FIBONACCI AND RELATED SEQUENCES IN DIGITAL FILTERING

GONZALO R. ARCE
University of Delaware, Newark, DE 19711
(Submitted September 1982)

1. INTRODUCTION

In many communication and signal-processing systems, desired signals (sequences) are embedded in noise. Linear filters have been the primary tool for smoothing or recovering the desired signal from the degraded signal. Linear filters perform particularly well where the spectrum of the desired signal is significantly different from that of the interference. In many situations, however, the spectrum of the signal and of the noise are mixed in the same range and the performance of linear filters is very poor. Median filters can be used to circumvent these problems. Tukey [1] is generally credited with the idea of introducing nonlinear filters based on moving sample medians of the input signal. In this paper, we do not address the filtering problem, but we analyze the signal (sequence) set of median filtered binary sequences. To best explain the goal of this paper, the implementation of the median filter is described first.

To begin, take a binary sequence of length n; across this signal we slide a window that spans $2s - 1$ samples of the binary sequence, for $s = 2, 3, \ldots$. At each point of the sequence, the median of the samples within the window of the filter is computed and the output of the filter at the center sample is set equal to the computed median. To account for start-up and end effects at the two endpoints of the n-length sequence, $s - 1$ samples are appended to the beginning and end of the sequence. The value of the appended samples to the beginning is equal to the value of the first sample; similarly, the value of the appended samples to the end of the sequence equals the value of the last sample of the sequence. Figure 1(a) shows a binary signal of length 10 being filtered by a filter of window of size 3. The filtered signal is shown below. The appended samples are shown as crosses (X). Figure 1(b) shows the same sequence filtered by a filter of window size 5. Figure 1(c) shows similar results with a larger window. An interesting observation is that there exist sequences that are not modified by the median filter. Moreover, it has been shown that any finite input sequence, after repeated median filtering, will be reduced to one of these invariant sequences [2]. A sequence that is not modified by the filtering process is called a "root" sequence. The following theorem provides the upper bound on the number of successive filter passes necessary to reduce an input sequence to a root sequence [2]:

Theorem

Upon successive median filter passes, any nonroot sequence will become a root sequence after a maximum of $(n - 2)/2$ successive filterings, where n is the sequence length.

If we observe the structure of binary root sequences, we can see that they consist of identically-valued segments of at least s samples. These segments of at least s consecutive equal-valued samples are called "constant neighborhoods."

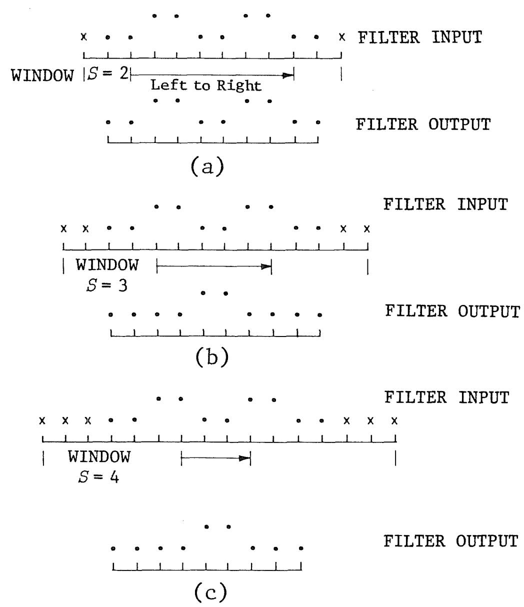


FIGURE 1. Signal Filtered by Three Different Median Filters: (a) $S = 2$ (b) $S = 3$ (c) $S = 4$

Any sequence that does not consist only of constant neighborhoods will be modified by the filter. As an example, consider a window of size 3 (i.e., $s = 2$); if a sequence contains the segment "...11011...," then, clearly, this sequence will be modified when the window is centered at the "0" sample. In this case, the shortest constant neighborhood we can have is two.

The problem addressed in this paper is concerned with the binary root sequence space of median filters. In particular, for a median filter of window size $2s - 1$, how many possible binary root sequences can we have for a given sequence length? For instance, for a window of size 3 and sequence length 4, the only possible root sequences are:

sequence 1	0 0 0 0
sequence 2	0 0 0 1
sequence 3	1 0 0 0
sequence 4	1 1 0 0
sequence 5	0 0 1 1
sequence 6	0 1 1 0
sequence 7	1 0 0 1
sequence 8	1 1 1 0
sequence 9	0 1 1 1
sequence 10	1 1 1 1

There are only 10 possible root sequences of length 4, compared to 16 possible binary sequences we can obtain if no restriction is imposed on the sequences. Thus, for a particular window size and sequence length n, we are interested in finding $R(n)$, the number of possible root signals.

2. TREE STRUCTURE FOR A WINDOW OF SIZE THREE

Consider a window of size 3 ($s = 2$). As mentioned above, the minimum constant neighborhood for this filter is 2. Now, Let us build a root signal (a signal that will not be modified by the filter). The first sample can take any arbitrary values; for purposes of illustration, let us choose the first sample to be a "0." Next, for filtering purposes, we append a sample to the left of the first "0" sample. So far the sequence is "0 0" (appended + root sequence). The second sample of the sequence can be either a "0" or a "1." Let us pick a "1" for the second sample; the entire sequence is now: "0 0 1." The third sample of the root sequence (fourth of the entire sequence) is of decisive importance; if we let it be a "1," the entire sequence would consist of two different constant neighborhoods satisfying the property of being invariant to the filter. On the other hand, if we let the third sample be a "0," then a non-allowed structure occurs and the resultant sequence would be affected by the filter. Figure 2 shows every allowable path that the root signal can take.

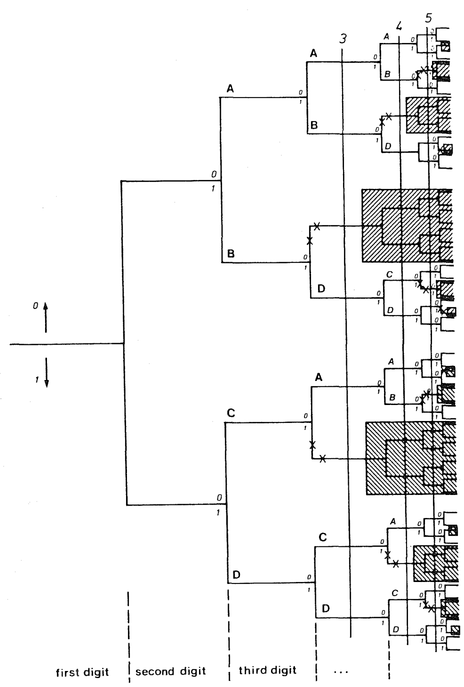


FIGURE 2. Tree Structure for a Filter of Window Size 3

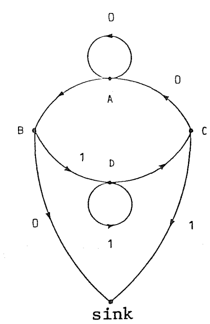
These paths branch in a tree structure fashion. If we take a close look at the tree structure, we can distinguish that sections of the tree repeat themselves as the tree propagates. This observation gives us the concept of the existence of discrete states. As is shown next, this is in fact true. These states are shown in Figure 2 and are denoted A, B, C, and D. Each state is determined by a sequence to two consecutive digits; for the filter of size 3, these states are:

$$A = \{0,\ 0\},\ B = \{0,\ 1\},\ C = \{1,\ 0\},\ D = \{1,\ 1\}.$$

Figure 2 shows how these states propagate as the sequence length increases.

Each state will generate other states; this can be seen in Figure 3, where a state transition diagram shows the state propagation. Notice that states B and C have only one allowable path. The nonallowed path is denoted by the "sink" in Figure 3. State A generates another state A plus a state B, state B generates a state D only, state C generates a state A, and finally state D generates a state D and a state C. Notice that the pattern of growth is predictable, in other words, given the number of states A, B, C, D at a given stage of the tree, we can predict the number of A, B, C, D states at the next stage. Let n denote the n^{th} stage (root sequence of length n), and let $A(n)$ be the number of A states that the tree structure has at this n^{th} stage. From the properties of the states, previously mentioned, we can write:

$$\begin{aligned} A(n+1) &= A(n) + C(n) \\ B(n+1) &= A(n) \\ C(n+1) &= D(n) \\ D(n+1) &= D(n) + B(n). \end{aligned} \tag{1}$$



State Diagram
Window = 3

FIGURE 3. State Diagram for a Filter with Window Size 3

Refer to the tree structure in Figure 2 and randomly select any stage, say stage 3. At that stage, we have two A states, two D states, one C state, and one B state; a total of 6 states (6 branches or possible roots). For a sequence of length 4, we have 10 states (or 10 possible root sequences). In general, the number of root sequences at the n stage is simply

$$R(n) = A(n) + B(n) + C(n) + D(n), \tag{2}$$

and at the $n + 1$ stage is

$$R(n+1) = A(n+1) + B(n+1) + C(n+1) + D(n+1).$$

Replacing (1) into (2), we obtain the recursive expression for $R(n + 1)$:

$$R(n+1) = 2A(n) + 2D(n) + C(n) + B(n), \tag{3}$$

with the initial conditions $A(2) = B(2) = C(2) = D(2) = 1$. Using this expression, a recursion table for the number of different states and number of roots is obtained and shown in Table 1.

Although the recursion table gives us a way to obtain the number of roots at any sequence length, a closed form solution for $R(n)$ is more desirable. From (3) and (2), we obtain

$$R(n+1) = R(n) + A(n) + D(n), \tag{4}$$

but, referring to the state diagram,

TABLE 1

Recursion Table for $R(n)$, Window = 3

Sequence Length n	$A(n)$	$B(n)$	$C(n)$	$D(n)$	$R(n)$
2	1	1	1	1	4
3	2	1	1	2	6
4	3	2	2	3	10
5	5	3	3	5	16
6	8	5	5	8	26
7	13	8	8	13	42
8	21	13	13	21	68
9	34	21	21	34	110
10	55	34	34	55	178
11	89	55	55	89	288
⋮	⋮	⋮	⋮	⋮	⋮

$$A(n) = A(n - 1) + C(n - 1),$$

and,

$$D(n) = D(n - 1) + B(n - 1).$$

Replacing these expressions for $A(n)$ and $D(n)$ into (4), we obtain

$$R(n + 1) = R(n) + R(n - 1).$$

We have obtained a difference equation for the number of roots of a binary sequence for a filter with window size 3 and initial conditions

$$R(1) = 2 \quad \text{and } R(2) = 4.$$

The solution is simply $R(n) = 2F(n + 1)$, where $F(n)$ is the Fibonacci sequence

$$F(n) = \frac{1}{\sqrt{5}}\left[\left(\frac{1 + \sqrt{5}}{2}\right)^n - \left(\frac{1 - \sqrt{5}}{2}\right)^n\right], \text{ for } n \geqslant 1.$$

3. TREE STRUCTURE FOR THE GENERAL WINDOW

Let us see what happens if we increase the window size to 5; later on we will generalize the window size to $2s - 1$. For this window, the minimum constant neighborhood length is 3. By using the same procedure as before, we obtain a tree structure for this size window and it is shown in Figure 4. The difference between the tree structures for the filters of size 3 and 5 is that for the latter we have two similar states B and two similar states C. For the filter with window size 5, the states are specified as follows:

$$A = \{0, 0, 0\},\ B1 = \{0, 0, 1\},\ B2 = \{0, 1, 1\},$$

$$C1 = \{1, 1, 0\},\ C2 = \{1, 0, 0\}, \text{ and } D = \{1, 1, 1\}.$$

The similarity between states $C1$ and $C2$ is that both sequences start a neighborhood of value "0," the difference is in that $C1$ is a delay state (will generate a state $C2$ only). Similar observations can be made for states $B1$ and $B2$. Figure 5 shows the state diagram for the filter of size 5, and the delay states

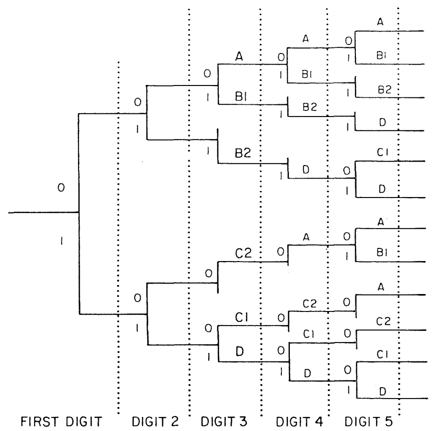


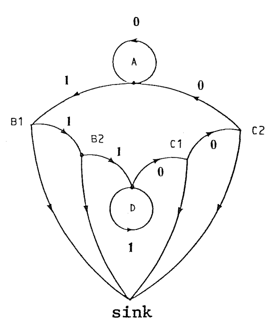
FIGURE 4. Tree Structure for a Filter with Window Size 5

can clearly be seen there. From the state diagram, we obtain the recursive expressions:

$$\begin{aligned} A(n) &= A(n-1) + C2(n-1) \\ B1(n) &= A(n-1) \\ B2(n) &= B1(n-1) \\ C1(n) &= D(n-1) \\ C2(n) &= C1(n-1) \\ D(n) &= D(n-1) + B2(n-1). \end{aligned} \tag{5}$$

As before,

$$R(n) = A(n) + B1(n) + B2(n) + C1(n) + C2(n) + D(n). \tag{6}$$



State Diagram
Window = 5

FIGURE 5. State Diagram for a Window of Size 5

Substituting (5) into (6), and after some manipulations, we find that

$$R(n + 1) = R(n) + R(n - 2). \tag{7}$$

Naturally, for a given sequence length, the number of roots decreases as we increase the window size. We have seen that, if we increase the window size, only delay states are added to the state diagram. Although by following the same procedure we could obtain the difference equation for larger window sizes, a general recursive expression for a general size filter is a more convenient result. This relation will be obtained next.

Figure 6(a) shows a state diagram for a filter of arbitrary window size $2s - 1$. The dotted line separates the diagram in odd symmetric parts. The odd symmetric correspondence is not only in a position sense, but in the multiplicity of the given states also (i.e., # of $B1$ states = # of $C1$ states, etc.). States Bi and Ci are delay states (each Ci or Bi state will be transformed into only one other state as we move along the diagram). On the other hand, states A and D not only have the previous property, but, also, they will generate another state of their own kind. Hence, for this $2s - 1$ window size filter, the number of roots is

$$R(n) = A(n) + B1(n) + B2(n) + \cdots + B[s - 1](n) + C1(n) + C2(n) + \cdots + C[s - 1](n) + D(n), \tag{8}$$

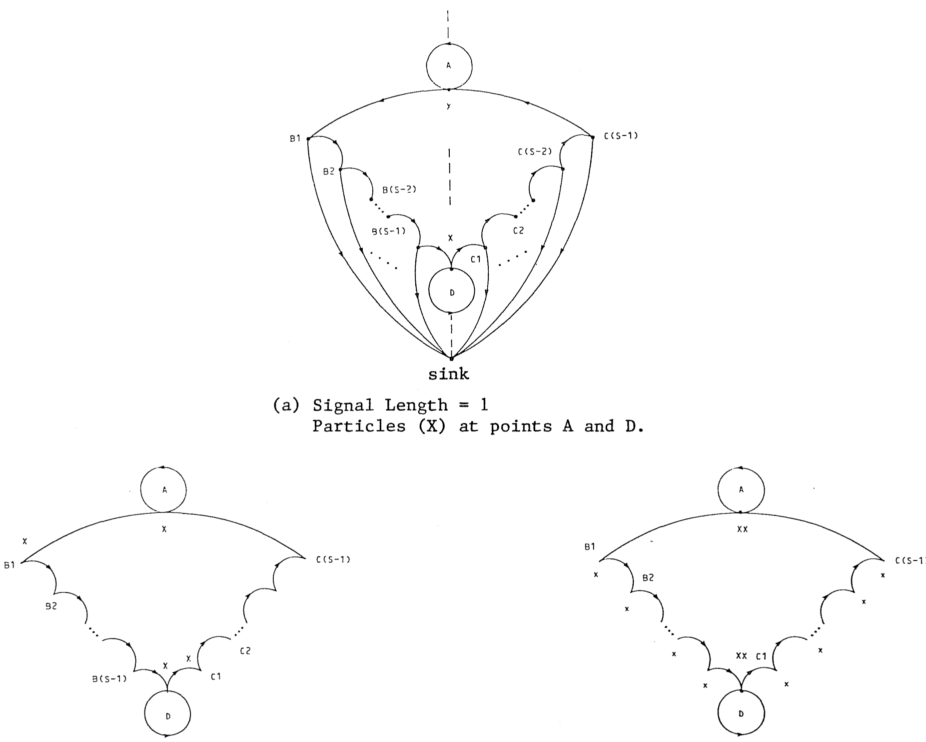
and

$$\begin{aligned} A(n) &= D(n) \\ B1(n) &= A(n - 1) \\ B2(n) &= A(n - 2) \\ &\vdots \\ B[s - 1](n) &= A(n - [s - 1]) \\ C1(n) &= B1(n) \\ C2(n) &= B2(n) \\ &\vdots \\ C[s - 1](n) &= B[s - 1](n). \end{aligned} \tag{9}$$

Therefore, $R(n)$ can be represented in terms of a recursion relation of the A states only. It is important to recall that s is the minimum constant neighborhood for a window of size $2s - 1$. We find that $R(n)$ can be written as

$$R(n) = 2\sum_{i=0}^{s-1} A(n - 1). \tag{10}$$

Let us now describe some properties of the multiplicity of A states. Refer to the state diagram for the general window size filter, Figure 6(a). Think of the state diagram as describing the propagation of particles in space. [Particles in Figure 6(a, b, c) are shown as X's.) A particle at point A represents a state A; if at a given time there would be 5 particles at point D, this would imply that there would be 5 states D, and so on. At a sequence of length 1, we have 1 state A and 1 state D; this is shown in Figure 6(a). Increasing the sequence length to 2, state will generate another state A and also generates a state $B1$. Similarly, state D generates a state D and also a state $C1$. As we can see in Figure 6(b), with a sequence of length 2, the number of states is the same as it was at a sequence of length 1. The states generated at point D move toward point A; this process goes on until the first state generated at point D gets to point A. As we can see in Figure 6(c), when the first particle generated by D reaches the A point, a particle in point A not only generates a new state A by itself, but also, it receives another state from the particle that has propagated from state A along points $C1$, $C2$, ..., $C[s - 1]$. In other words, point A has to wait s discrete intervals until the number of states in that location increases by the number of particles at point D, s intervals ago.



(a) Signal Length = 1
Particles (X) at points A and D.

(b) Signal Length = 2
(sink omitted)

(c) Signal Length = $S+1$
(sink omitted)

FIGURE 6. State Propagation for a Filter with Window Size $2s - 1$

Since the number of particles at the D point is the same as the number of particles at the A point at any time, the previous observation can be written as

$$A(n) = A(n - 1) + A(n - n). \tag{11}$$

Replacing (11) into (10), we find, after some manipulations, that

$$R(n) = R(n - 1) + R(n - s) \tag{12}$$

is the recursive expression for the number of root sequences of a filter with window size $2s - 1$, for any sequence length n. Letting

$$R(n + i - 1) = x_i(n), \tag{13}$$

we can see from (12) and (13) that

$$\begin{aligned} x_1(n + 1) &= x_2(n) \\ x_2(n + 1) &= x_3(n) \\ x_{s-1}(n + 1) &= x_s(n) \\ x_s(n + 1) &= x_s(n) + x_1(n). \end{aligned} \tag{14}$$

We can represent (14) in vector notation as

$$X(n + 1) = AX(n), \tag{15}$$

where

$$X(n) = [x_1(n),\ x_2(n),\ \ldots,\ x_s(n)]^T,$$

and where A is the bottom companion matrix:

$$A = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 1 & & 0 & 0 & 0 \\ & \vdots & & & & \vdots & \\ 0 & 0 & 0 & & 0 & 1 & 0 \\ 0 & 0 & 0 & & 0 & 0 & 1 \\ 1 & 0 & 0 & & 0 & 0 & 0 \end{bmatrix}.$$

From (13), $R(n) = [1,\ 0,\ 0,\ \ldots,\ 0]X(n)$, where $X(n)$ is the solution of (15),

$$X(n) = A^n X(0), \tag{17}$$

and where $X(0)$ are the initial conditions obtained from the tree structure or recursion table; hence,

$$R(n) = [1,\ 0,\ 0,\ \ldots,\ 0]A^n X(0). \tag{18}$$

The characteristic equation of the A matrix in (16) is obtained to be

$$\lambda^s - \lambda^{s-1} - 1 = 0. \tag{19}$$

With the help of Sturm's theorem [3], we can show that (19) does not have repeated eigenvalues; hence, we can find $R(n)$ from (18) as

$$R(n) = [1,\ 0,\ 0,\ \ldots,\ 0]MD^n M^{-1}X(0), \tag{20}$$

where M is the matrix that diagonalizes A as $M^{-1}AM = D$. In this case,

$$D = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & & \\ \vdots & & & \\ 0 & & & \lambda_s \end{bmatrix}, \tag{21}$$

and

$$\begin{array}{c} \begin{matrix} 1 & \quad 1 & \quad\quad 1 \end{matrix} \\ M = \begin{bmatrix} \lambda_2 & \lambda_2 & \lambda_s \\ \lambda_1^2 & \lambda_2^2 & \lambda_s^2 \\ \vdots & \vdots & \vdots \\ \lambda_1^{s-1} & \lambda_2^{s-1} & \lambda_s^{s-1} \end{bmatrix}, \end{array} \tag{22}$$

where $\lambda_1, \ldots, \lambda_s$ are the s distinct eigenvalues of A. Replacing (21) and (22) into (20), we obtain the general solution for $R(n)$:

$$R(n) = [\lambda_1,\ \lambda_2,\ \ldots,\ \lambda\ ]M^{-1}X(0).$$

4. CONCLUSION

We have developed a tree structure for the root sequence set of median filters of binary signals. This structure has the characteristic that the number of branches it has at each stage is described by a simple recursive expression. In the case of the filter with window = 3, the number of branches is related to the Fibonacci sequence. In general, it is shown that the number of roots $R(n)$ for a sequence of length n and window size $2s - 1$ is represented by the recurrence relation

$$R(n) = R(n - 1) + R(n - s).$$

REFERENCES

1. J. W. Tukey. *Exploratory Data Analysis*. Reading, Mass.: Addison-Wesley, 1971.
2. N. C. Gallagher & G. L. Wise. "Theoretical Analysis of the Properties of Median Filters." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, no. 6 (December 1981).
3. M. E. Van Valkenburg. *Modern Network Synthesis*. New York: Wiley, 1960.

◆◇◆◇◆