Applications of a Nonlinear Smoothing Algorithm to Speech Processing

LAWRENCE R. RABINER, SENIOR MEMBER, IEEE, MARVIN R. SAMBUR, AND CAROLYN E. SCHMIDT

***Abstract*—In this paper a nonlinear smoothing algorithm recently proposed by Tukey is described and evaluated for speech processing applications. Simple linear smoothing routines generally fail to provide adequate smoothing for data which exhibit both local roughness and sharp discontinuities. The proposed nonlinear smoothing algorithm can effectively smooth such data by using a combination of median smoothing routines and linear filtering. The concept of double smoothing is introduced as a refinement on the smoothing algorithm. Examples of the application of the nonlinear smoothing methods to typical speech parameters are included in this paper.**

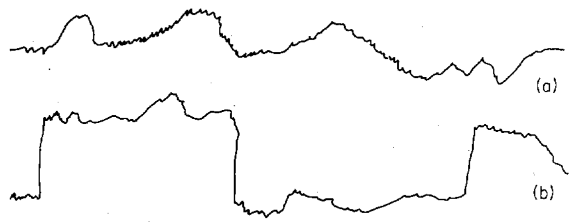

Fig. 1. Two examples of noisy signals to be smoothed.

I. Introduction

In most signal processing applications the concept of a linear smoother or equivalently a linear filter is a basic one. Linear smoothers are generally used in such systems because of their inherently desirable properties which include: 1) linear smoothers obey a superposition principle and 2) linear smoothers are time or shift invariant. Because of these properties, the theory of linear time-invariant systems has become well developed in the digital signal processing literature [1]-[3]. For some applications, however, linear smoothers are not completely adequate due to the nature of the data being smoothed. Fig. 1 shows two examples of data sequences which are to be smoothed. In the first example, a fairly slowly varying waveform has been corrupted by a high-frequency noise component. For this case a linear smoother (or low-pass filter) is entirely adequate for filtering out the noise. In the second example, however, although there is a noise-like component superimposed on the signal, the signal displays noticeable sharp discontinuities. Such discontinuities contain much high-frequency energy, and are essentially indistinguishable from the noisy component, as far as their spectral content. A linear smoother would therefore smear out the sharp changes in the data as well as filter out the noise. Such smearing of the data is unacceptable in many applications.

For cases such as the one shown at the bottom of Fig. 1, one must contemplate using some type of nonlinear smoothing algorithm which is capable of preserving sharp discontinuities in the data and still able to filter out noise superimposed on the data. Although such an ideal nonlinear smoothing algorithm does not as yet exist, a method recently proposed by J. W. Tukey [4], [5] can be shown to have approximately the desired properties. This algorithm is based on using a combination of running medians and linear smoothing. It is the purpose of this paper to discuss a generalized version of the Tukey smoothing method and to show how it is applicable to smoothing some typical speech parameters for applications to speech recognition, speech synthesis, and pitch detection.

The organization of this paper is as follows. In Section II we present the generalized nonlinear smoothing algorithm, discuss its properties, and show it can be practically realized. In Section III we show some typical examples of how the method has been used to smooth speech parameters. Included in this section are several comparisons between linear and nonlinear smoothers operating on the same data. Finally, in Section IV we discuss the computational considerations in using this algorithm.

Manuscript received January 9, 1975; revised May 30, 1975.
The authors are with Bell Laboratories, Murray Hill, N.J. 07974.

II. Nonlinear Smoothing Algorithm

The basic concept of a linear smoother is the separation of signals based on their nonoverlapping frequency content. For nonlinear smoothers it is more convenient to consider separating signals based on whether they can be considered smooth or rough (noise-like). Thus a signal $x(n)$ can be considered as

$$x(n) = S[x(n)] + R[x(n)] \tag{1}$$

where $S[y]$ is the smooth part of the signal y and $R[y]$ is the rough part of the signal y. The candidate proposed by Tukey for extracting $S[x(n)]$ from $x(n)$ was to use running medians of the data. Running medians have several good properties which make them good candidates for a smoother. These include the following properties.

Property 1: Median $[\alpha x(n)] = \alpha$ median $[x(n)]$.

Property 2: Medians will not smear out sharp discontinuities in the data, as long as the duration of the discontinuity exceeds some critical duration.

Property 3: Medians will approximately follow polynomials.

Property 1 is important in that scaling the input data leads to a scaling of the smoothed output data. It is emphasized that since medians are nonlinear, the superposition property

$$\text{median}\,[\alpha x_1(n) + \beta x_2(n)] = \alpha\,\text{median}\,[x_1(n)] + \beta\,\text{median}\,[x_2(n)] \tag{2}$$

does not hold. Property 2 is one of the key properties for median smoothers. Fig. 2 illustrates this property for a simple example. The input $x(n)$ exhibits sharp discontinuities at $n = 6$ and $n = 11$. The output $y(n)$ is defined as the 3-point median of $x(n-1)$, $x(n)$, and $x(n+1)$, i.e., the middle value when these three inputs are ordered in value. Neglecting, for the time being, how the outputs $y(0)$ and $y(15)$ are computed (the problem of initial and final conditions comes in here), the output $y(n)$ is seen to follow the input $x(n)$ exactly. If the 3-point median box in Fig. 2 were replaced by a 5-, 7-, or even a 9-point median, the output $y(n)$ would remain exactly as shown in Fig. 2. However if a median greater than 9 were used, the discontinuity would be smoothed out and $y(n)$ would be flat. Thus the size of running medians which can be used is strictly dependent on the minimum duration of discontinuity which the user wishes to preserve. The feature is especially of value for speech processing applications where, because of measurement and/or processing errors, the data often will exhibit single- or double-point sharp discontinuities. It is readily seen that running medians of 3's and/or 5's will eliminate such error discontinuities, but *will preserve* longer duration discontinuities.

The third property of median smoothers, i.e., their ability to approximately follow low-order polynomial trends in the data, is readily seen by considering the effects of various median smoothing routines on low-order polynomials. Fig. 3 shows a quadratic, a cubic, and a quartic polynomial and the results of smoothing these inputs by running medians of 3, 5, and 7. It is seen in this figure that 3-point medians essentially follow all these polynomials, whereas 7-point medians have smoothed out the quartic polynomial considerably. Thus the general trend is the longer the running median, the more it smoothes out lower order polynomials.

Although median smoothing preserves sharp discontinuities in the data, it fails to provide sufficient smoothing of the undesirable noise-like components for which the smoothing was originally designed. A fairly good solution is a smoothing algorithm based on a combination of running medians and linear smoothing. Since the running medians provide a fair amount of smoothing already, the linear smoothing can consist of a fairly low-order system and still give adequate results. Tukey proposed the use of a 3-point Hanning window as one candidate for the linear smoother.

Fig. 4(a) shows a block diagram of the simple smoothing algorithm. The output $y(n)$ is an approximation to the signal $S[x(n)]$. To the extent that the approximation is not ideal, one can consider the use of a "double smoothing" routine as shown in Fig. 4(b). Since

$$y(n) \approx S[x(n)] \tag{3}$$

then

$$z(n) = x(n) - y(n) \approx R[x(n)]. \tag{4}$$

Thus additional smoothing of $z(n)$ yields a correction term which is added back to $y(n)$ to give $w(n)$, the second approximation to $S[x(n)]$. Thus $w(n)$ satisfies the relation

$$w(n) \approx S[x(n)] + S[R[x(n)]]. \tag{5}$$

If $z(n) = R[x(n)]$, i.e., the smoother were ideal, then $v(n)$, the output of the second smoother, would be identically zero, and the second-order correction would be unnecessary.

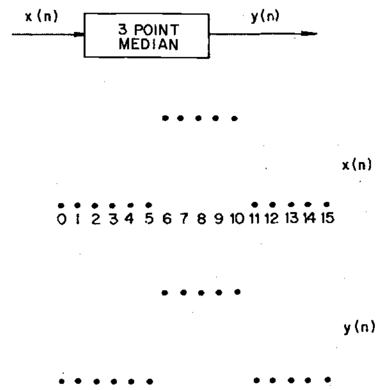

Fig. 2. Median smoothing of a sequence with a discontinuity.

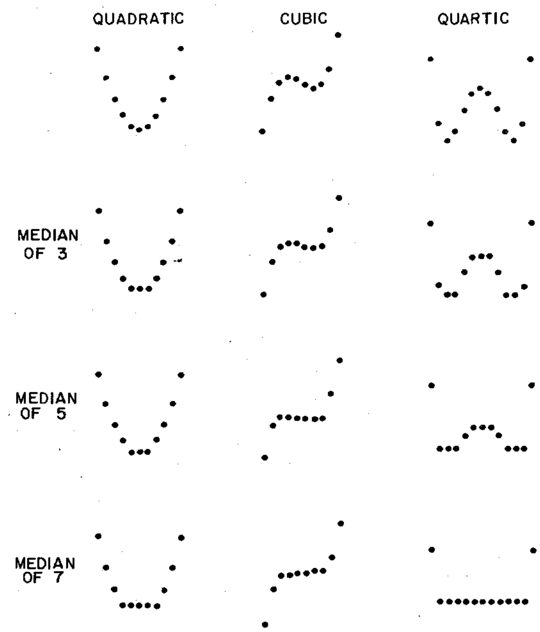

Fig. 3. Effects of various median smoothers on low-order polynomials.

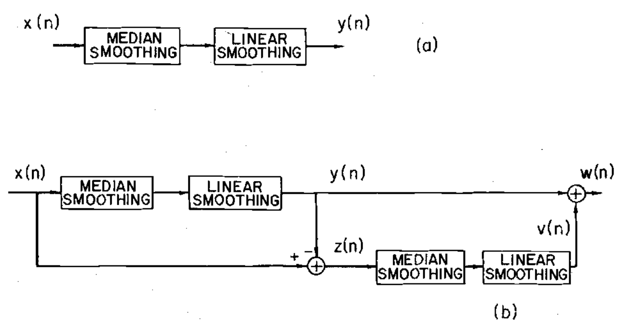

Fig. 4. Block diagrams of smoother and double-smoothing algorithms.

In order to implement the system shown in Fig. 4(b), one must take care to account for the delays in each path of the smoother. The median smoother has a delay associated with the size median used, and the linear smoother has a delay associated with the impulse response used. For example, a median of 5 routine has a delay of 2 samples, and a 3-point Hanning window has a delay of 1 sample. Thus, the total delay of the first smoother is 3 samples. Fig. 5 shows a block diagram of the overall smoother incorporating the appropriate delays in each path.

The only remaining task required to implement the system of Fig. 5 is to provide an algorithm for handling the endpoints of the data. Several techniques for generating the set of additional initial and final values (i.e., those outside the interval in which the data are defined) were investigated experimentally, including constant, linear, and quadratic extrapolation. For most of the applications to be discussed here, constant extrapolation from the initial or final data point proved to be entirely adequate.

Fig. 6 shows a comparison between several alternative smoothing algorithms for an artificially created test input sequence. Fig. 6(a) shows the input sequence, and Fig. 6(b)–(d) show the outputs of a linear smoother (a 19-point finite impulse response (FIR) low-pass filter), a combination of median and linear smoother (a running 5 median and a 3-point Hanning window), and a median of 5 smoother, respectively. The smearing effects of the linear smoother at each input discontinuity are clearly in evidence in this figure, whereas the median smoother alone essentially preserves the data exactly. The combination of median and linear smoothing is seen to provide a good compromise between the median and linear smoothers in this example. Fig. 7 shows the effects of adding broad-band noise to the input of Fig. 6. In this case, the median smoother is inadequate for filtering out the broad-band noise on the input, thereby producing a rough output sequence, as shown in Fig. 7(d). The linear smoother does an excellent job of filtering out the noise, as expected, and the output shown in Fig. 7(b) is almost identical to the output in Fig. 6(b) when there was no additive noise. Finally, the combination smoother is seen to again be a good compromise between the linear and the median smoothers. As seen in Fig. 7(c), the noise is smoothed a great deal, and the discontinuities in the input are fairly well preserved.

In summary, a smoothing algorithm consisting of a combination of running medians and linear smoothing appears to be a reasonable candidate for smoothing noisy sequences with discontinuities. In the next section we present several examples of how this combination smoothing algorithm has been applied to speech processing problems.

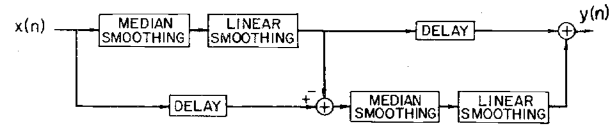

Fig. 5. Realizable block diagram of overall smoothing algorithm.

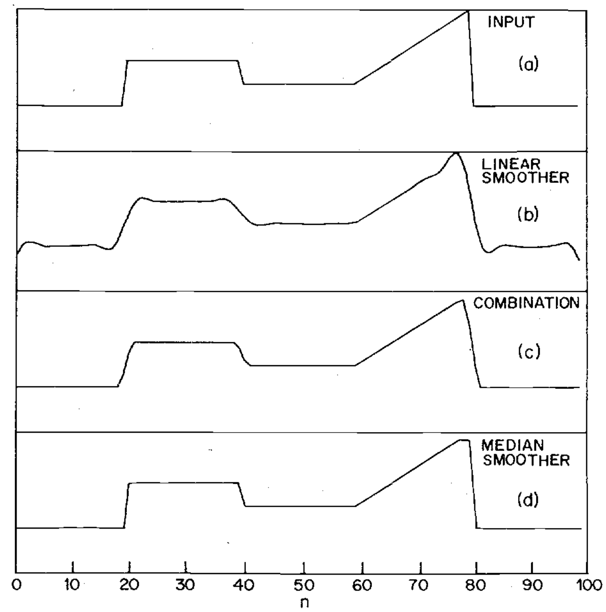

Fig. 6. Examples of several smoothed outputs for a simple test input.

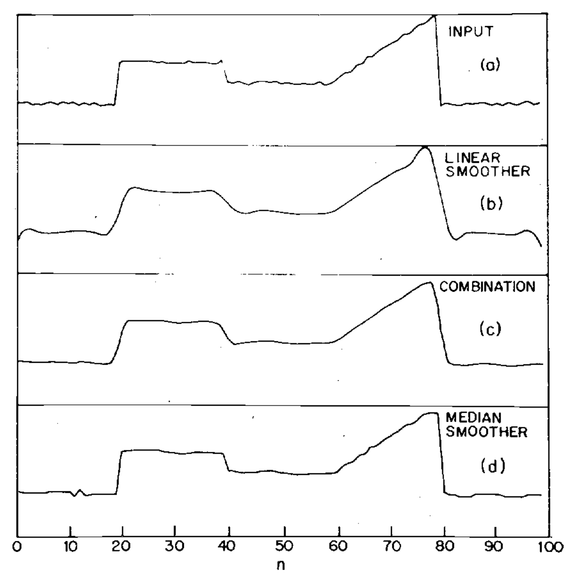

Fig. 7. Effects of additive noise on several smoothers.

III. Applications to Speech Processing

One of the most promising applications of the smoothing algorithm discussed in the previous section is in the areas of speech processing. In particular, we have applied this technique in investigations in the areas of speech recognition, speech synthesis, and pitch detection. In this section we show several examples of representative input sequences from each of these areas and the resulting outputs from the smoothing algorithm.

Figs. 8–10 show plots of input sequence from speech recognition work [6], the linearly smoothed outputs, the outputs from the combination of medians and linear smoothing, and the outputs from a median routine alone. For these examples the linear smoother was the same one used in the examples of Figs. 6 and 7. The all-median smoother was implemented as shown in Fig. 4 without the linear smoothing in each of the forward paths. The median smoothing routine consisted of a running median of 3 followed by a running median of 5 in both smoothing paths. The combination smoother was identical to the median smoother with the inclusion of a 3-point Hanning window as the linear smoother in both smoothing paths.

In Fig. 8 the input is the measured energy of the speech utterance /919/. The energy was computed 100 times/s using a 10 ms averaging window. The roughness in the data is due to the fixed averaging time of the measurement, which is sufficiently short that it interacts with the pitch period (which

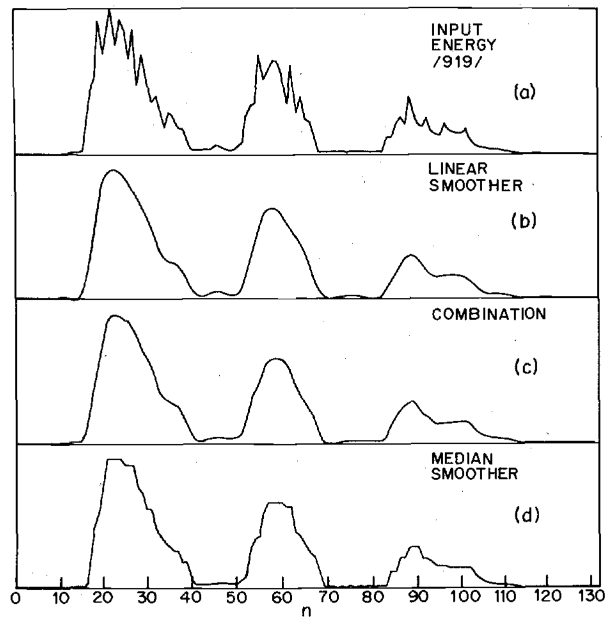

Fig. 8. Effects of linear, combination, and median smoothers on speech input energy for the sequence /919/.

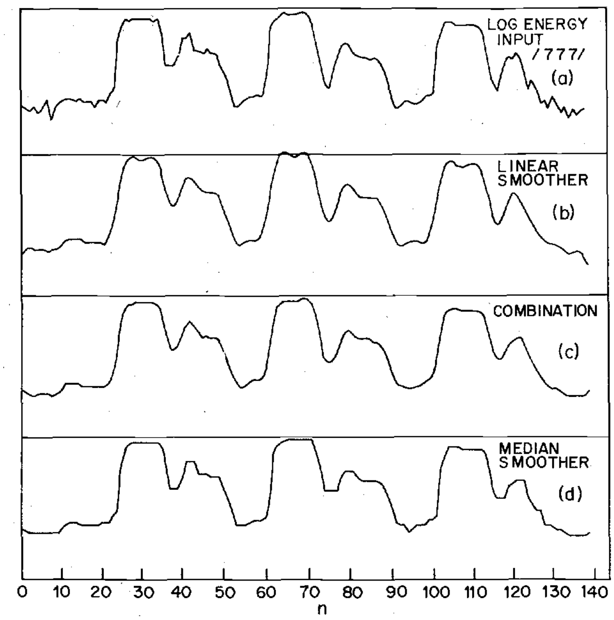

Fig. 10. Effects of linear, combination, and median smoothers on speech log energy for the sequence /777/.

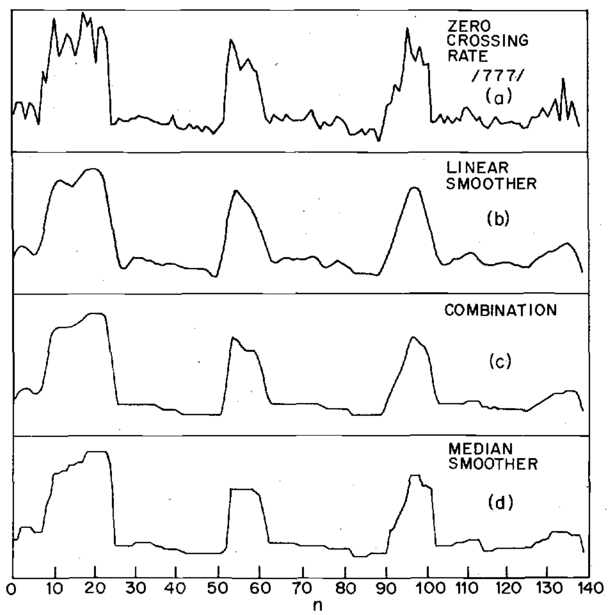

Fig. 9. Effects of linear, combination, and median smoothers on speech zero-crossing rate for the sequence /777/.

varies from about 5 to 20 ms for male speakers). Thus, depending on the placement of the window and the fraction of pitch periods within the window, the measurement of energy will fluctuate as shown. As seen in Fig. 8(d), the output of the median smoother has a block-like effect due to the lack of any linear smoothing, i.e. the high-frequency components in the data are readily seen in the regions where one desires them to be smoother. Comparing Fig. 8(b) and (c), it is seen that the combination smoother provides sufficient smoothing of the noisy component of the input; yet, it still provides better approximations to the sudden changes in amplitude of the data than does the linear smoother.

Figs. 9 and 10 show similar results on the smoothing of two of the other parameters used in the digit recognition work—namely zero-crossing rate and log energy of the speech waveform. Fig. 9 shows the measured zero-crossing rate for the utterance /777/ and the three possible smoothings of these data. Fig. 10 shows the measured log energy for this same utterance and the three smoothed data sequences. In these, and similar examples, the combination smoothing routine afforded the best tradeoffs between the desired linear smoothing to eliminate roughness inherent in the measurements and the desired nonlinear smoothing to preserve sharp changes in the data themselves.

Fig. 11 shows the effect of several versions of the smoothing algorithm on intensity data used in speech synthesis experiments [7]. Indicated in each part of Fig. 11 is the size medians used (two numbers indicate use of two median sizes, i.e., 3, 5 means a running median of 3 followed by a running median of 5), an indication of whether linear smoothing was employed (and what it was if used), and finally an indication of whether the double smoothing was used. Contrasting Fig. 11(b) and (d), or Fig. 11(c) and (e), the additional smoothing obtained using higher order medians is clearly seen. Further, the differences between using median smoothing alone and the combination with linear smoothing are seen be contrasting Fig. 11(b) with (c), or Fig. 11(d) with (e).

Fig. 12 shows the results of smoothing the pitch period contour for synthesis. Informal listening tests on the synthetic speech indicate that the smoothed pitch contours are not detrimental in any way to the quality of the synthetic speech [8].

Finally, Figs. 13 and 14 illustrate the application of the smoothing algorithm to pitch contours in which the pitch detection algorithm made gross errors in estimating the pitch period [9]. Generally, such errors must be corrected before smoothing can be applied. One of the side benefits of median smoothing is that it can inherently correct isolated errors in the data (i.e., sharp discontinuities of short duration, e.g., one or two samples), at the same time combining this operation with the desired smoothing. Fig. 13 shows an example with a

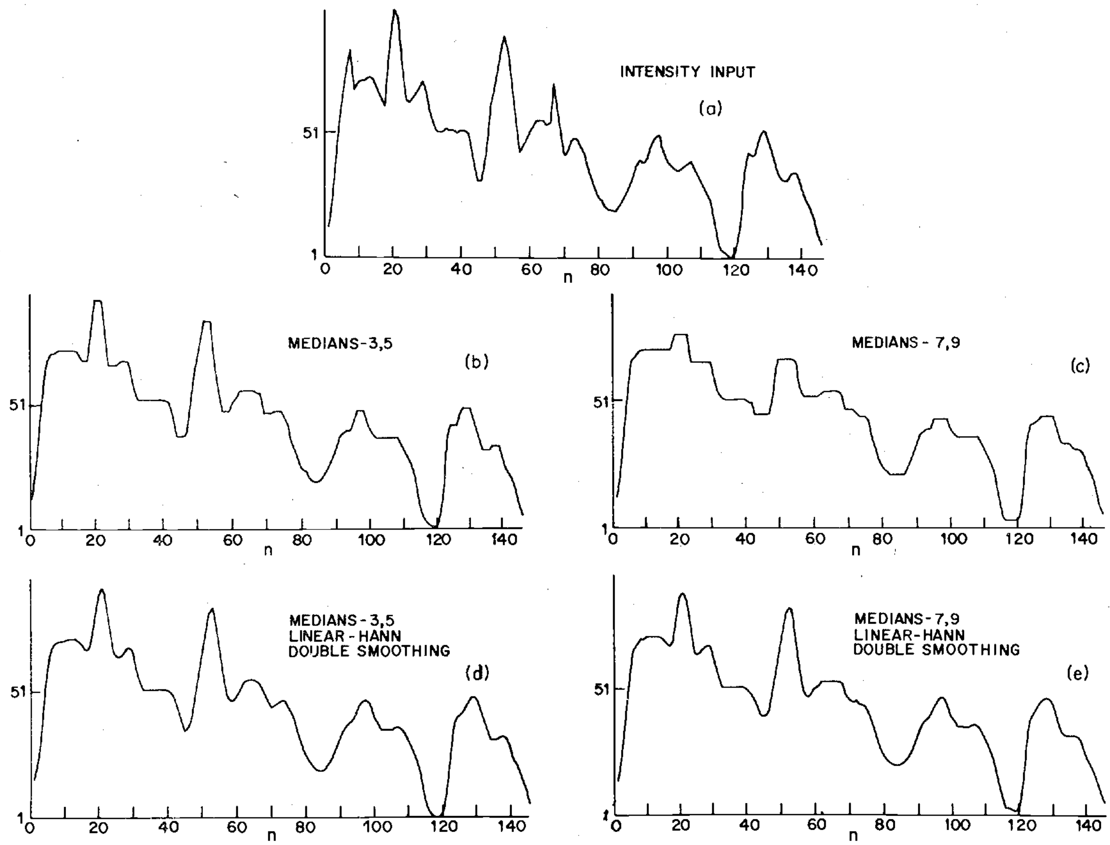

Fig. 11. Effects of several versions of the smoothing algorithm on a speech intensity contour.

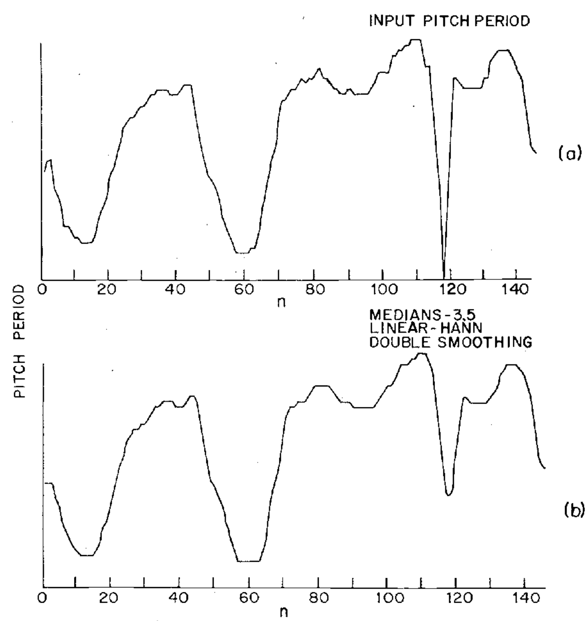

Fig. 12. Effects of the combination smoother on the pitch period contour for a synthesis experiment.

fairly large number of isolated gross measurement errors. Fig. 13(b) and (c) illustrate how median smoothing alone does an excellent job of eliminating these gross errors in measurement. In Fig. 13 (b) a median of 5 routine was used. In Fig. 13(c), a median of 3 routine was used followed by a median of 5 routine. The differences, although small, can be seen by comparing these figures. The combination of median 3 and median 5 smoothing is capable of correcting up to 3 isolated gross errors in a 5-point interval, whereas a median 5 smoother can only correct 2 gross errors in a 5-point interval. This is one reason we have used a combination median smoother in our examples in this paper. Fig. 13(d) shows the final smoothed, error-corrected pitch period output using the full smoother of Fig. 4. The overall pitch contour is a smoother version of the input sequence with no gross discontinuities.

As a final example, Fig. 14 shows another example of the application of the smoother to pitch period measurements with isolated gross errors in the data. Fig. 14(d) shows how the smoother is capable of following true discontinuities in pitch period and correcting isolated errors of measurement.

IV. Summary

The smoothing algorithm has been shown to be a reasonable candidate for smoothing data with some or all of the following characteristics.

1) The data have sharp discontinuities of reasonable duration.

2) The data have roughness due to noise in the measurement (the noise may be inherent in the measurement process itself being imperfect).

3) The data have sharp, isolated discontinuities of very short duration which are to be eliminated. Such discontinuities may be due to errors in transmitting the data, or to imperfect analysis procedures as in the speech examples of Figs. 13 and 14.

The discussion until now has concerned the applicability of the smoothing algorithm to speech processing problems. Another issue with any smoothing algorithm is the required computation to implement it. The smoothing algorithm of

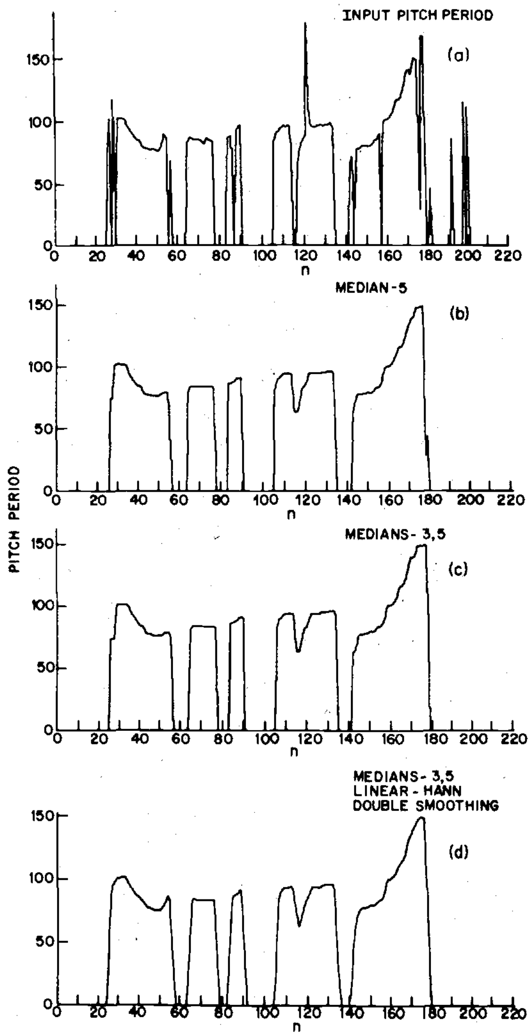

Fig. 13. Comparison of two median smoothers and the combination smoother for a pitch period contour with gross errors.

Fig. 4 has the virtue that it can be implemented in a simple and straightforward manner. The median smoothing routines involve simple sorting of numbers. For small size medians this means sorting groups of from 3 to 9 points. Efficient sorting procedures for this type of application have been described by Knuth [10]. The algorithm also requires the use of a linear smoother. The smoother used in all the examples here was a 3-point Hanning window with coefficients of $\frac{1}{4}, \frac{1}{2}, \frac{1}{4}$. Thus, the Hanning window can be implemented without the use of multipliers simply by shifting the data 1 or 2 places to the right and accumulating the results. Hence, computationally, the smoothing algorithm is quite efficient and competitive with other proposed smoothers.

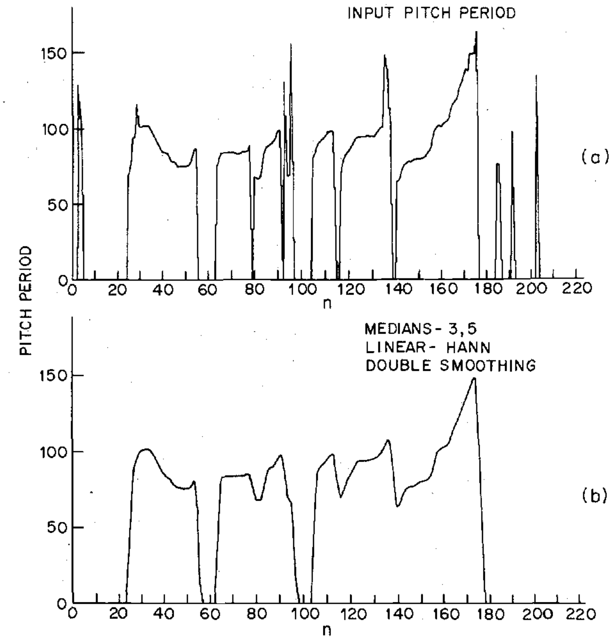

Fig. 14. Effects of the combination smoother on a second pitch period contour with gross errors in the data.

Acknowledgment

The authors would like to thank Dr. H. Helms of Bell Laboratories for bringing the Tukey algorithm to their attention and for his suggestions on its use.

References

[1] B. Gold and C. M. Rader, *Digital Processing of Signals.* New York: McGraw-Hill, 1969.
[2] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing.* Englewood Cliffs, N.J.: Prentice-Hall, 1975.
[3] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing.* Englewood Cliffs, N.J.: Prentice-Hall, 1975.
[4] J. W. Tukey, "Nonlinear (nonsuperposable) methods for smoothing data," in *Cong. Rec., 1974 EASCON*, p. 673.
[5] A. E. Beaton and J. W. Tukey, "The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data," *Technometrics*, vol. 16, pp. 147-185, May 1974.
[6] M. R. Sambur and L. R. Rabiner, "A speaker independent digital recognition system," *Bell Syst. Tech. J.*, vol. 54, pp. 81-102, Jan. 1975.
[7] R. W. Schafer and L. R. Rabiner, "System for automatic formant analysis of voiced speech," *J. Acoust. Soc. Amer.*, vol. 47, pp. 634-648, Feb. 1970.
[8] M. R. Sambur, "An efficient linear prediction vocoder," *Bell Syst. Tech. J.*, vol. 54, Dec. 1975.
[9] M. Cheng, "A comparitive performance study of several pitch detection algorithms," M.S. thesis, Dep. Elec. Eng., Mass. Inst. Technol., Cambridge, 1975.
[10] D. E. Knuth, *Sorting and Searching.* Reading, Mass.: Addison-Wesley, 1973.