of the normalized peak amplitude within each frame as a function of the frame number. The peak uninterpolated and interpolated values versus even frame numbers are shown in Fig. 14. Based upon the previously discussed model, the peak correlation value excluding the origin would be expected to vary from around 0.30 to 0.70, depending upon the number of pitch periods per window and the variation between pitch period values within the window.

For unvoiced speech, the correlation between samples is small, with the actual value depending upon the number of samples analyzed. The peak correlation (excluding the origin) for unvoiced speech is usually less than 0.35 if $N \geq 60$. In fact, within a 99.9 percent confidence interval the peak amplitude in an unvoiced portion will be less than 0.378. Thus, with good theoretical justification it is possible to use a fixed threshold to make voiced-unvoiced decisions, using the SIFT algorithm. As a conservative value, 0.4 is chosen as the threshold. Although this value effectively guarantees that an unvoiced segment will not be defined as voiced, it does not guarantee that voiced segments will not occasionally be mistaken for unvoiced segments. Detection of these conditions and their correction is accomplished automatically by the flow chart presented previously in Fig. 12. For visual comparison, the automatically estimated boundaries are shown on a spectrogram of the utterance in Fig. 15. Note that the actual phrase analyzed has nearly all of the initial plosive /p/ missing and thus the analysis defines the beginning as voiced. All of these boundaries are within ± 1 frame (16 ms) of those estimated manually from cepstral analysis output data. Representative time series from which the voiced-unvoiced decisions and fundamental frequency estimates were obtained are compared along with the input data in Fig. 16. Every other frame of data, from frame 22 through frame 46 is shown. The leftmost series is the cepstral output data from 2 to 16 ms. A single normalization factor was applied to all frames of data for plotting purposes. The middle series is the output data from the SIFT algorithm just before the peak peaking and interpolation. On the right is shown the corresponding frames of input data from which the cepstral and SIFT algorithm results were obtained.

By inspection of the input data, it can be seen that frames 22–24 are unvoiced while voicing is beginning in frame 26. Frames 28–32 are clearly voiced, while frame

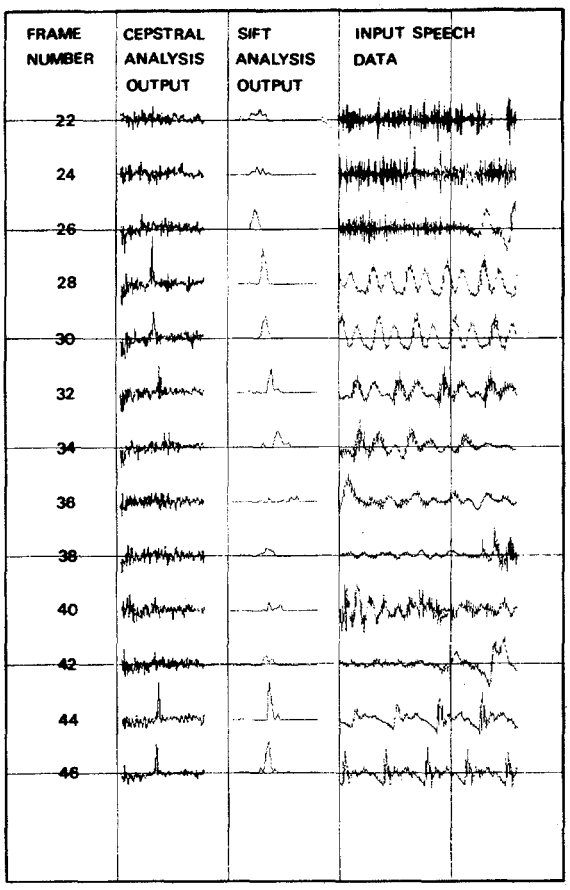

Fig. 16. Input and output waveforms for the cepstral and SIFT algorithm.

34 shows a voiced segment ending. Frames 36–42 are clearly unvoiced, while frames 44–46 are clearly voiced.

The voiced-unvoiced decisions can also be easily made by visual inspection of the cepstral data. Frames 28–32 and 44–46 are voiced. All other frames are unvoiced.

The SIFT results shown in these frames correspond to the data calculated within the tracking window of length 8 ms with initial center location chosen as 6 ms. At frame 26, the algorithm starts tracking the voiced portion. Tracking is continued with the center location of the window equal to the previous pitch period estimate until frame 36 at which time an unvoiced decision is made and the window is shifted back down to the initial location for frame 38. Only positive correlation values can be candidates for pitch peaks and thus negative values were set to zero for plotting purposes. Frames 28–34 and 44–46 were automatically defined as voiced. It is interesting to note that for these frames of data the relative values of the peak amplitudes in both the cepstral and SIFT output are quite similar.

Referring to Fig. 14, one additional advantage of performing interpolation can be seen. For the regions in which voicing occurs, interpolation significantly increases the peak values, while for unvoiced portions interpolation makes very little difference. Thus, the effective desired peak to undesired signal ratio is increased, allowing for somewhat easier discrimination between voiced and unvoiced sounds.

In Fig. 17, the estimated fundamental frequency versus even frame numbers has been presented for several situations. The triangles denote cepstral analysis results (resolution of 0.1 ms) which will be considered as reference values. The circles indicate F_0 estimates from SIFT without interpolation (that is, estimates with a resolution of 0.5 ms) and the solid dots indicate F_0 estimates with interpolation (corresponding to a resolution of 0.125 ms).

The results just shown are extremely close for each voiced segment. Note that the interpolated values do generally lie much closer to the cepstral analysis estimates than noninterpolated estimates.

IV. Summary

A new algorithm for efficient accurate automatic extraction of fundamental frequency from speech has been developed. Experimental results have been presented to demonstrate the accuracy of the SIFT algorithm with respect to the widely used cepstral analysis method.

Rather difficult analysis examples were purposely chosen to illustrate both the capabilities and limitations of the method. The algorithm does not guarantee error-free analysis; in one example, a gross error was shown. Also, it was shown that in practice, a few more tests in addition to a simple "yes-no" threshold decision are necessary to determine whether a segment is voiced or unvoiced.

Furthermore, it has been experimentally demonstrated that the difficult problem of detecting voicing during the transition from a voiced to unvoiced interval is not completely resolved. An illustration of this problem can be seen from close inspection of the vicinity of frame 35 in Fig. 15. (It should be pointed out, however, that whenever the SIFT algorithm failed to extract correct voicing, cepstral analysis also failed.)

Even with these possible limitations, the SIFT algorithm is believed to be a very worthwhile approach for consideration as a fundamental frequency extraction technique in automated digital speech analysis systems for the following reasons: 1) the unvoiced-voiced decision algorithm is quite simple; 2) implementation requires only the four elementary arithmetic operations, without any table lookups or complex indexing (such as necessary for FFT implementations); 3) the algorithm is very efficient computationally; and 4) informal listening tests of synthetic speech show no significant perceptual differences when cepstral analysis and the SIFT analysis F_0 contours were compared.

The SIFT algorithm is conservatively an order of

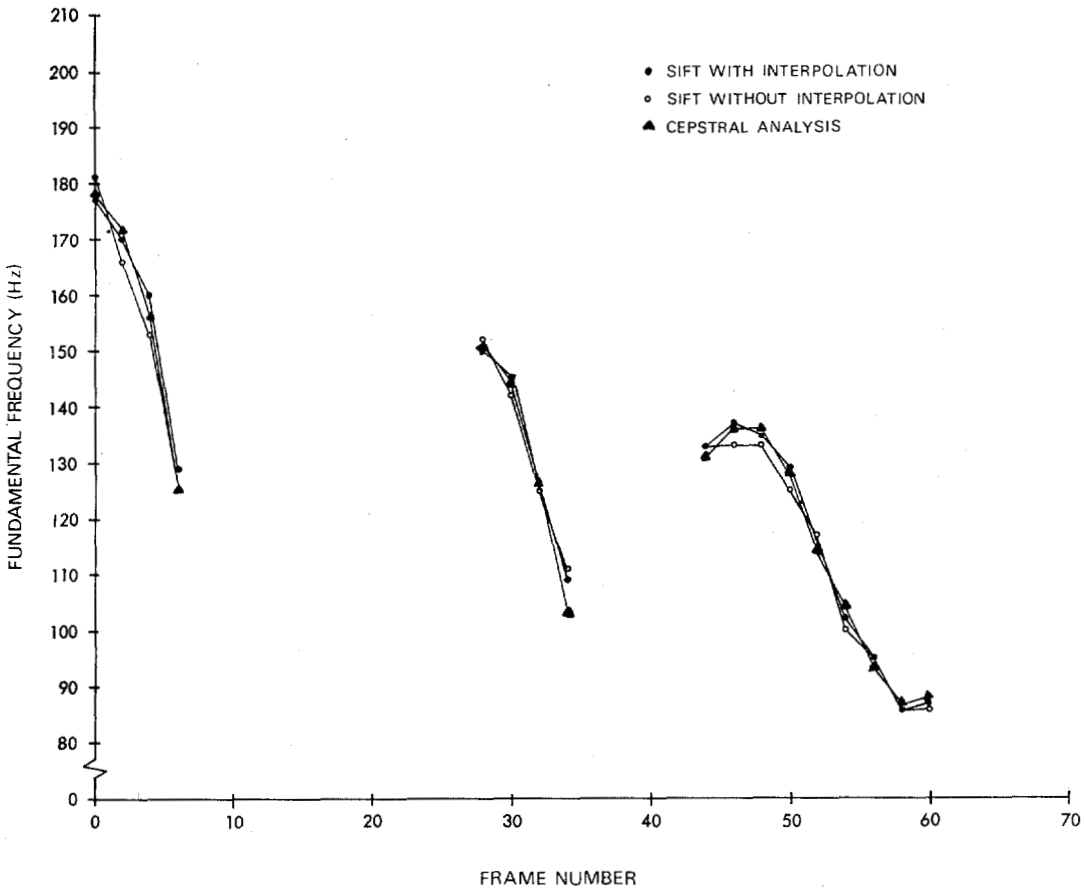

Fig. 17. Estimated fundamental frequency curves versus frame number for utterance "put she can."

magnitude faster than cepstral analysis, if prefiltering is done with an analog filter. It was demonstrated that the filter response characteristics are generally not critical since all results presented here were accomplished by digitally simulating a low-order Chebyshev filter. Even with general purpose minicomputers it should be possible to accurately analyze fundamental frequency contours within ten or twenty times real time using the SIFT algorithm. Presently techniques are being considered for implementing a real-time hardware version of the SIFT algorithm.

Acknowledgment

The author wishes to thank Dr. D. Broad for several discussions and many suggestions. He is also grateful for the careful reading and constructive criticisms of the reviewer.

References

[1] J. D. Markel, "Automatic formant and fundamental frequency extraction from a digital inverse filter formulation," in *Conf. Rec. 1972 Int. Conf. Speech Commun. and Processing*, 1972, Paper B9, pp. 81–84.
[2] M. R. Schroeder, "Vocoders: Analysis and synthesis of speech," *Proc. IEEE*, vol. 54, pp. 720–734, May 1966.
[3] A. M. Noll, "Cepstrum pitch determination," *J. Acoust. Soc. Amer.*, vol. 41, pp. 293–309, Feb. 1967.
[4] M. R. Schroeder, "Parameter estimation in speech: A lesson in unorthodoxy," *Proc. IEEE*, vol. 58, pp. 707–712, May 1970.
[5] A. M. Noll, "Short-time spectrum and 'cepstrum' techniques for vocal-pitch detection," *J. Acoust. Soc. Amer.*, vol. 36, pp. 296–302, Feb. 1964.
[6] J. D. Markel, "A linear least-squares inverse filter formulation for formant trajectory estimation," Speech Commun. Res. Lab., Santa Barbara, Calif., SCRL Monograph 7, Aug. 1971.
[7] N. Levinson, "The Wiener RMS (root-mean square error) criterion in filter design and prediction," Appendix in N. Wiener, *Extrapolation and Smoothing of Stationary Times Series*, Cambridge, Mass.: M.I.T. Press, 1966, pp. 129–148.
[8] G. C. M. Fant, *Acoustic Theory of Speech Production*. 's-Gravenhage: Mouton, 1960, pp. 42–46.
[9] R. W. Schafer and L. R. Rabiner, "System for automatic analysis of voiced speech," *J. Acoust. Soc. Amer.*, vol. 47, pp. 634–648, Feb. 1970.