

Generation and Combination of Grains for Music Synthesis

Author(s): Douglas L. Jones and Thomas W. Parks

Source: *Computer Music Journal*, Vol. 12, No. 2 (Summer, 1988), pp. 27-34

Published by: [MIT Press](#)

Stable URL: <http://www.jstor.org/stable/3679939>

Accessed: 18-11-2015 11:02 UTC

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at <http://www.jstor.org/page/info/about/policies/terms.jsp>

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.



MIT Press is collaborating with JSTOR to digitize, preserve and extend access to *Computer Music Journal*.

<http://www.jstor.org>

Douglas L. Jones¹

Lehrstuhl fuer Nachrichtentechnik
Universitaet Erlangen-Nuernberg
Cauerstrasse 7
8520 Erlangen
West Germany

Thomas W. Parks

Electrical Engineering
312 Phillips Hall
Cornell University
Ithaca, New York 14853 USA

Generation and Combination of Grains for Music Synthesis

Introduction

Granular synthesis has been used for a number of years as a sound synthesis technique. This method constructs an acoustic signal from a number of short-duration acoustic elements, called *grains*. These grains are produced either by extracting short segments of natural sounds such as speech, or by synthesis according to a mathematical description of the grains. Efficient methods of producing and combining these grains are currently especially interesting, because the recent development of special-purpose digital signal processing (DSP) microprocessors has made real-time granular synthesis possible at a reasonable hardware cost. This paper discusses the computational aspects of granular synthesis. Methods of extracting grains from natural signals and recombining them with phase alignment are described. Fast methods for digital generation of synthetic signals with linear frequency modulation (FM) are also presented.

Granulation and Synthesis from Natural Sounds

Granular music synthesis through modification and combination of natural sounds offers interesting possibilities in music synthesis. In actuality, such techniques have been applied to the problem of time-scale modification of speech signals for a number of years. Time-scale modification involves changing the time-rate of a signal without altering

the pitch. For this task, simply increasing the playback rate is unacceptable, because the frequencies are also accordingly increased. One solution is to decompose the signal into a long series of grains; these grains can be recombined with different spacing to alter the time-scale of the signal, while still preserving the frequency content of the signal in the individual grains.

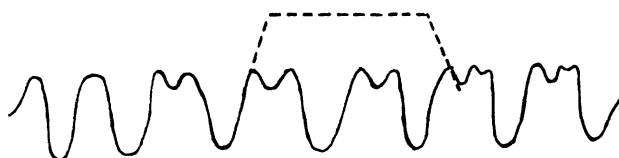
From a sampled digital signal, $s(m)$, which in principle extends for all time, the extraction of a grain of relatively short time duration is desired. The simplest way to do this is to multiply the signal sample-by-sample by a *window function* $w(m - n)$, which is centered at sample n and is nonzero only for the samples $w(-N/2 + 1)$ through $w(N/2)$. The resulting grain is also nonzero only for N samples, $n - N/2 + 1$ through $n + N/2$. Figure 1 depicts this operation. The window chosen is always a smooth, positive function in the time domain so that the grain resembles the signal over the nonzero interval. In the frequency domain, this windowing operation introduces a convolution with the window spectrum, resulting in a smoothing of the spectrum. This smoothing still preserves the basic frequency structure of the local section of the signal.

It is possible to decompose a signal into a series of grains and later exactly reconstruct the signal, if the windows are chosen so that their sum is exactly unity at every point, simply by summing the grains back together. However, once the signal has been decomposed, these grains can be treated as individual sound units and can be modified, moved around, and recombined as desired. It is now easy to clarify how time-scale modification can be accomplished. For compression by a factor of two, every other grain is thrown away, and remaining grains

1. On leave from the University of Illinois, Urbana/Champaign.

Fig. 1. Grain extraction from a signal.

(a) Signal and window



(b) Resulting grain



are shifted to earlier times to eliminate the gaps. The grains are then summed to produce a shorter-in-time signal. Figure 2 illustrates this procedure. The grains preserve the basic frequency content of the signal, so the pitch is not altered. It should be noted that a time-scaled signal could also be constructed using all the grains simply by constructing two time-scale modified signals as in Figure 2c, one from the even-numbered grains, and one from the odd-numbered grains, and then summing the two signals together. In effect, this merely results in a denser overlap of grains, with effectively two significant grains at every time location. Such a signal usually sounds almost identical to a signal constructed from every other grain.

Figure 2 masks a problem with this technique; that is, after the time-shift of the grains, the phases at the grain edges may not be identical. Summing these grains can produce destructive interference, resulting in beats. Figure 3 illustrates this problem. Overlapping window amplitudes that do not sum to unity can also produce beats. Such beats can, of course, at times be desired by the composer, but for rate modification or smoothly changing signals they are usually undesirable.

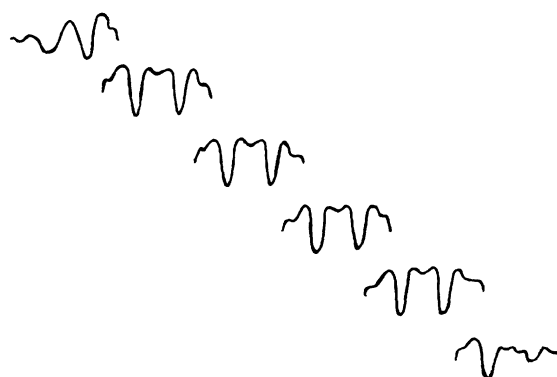
One solution to the problem of beats is to use windows of length such that the overlap is always

Fig. 2. Granular time-scale modification procedure.

(a) Signal



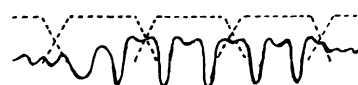
(b) Grains



(c) Time-shifted alternate grains



(d) Time-scaled signal

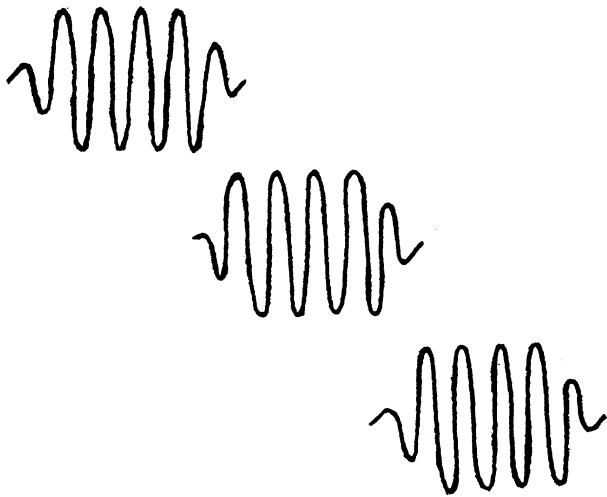


in phase; *time-domain harmonic scaling*, a time-scale modification technique developed by Malah (1979), uses this approach. This method has the disadvantage that the window length must vary with the fundamental frequency of the signal, and thus the window length cannot be freely chosen. Here is presented a different approach, first developed by Roucos and Wilgus (1985), that overcomes this disadvantage with time-domain harmonic scaling.

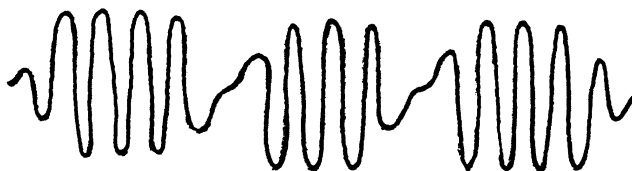
In-phase overlap of the grains can be achieved by adjusting the starting location of the windows so

Fig. 3. Reconstructed signal with out-of-phase grains.

(a) Grains



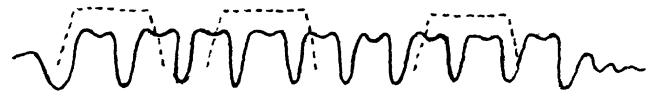
(b) Reconstructed signal



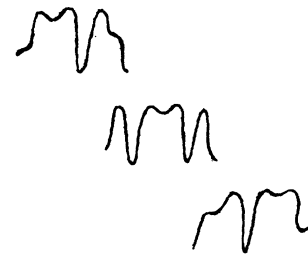
that the grains overlap in phase. Figure 4 illustrates this idea. Note that the spacing between the window starting locations is no longer equal; however, the length of the grains is now independent of the signal frequency. It remains only to determine the starting locations of the grains. Figure 5 illustrates the following method. First, a series of *nominal starting locations* L_i for the leading edges of the grains are selected, according to the desired effect. For example, to compress the time-scale by a factor of two, these would be evenly spaced locations. The spacing is a function of the window length such that when the grains are overlapped in the synthesis procedure, the signal is shortened in time by a factor of two. Next, a *search interval* of length N_s

Fig. 4. In-phase reconstruction by varying the grain-starting locations.

(a) Signal, and grain windows



(b) Time-shifted grains



(c) Reconstructed signal

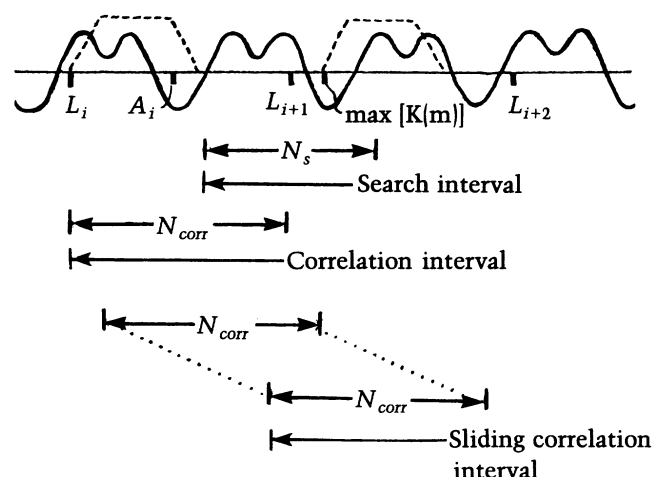


around the nominal starting location of the next grain is chosen; the algorithm searches over this interval for the best starting location. This search interval should be at least as long as the longest period expected in the signal, to assure that phase-alignment is possible. A *correlation interval* of length N_{corr} around the *overlap starting location* A_i on the old grain is selected, and this is correlated with a sliding correlation interval over the search interval. (Note that this is not strictly a true correlation, since the data points in the sliding correlation window change with the offset.) This correlation interval should be at least as long as the lowest expected pitch period. This operation can be written as

$$K(m) = \sum_{k=-\frac{N_{corr}}{2}+1}^{\frac{N_{corr}}{2}} x(A_i + k)x(L_{i+1} + m + k),$$

$$-\frac{N_s}{2} + 1 \leq m \leq \frac{N_s}{2}. \quad (1)$$

Fig. 5. Illustration of the phase-alignment technique.



The maximum correlation value $K(m)$ over the search interval indicates the location of best overlap between the two grains, and the new grain window is started at this point. The window for the first grain is simply started at the nominal starting location. An important question with this technique is whether it works with nonperiodic or noise-like signals. Roucos and Wilgus (1985) have shown theoretically that the alignment technique should not alter the characteristics of noise-like signals, and this is confirmed by our experimental work.

This approach is extremely flexible, allowing the window lengths, the nominal starting locations, and the number of grains to be overlapped to be entirely independent. Granular analysis/synthesis using this approach is thus a powerful technique for modifying natural sounds, allowing time-varying rate-changing and frequency variation. The combination or mixing of grains from different sources is easily accomplished using the same procedure between sections of two different signals, rather than different sections of the same signal, to determine the window starting locations. A problem may exist when the two grains have a different fundamental frequency; in this case, it may be best just to choose a nominal starting location with amplitude as close as possible to that at the overlap starting location, to minimize the discontinuity. This is actually simpler and faster than the correlation-based method described above.

Comments on this Technique

It is possible to realize a wide variety of effects using this technique. Time-scale expansion is accomplished by a similar process, with overlapping grains that are then spread out in time. With speech, rates of compression much greater than two are usually too fast to understand, and expansion by a factor of much more than two results in a robotic, excessively tonal quality. Of course, such effects are sometimes desired. In addition to time-scale modification, frequency shifting while maintaining the same time-scale can be obtained merely by time-scaling the signal and then altering the playback rate to achieve the same time-scale, thus producing a corresponding frequency shift. It is also possible to apply this technique on different frequency bands, creating several overlapping grains from one signal segment with different frequency content, merely by bandpass filtering the signal before windowing. These grains could then be selectively discarded to obtain the "evaporative" effect described by Roads (1985). For this processing, the phase alignment need be computed only once; the same starting location can be used with all of the grains in different frequency bands.

With time-scale modification of speech, grain durations of much less than 20 msec result in significantly reduced perceived quality in the modified speech, since effects due to the grain splicing become audible. Unless these effects are desired, 20 msec represents a lower limit for grain length for granular analysis/synthesis with speech. The lack of phase synchronization for grains of duration much less than 35 msec results in significantly reduced perceived quality with speech; for longer grains, phase synchronization is not as important and can perhaps be omitted if it is necessary to reduce the computational load. A smooth transition between the grains is necessary even when phase synchronization is present, so some tapering of the window edges is required. The actual shape of the window does not appear to be very important; very good results are obtained with triangular windows. In experiments with overlapping up to four grains simultaneously from the same signal, we find the sound quality to be slightly muddier than when

only two grains were overlapped. When the grains come from a natural source with a rich acoustic content, it appears unnecessary and perhaps undesirable to combine more than two grains at any particular time to produce the effect of natural-sounding but modified sound.

Granular analysis and synthesis from natural sounds can be accomplished with the current generation of DSP microprocessors, which support several million mathematical operations per second. The computation required to construct and recombine the grains is relatively small: the window values must be obtained and multiplied by the signal, and the active grains must be added together. For only a few active grains, these operations require a small fraction of the available processor cycles. The alignment process is expensive computationally because of the correlation, but the computational requirements can be greatly reduced by lowpass filtering the data to retain only the fundamental frequency and then decimating this new signal before correlating it. This new signal is used only to determine the alignment; the grains are extracted from the original signal. We have implemented a real-time time-scale modification system with one TMS32010 microprocessor, but this algorithm requires most of the computational power of this chip. This should give the reader a rough baseline from which to estimate the amount of hardware required for a real-time natural-sound granular analyzer/synthesizer.

Fast Generation of Frequency-Modulated Grains

Although granular synthesis from natural sounds is a powerful technique for granular music synthesis, to date most of the work in granular music synthesis has been with artificially generated grains. These grains have so far consisted of single-frequency oscillations with a time envelope, or window. The possibility of increasing the palette of available synthetic grain types is of great interest, but new grain types must be both acoustically interesting and simple to compute, since synthetic granular synthesis requires the production of up to several thousand grains per second. In particular, grains with linear frequency modulation may be of

interest. Roads (1985) has composed music with clouds of single-frequency grains of linearly increasing mean frequency; frequency modulation in the grains as well introduces new possibilities for such compositions. This section reviews a technique for generating a grain with a linear frequency modulation and a Gaussian envelope requiring only two complex multiplications per sample (Jones and Parks 1987). This method can be extended to compute exponentials of complex polynomials of any order, with a computational cost of one complex multiply per degree of the polynomial (Nuttall 1987). This technique allows an extraordinary variety of frequency-modulated grains to be generated with little more computation than required for digital sinusoid generation.

The complex Gaussian function

$$f_n = f(n) = \exp(C_0 + C_1 n + C_2 n^2), \quad (2)$$

where C_0 , C_1 , and C_2 are complex-valued, can be described as a modulation with a linearly increasing (or decreasing) frequency and a Gaussian envelope. Figure 6 shows the real part of one example of this function. This function can equivalently be written in terms of more intuitive, real-valued parameters as

$$f_n = A \exp \left[j\theta - j\omega_0(t_0 + n\tau) - \left(a + j\frac{r}{2} \right) (t_0 + n\tau)^2 \right], \quad (3)$$

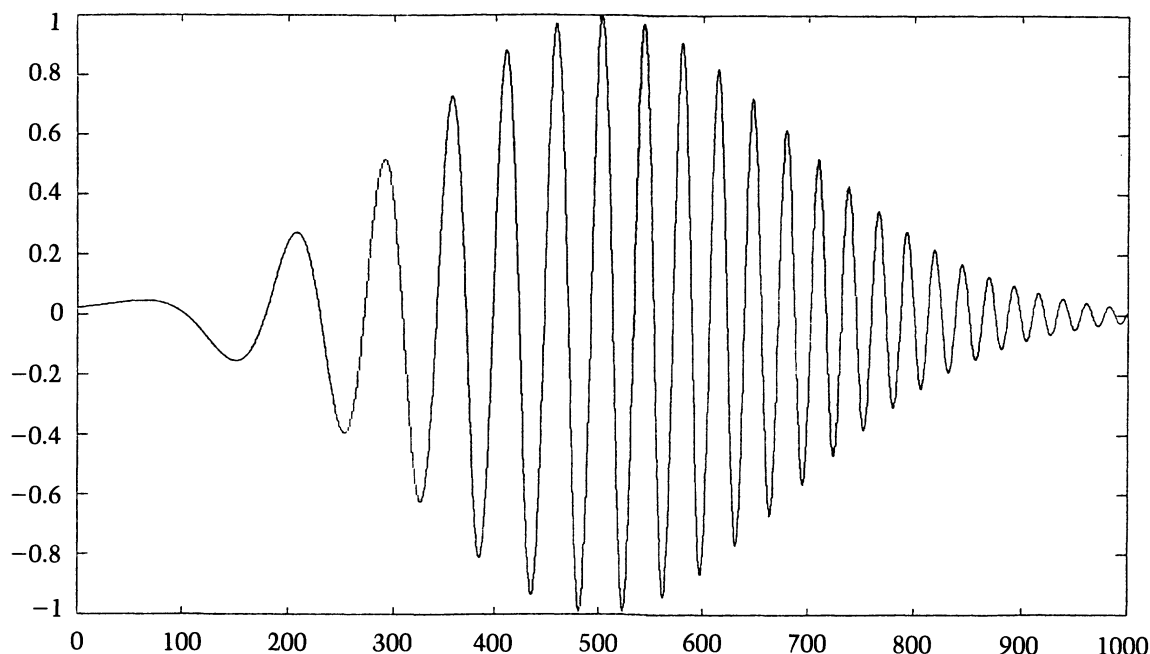
where A is the peak amplitude, θ is the phase at the peak, t_0 is the starting time value, τ is the sampling interval, ω_0 is the center frequency, a determines the time width of the Gaussian, and r is the chirp rate (linear frequency modulation rate) of the Gaussian function. The relationship between these parameters and the complex parameters C_i is as follows:

$$C_0 = \ln(A) - at_0^2 + j\left(\theta - \frac{r}{2}t_0^2 - \omega_0 t_0\right) \quad (4)$$

$$C_1 = -2art_0 - j\tau(rt_0 + \omega_0) \quad (5)$$

$$C_2 = -\left(a + j\frac{r}{2}\right)\tau^2, \quad (6)$$

Fig. 6. A typical Gaussian frequency-modulated grain.



or

$$a = \frac{-\text{Re}[C_2]}{\tau^2} \quad (7)$$

$$r = \frac{-2\text{Im}[C_2]}{\tau^2} \quad (8)$$

$$t_0 = \frac{-\text{Re}[C_1]}{2\tau a} \quad (9)$$

$$\omega_0 = -rt_0 - \frac{\text{Im}[C_1]}{\tau} \quad (10)$$

$$A = \exp[\text{Re}[C_0] + at_0^2] \quad (11)$$

$$\theta = \text{Im}[C_0] + \frac{r}{2}t_0^2 + \omega_0 t_0 \quad (12)$$

It is easy to show that the equality

$$f_{n+1} = \frac{f_n^2}{f_{n-1}} \exp(2C_2) \quad (13)$$

holds for all n for the sampled Gaussian function simply by plugging (2) into (13). This second-order, nonlinear difference equation can be used to gener-

ate any complex Gaussian function. Kaiser has found a clever way to implement this difference equation using two first-order coupled difference equations, each requiring only one multiplication per sample (Kaiser 1987). By computing the two coupled complex-valued difference equations

$$h_{n+1} = h_n \exp(2C_2) \quad (14)$$

$$f_{n+1} = h_{n+1} f_n \quad (15)$$

with initial conditions

$$f_0 = \exp(C_0) \quad (16)$$

$$h_1 = \exp(C_1 + C_2) \quad (17)$$

the discrete complex-valued Gaussian function in (2) is computed.

Discussion

The above method is a very fast way to compute Gaussian grains with linear modulation. Roads (1985) has indicated, however, that other envelopes besides Gaussians may be better suited for granular

synthesis. In this case, it may be desirable only to generate a linear frequency modulation with a constant amplitude and construct the windowing function another way. This is easily accomplished simply by setting the real parts of C_0 , C_1 , and C_2 to zero (or, equivalently, $A = 1$ and $a = 0$). Higher-order Gaussian envelopes can also be generated using the extensions of this technique developed by Nuttall (1987); as the order increases, these envelopes approach the rectangular window, at the cost of more multiplications for the higher orders. Nuttall also shows an equivalent way to generate polynomials using only a single addition for each order of the polynomial; this technique may be useful for the generation of windows or other functions.

Certain questions regarding implementation arise with this scheme, particularly if the computation is done on a fixed-point computer such as one of the DSP microprocessors without floating-point arithmetic. In terms of computation, these machines can compute several grains simultaneously at audio rates. This recursion scheme is an extension of the standard recursion (14) for generating a complex sinusoid, which is rarely used in signal generation applications for two reasons: first, it is feared that error due to quantization may accumulate and lead to a noisy signal; and second, the possible frequencies available with the quantized coefficients are unevenly distributed. The first problem is probably not a major concern in this application, since the grains are of relatively short duration. The uneven distribution of frequencies and chirp rates leaves relatively large gaps at lower chirp rates and frequencies. If this is a problem, it can be remedied in the usual way, using an interpolating table lookup scheme. For linear frequency modulation, the phase is described by a second-order polynomial. This polynomial can be easily calculated with only two additions per sample using Nuttall's recursion, and this phase is used as the index into the sine-cosine table, as in traditional table-lookup schemes (Rabiner and Gold 1975; Garcia 1987). If the Gaussian envelope is also desired, it can be computed using the Gaussian recursion above, using real values in the recursion. This modification should run almost as fast as the recursion and should be used with fixed-

point machines if highly accurate chirp rates and frequencies are required.

Conclusion

The inclusion of natural sounds and frequency-modulated synthetic grains greatly increases the range of possibilities available to the composer or researcher using granular music synthesis. The techniques for granular synthesis presented here are proven methods collected from the signal processing literature, and one can be confident that they will yield good results in the granular synthesis application. As with current granular synthesis usage, the sheer volume of parameters with the new grains generally requires the use of higher-level compositional units (Roads 1985). Frequency-modulated synthetic grains could easily be incorporated into current higher-level compositional systems. Natural sound granular analysis/synthesis probably requires a different framework. Real-time granular synthesis from both natural sounds and synthetic grains is now possible with one or only a few DSP microcomputers, with a current total hardware cost in the hundreds of dollars or less, and this cost is dropping rapidly. Expense or computational demands are no longer factors that limit the potential of granular synthesis; the burden now lies with researchers and composers to find ways to use this technique.

Acknowledgment

The authors would like to thank Curtis Roads for his encouragement and for introducing us to granular synthesis. We thank R. E. Crochiere and M. M. Sondhi for pointing out the work by Roucos and Wilgus to us and for their comments on various time-scale modification techniques. We also thank C. A. Schmidt-Jones for proofreading and for making the figures. We thank the National Science Foundation (NSF Grant ECS 83-14006 and a Graduate Fellowship), Schlumberger Well Services, and the Fulbright Exchange Program between West Germany and the USA for their financial support of this work.

References

- Garcia, D. 1987. "Precision Digital Sine-Wave Generation with the TMS32010," in K.-S. Lin, ed. *Digital Signal Processing Applications with the TMS320 Family, Volume I*. Englewood Cliffs: Prentice-Hall, pp. 269–289.
- Jones, D. L., and T. W. Parks. 1987. "On Computing Equally Spaced Samples of a Complex Gaussian Function." *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-35(10): 1479–1480.
- Kaiser, J. F. 1987. "On the Fast Generation of Equally Spaced Values of the Gaussian Function $A \exp(-at^2)$." *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-35(10): 1480–1481.
- Malah, D. 1979. "Time-Domain Algorithms for Harmonic Bandwidth Reduction and Time Scaling of Speech Signals." *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-27(2): 121–133.
- Nuttall, A. 1987. "Efficient Evaluation of Polynomials and Exponentials of Polynomials for Equispaced Arguments," *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-35(10): 1486–1487.
- Rabiner, L. R., and B. Gold. 1975. *Theory and Application of Digital Signal Processing*. Englewood Cliffs: Prentice-Hall.
- Roads, C. 1985. "Granular Synthesis of Sound." In C. Roads and J. Strawn, eds. *Foundations of Computer Music*. Cambridge, Massachusetts: MIT Press, pp. 145–159.
- Roucoux, S., and A. M. Wilgus. 1985. "High Quality Time-Scale Modification of Speech." *IEEE ICASSP-85 Proceedings*, New York: IEEE, pp. 493–496.