CONVERGENCE BEHAVIOR AND CONVERGENCE RATES OF STACK FILTERS

Moncef Gabbouj [†], Pao-Ta Yu [††] and Edward J. Coyle [‡]

[†]Signal Processing Laboratory, Tampere University of Technology,
P.O. Box 527, SF-33101 Tampere, Finland.

[††]Institute of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi, Taiwan 62107, R.O.C.

[‡]School of Electrical Engineering, Purdue University,
West Lafayette, IN 47907, USA.

ABSTRACT

The four different types of stack filters, type-0 through type-3, are determined by four different shapes of the on-set of the positive Boolean function from which the stack filter is constructed.

Under all three appending strategies commonly considered in the literature -- first and last value carry on strategy, constant value carry on strategy, and the circular approach -- stack filters of type-0 through type-2 possess the convergence property, while type-3 stack filters do not all share this property. Examples of cyclic behavior in type-3 stack filters are given.

The rates of convergence for stack filters of type-1 and type-2 are determined for each appending approach.

1. Introduction

The convergence property [1]-[3] and the structure of the root signals of median filters are very important for determining which filtering problems can be solved by the median filter [2]-[4]. If similar results can be obtained for stack filters, then it will become more clear which filtering problems can be solved with stack filters.

In [1] it was shown that median filters possess the following *convergence property:* any finite length signal will be filtered to a root signal after a finite number of passes of the median filter. It was shown in [3] that a signal of length L requires at most $3\left\lceil \frac{L-2}{2(N+2)} \right\rceil$, passes to guarantee convergence to a root when filtered by a median filter of window width $2N+1$.

In this paper we continue the study of the convergence behavior of the class of stack filters, which are a generalization of median filters [6]. A general theory for characterizing the existence and nature of root signals for stack filters is provided in [5] [7]. The characterization is in terms of cycles and toroids in directed graphs and the behavior of root signals constructed from these cycles. The work in [8] and [9] provides methods for determining whether any given signal is a root signal for some nontrivial stack filter, and also introduces methods for designing stack filters which have a desired set of root signals. In [10,11] it was shown that one- and two-dimensional stack filters based on symmetric threshold functions either always converge to a root signal or to an oscillation of length 2, and that recursive stack filters of this type always yield convergence to a root.

The goals of this paper are to determine the convergence behavior of stack filters under three appending strategies which have appeared in the literature, to find operations on stack filters with the convergence property which produce other filters with the convergence property, to find the number of passes which guarantees that certain stack filters will filter any input signal to a root signal, and to characterize the set of root signals of type-3 stack filters.

The paper is organized as follows. In Section 2, the different appending strategies that have been used for median and other rank order filters will be defined. They must be considered since the appending strategy which is used has a significant effect on the convergence behavior of stack filters. Section 3 then summarizes the four types of stack filters considered in the rest of the paper. Section 4 presents many properties regarding the convergence behavior of the different types of stack filters. The number of passes needed to guarantee convergence to a root signal is considered in Section 5.

Before proceeding, we summarize the notation used throughout the paper:

- X is used to indicate any signal; unless specified otherwise, a signal is assumed to consist of L samples. Each sample takes on values from the set $Q=\{0,1,\ldots,M-1\}$ for some integer $M>0$.
- $S_f(\cdot)$ denotes the stack filter based on the positive Boolean function $f(\cdot)$. $S_f(\cdot):Q^L\rightarrow Q^L$. In other words, it maps a signal of length L into another signal of length L.
- $\hat{X}$ is the appended version of the signal X.
- X^i is the output after the i'th application of the stack filter $S_f(\cdot)$: $X^{i+1}=S_f(\hat{X}^i)$.
- $f(\cdot)$ denotes a Boolean function of $2N+1$ variables where N is any nonnegative integer: $f(\cdot):B^{2N+1}\rightarrow B$, where $B=\{0,1\}$.
- $on-set(f)$ is the set $\{X\in B^{2N+1}:f(X)=1\}$.
- $off-set(f)$ is the set $\{X\in B^{2N+1}:f(X)=0\}$.

2. Appending Strategies and their Effect on Convergence

As mentioned previously, the first problem one encounters in considering the filtering of finite length signals is how to define the filtering operation when the window of the stack filter, which has a fixed window width, extends beyond the end of the signal. Three different approaches to this problem have appeared in the literature. In the description of these three approaches which appears below, we assume without loss of generality that the output of the filter is assigned the position of the center sample in the window.

CH 3006-4/91/0000 – 0216 $1.00 © IEEE

The first approach is called the first and last value carry on strategy. It consists of repeating the end values of the signal N times when the filter has a window of width $2N+1$. *This appending strategy will be referred to as strategy S1.*

The second appending strategy is called the constant value appending approach and is widely used by researchers in the field of morphology. In this approach all zeros or all ones are appended at the boundaries of the signal when the signal is binary. *For binary signals, this appending strategy will be referred to as S2-0 when appending 0^N at the boundaries of a signal and S2-1 when appending 1^N.*

The third appending approach that has been considered is called the circular appending approach. If $X = \{x(1)\ x(2)\ \cdots\ x(L)\}$ is the input signal, then the appended signal $\hat{X}$ is

$$\hat{X} = \{x(L-N+1)\ x(L-N+2)\ \cdots\ x(L)\ x(1)\ x(2)\ \cdots\ x(L)\ x(1)\ x(2)\ \cdots\ x(N)\}.$$

As indicated by the above equation, the input signal is wrapped around on itself in order to define the filtered values at the boundary points. *This appending strategy will be referred to as S3.*

The appending strategy that is used has a serious impact on the convergence properties of stack filters. A signal which is filtered to a root by a given stack filter under one appending strategy may be filtered, by the same stack filter, to a different root or to an oscillation when a different appending strategy is used.

Our goal is to determine under what conditions we can say convergence occurs under all of the above appending strategies and to determine what effect the different appending strategies have on the number of passes required to guarantee convergence. As will be seen later, both type-1 and type-2 stack filters possess the convergence property -- they filter all input signals to roots -- regardless of the appending strategy. Generalizations of this result are also obtained.

3. Classification of Stack Filters

We now review the classification of stack filters into the four different types, namely type-0 through type-3 [9].

Definition 1:

A stack filter $S_f(\cdot)$ based on the positive Boolean function $f(\cdot)$ of $2N+1$ variables is said to be

i) a type-0 stack filter iff $f \equiv 1, f \equiv 0$, or $f(\cdot)$ is the identity Boolean function.

ii) a type-1 stack filter iff the $on\text{-}set(f)$ is a proper subset of $x^N 1 x^N$, where $x \in \{0,1\}$ and x^N is any binary string of length N. Note that $on\text{-}set(f)$ must not be empty (or we would have a type-0 filter).

iii) a type-2 stack filter iff the $off\text{-}set(f)$ is a proper subset of $x^N 0 x^N$. Note that $off\text{-}set(f)$ must not be empty.

□

All those positive Boolean functions on which stack filters of type-0 are based are referred to as trivial positive Boolean functions.

Definition 2:

Any stack filter that is not a type-0, type-1 or type-2 is a type-3 stack filter.

□

Theorem 1:

The following statements are equivalent:

i) $f(\cdot)$ is a positive Boolean function which yields a type-1 stack filter of window width $2N+1$.

ii) $f(\cdot)$ is a positive Boolean function and $on\text{-}set(f)$ is a non-empty proper subset of $x^N 1 x^N$.

iii) $f(\cdot)$ is a positive Boolean function of $2N+1$ variables and

$$f(\vec{x}) = \begin{cases} 0 & \text{if } x_{N+1} = 0 \\ g(\vec{y}) & else, \end{cases}$$

where $g(\cdot)$ is a non-trivial positive Boolean function, $\vec{x} = (x_1, \ldots, x_{2N+1})$, and $\vec{y} = (x_1, \ldots, x_N, x_{N+2}, \ldots, x_{2N+1})$.

□

Theorem 2:

The following statements are equivalent:

i) $f(\cdot)$ is a positive Boolean function which yields a type-2 stack filter of window width $2N+1$.

ii) $f(\cdot)$ is a positive Boolean function and $off\text{-}set(f)$ is a non-empty proper subset of $x^N 0 x^N$.

iii) $f(\cdot)$ is a positive Boolean function and

$$f(\vec{x}) = \begin{cases} 1 & \text{if } x_{N+1} = 1 \\ h(\vec{y}) & else, \end{cases}$$

where $h(\cdot)$ is a non-trivial positive Boolean function, $\vec{x} = (x_1, \ldots, x_{2N+1})$, and $\vec{y} = (x_1, \ldots, x_N, x_{N+2}, \ldots, x_{2N+1})$.

□

We next define a partial ordering property between Boolean functions.

Definition 3:

Let $f(\cdot)$ and $g(\cdot)$ be two positive Boolean functions of n variables. We say that $f(\cdot)$ is less than or equal to $g(\cdot)$, denoted by "$f \le g$", if and only if $on(f) \subseteq on(g)$.

□

Further decomposition of type-3 stack filters is possible and helpful. It was shown in [9] that for all type-3 stack filters $S_f(\cdot)$,

$$|on(f) \cap x^N 0 x^N| \le |on(f) \cap x^N 1 x^N|, \qquad (3.1)$$

where $|A|$ is the cardinality of the set A.

Definition 4:

If equality holds in (3.1), then $S_f(\cdot)$ is called a type-3-1 stack filter; if the inequality in (3.1) is strict, then $S_f(\cdot)$ is called a type-3-2 stack filter.

□

The convergence behavior of the different stack filters defined above, and of some other stack filters, is considered in the next section.

4. Convergence Properties of Stack Filters

Some stack filters, such as the median filter, behave nicely in the sense that they will filter any arbitrary input signal of finite length to a root after successive passes of the filter. This property is referred to as the *convergence property*. More formally:

Definition 5:

A stack filter $S_f(\cdot)$ is said to possess the convergence property if and only if for each input signal X of finite length, there is a finite integer I such that $X^{I+1} = S_f(\hat{X}^I) = X^I$.

□

The results which follow hold for all three appending strategies unless specified otherwise.

Fact 1:

All three type-0 stack filters possess the convergence property and any arbitrary input signal is either a root to the filter or will be filtered to a root in a single pass.

□

This fact follows easily from the trivial nature of type-0 stack filters, which are based on positive Boolean functions which either put out 1 for every binary input, put out 0 for every binary input, or are the identity operation.

That all stack filters of window width 3 possess the convergence property, assuming S1 (first and last value carry on strategy), has been proven in [6]. In [8], we used the fact that any monotonic bounded sequence of finite-dimensional vectors converges, to analyze the convergence behavior of type-1 and type-2 stack filters, again assuming S1. Those results, see [8], are stated in the following theorem and are valid under all three appending strategies.

Theorem 3:

All type-1 and type-2 stack filters possess the convergence property.

□

Type-1 and type-2 stack filters can be considered generalizations of the erosion and dilation operations considered in morphology. Each type-1 or type-2 stack filter, or any stack filter for that matter, can have a representation in terms of a composition of erosion and dilation operations.

When the window width $2N+1$ of the filter is greater than 3, type-3 stack filters do not necessarily possess the convergence property, as the following example demonstrates.

Example 1:

Consider the following type-3 stack filter $S_f(\cdot)$ based on $f(x_1,x_2,x_3,x_4,x_5) = x_1x_5$. This filter does not possess the convergence property. Moreover, it will produce, under S1, an alternating sequence of period 2 when, for instance, the input is the binary signal 10111.

□

With the above example as motivation, we can construct a type-3-1 stack filter having a cycle with a period of any desired length.

Fact 2:

For any positive integer $n \geq 2$, there exists a type-3 stack filter which filters, under appending strategy S1, an input signal X of appropriate length L such that the sequence $\{X^i, i = 1,2,\cdots\}$ is periodic with fundamental period n; that is, $X^i = X^{i+k}$ if and only if k is an integer multiple of n.

□

Not all type-3-1 stack filters produce cycles; some possess the convergence property. The following result is extended from [6] and is valid under strategy S1.

Fact 3:

The stack filters based on the Boolean functions $f(x_1,x_2,\ldots,x_{2N+1}) = x_i$ or x_ix_{i+1} or x_ix_{i+2} possess the convergence property where i is an appropriate index in $\{1,2,...,2N+1\}$.

□

All rank-order filters possess the convergence property under appending strategy S1, [1] [4]. Rank order filters which put out the i'th largest value in the window, for $i = 2, ..., 2N$, are all type-3-2 stack filters. In other words, all rank-order filters which are not the max or min filters are type-3-2 stack filters. Although every filter in this subclass of type-3-2 filters possesses the convergence property under S1, it does not mean that every type-3-2 stack filter possesses the convergence property.

Since not all type-3 stack filters possess the convergence property, some subclasses which have some overlap with type-3 stack filters are considered in [12] where their convergence behavior is also analyzed.

Any type-3 stack filter can be decomposed into a type-1 stack filter and a type-2 stack filter [9]. This decomposition and one of its consequences are shown in the following theorems.

Theorem 4:

If $S_f(\cdot)$ is a type-3 stack filter based on the positive Boolean function $f(\cdot)$ of $(2N+1)$ variables, then

$$f(x_1,\ldots,x_{2N+1}) = g(x_1,\ldots,x_{2N+1}) + \bar{x}_{N+1}h(x_1,\ldots,x_{2N+1}),$$

where $on(g) = (on(f)\cap x^N 1 x^N)$ and $on(h) = (on(f)\cap x^N 0 x^N)\cup x^N 1 x^N$.

□

By the decomposition of any type-3 stack filter mentioned in Theorem 4, the filtering sequence produced by a type-3 stack filter $S_f(\cdot)$ can be bounded by the filtering sequences of its decompositions $S_g(\cdot)$ and $S_h(\cdot)$.

Theorem 5:

Let $S_f(\cdot)$ be a type-3 stack filter with the decomposition defined in Theorem 4. Then, assuming S1, for any finite input signal X,

$$S_g(\hat{X}^k) \leq S_f(\hat{X}^k) \leq S_h(\hat{X}^k), \quad \textit{for all } k = 0,1,\cdots.$$

□

5. The Number of Passes to Produce a Root for Binary Inputs

In this section, we will discuss the number of passes needed to reach a root signal for type-1 and type-2 stack filters and their generalizations. Throughout this section, the first appending strategy S1, first and last value carry on strategy, is assumed. The difference between this approach and the other two appending approaches will be stated separately following each result.

Only binary signals are considered in this section. The number of passes needed to produce a root for a multi-valued signal is just the largest number of filter passes required by any of its binary threshold signals to produce a binary root.

It was stated previously that any stack filter of type-0 will take at most a single pass to filter any input to a root. This fact holds for all three appending strategies discussed earlier.

For type-1 and type-2 stack filters, the next two theorems show that the number of required passes is linear in the length of the input signal.

Let $X \in B^L$, i.e., X is a binary sequence of length L. Denote the i'th element of X by $x(i)$. Suppose X is non-constant but otherwise arbitrary. Then we have the following.

Theorem 6:

Any non-constant input signal $X \in B^L$ will take at most $\sum_{i=1}^{L} x(i)$ iterations in order to converge to a root by any type-1 stack filter. Any non-constant input signal $X \in B^L$ will take at most $(L - \sum_{i=1}^{L} x(i))$ iterations to converge to a root by any type-2 stack filter.

□

A type-1 stack filter that actually achieves this bound is the one based on the following positive Boolean function:

$$f(x_1, \ldots, x_{2N+1}) = x_N x_{N+1}. \quad (5.2)$$

It is easy to show that $S_f(\cdot)$ is a type-1 stack filter and that when operating on the input signal $X = 01^{L-1}$, X will converge to $X_R \equiv 0^L$ in exactly

$$\# 1_X = \sum_{i=1}^{L} x(i) = L-1$$

iterations.

The circular and the constant value carry-on appending strategies yield similar results, that is the rate of convergence of type-1 stack filters for signals of length L is of order L.

A type-2 stack filter that actually meets this upper bound in Theorem 6 is the following:

$$f(x_1, \ldots, x_{2N+1}) = x_N + x_{N+1}.$$

When $f(\cdot)$ operates on the input signal $X = 10^{L-1}$, it reduces it to a root $X_R = 1^L$ in exactly $L - 1 = L - \sum_{i=1}^{L} x(i)$ iterations.

The same observations about the other two appending strategies are also valid for type-2 stack filters. That is, they require $L-1$ iterations in the worst case to reach a root. However, we shall make the following important remark.

Remark:

A given binary signal X might converge to different roots when filtered with a type-1 or a type-2 stack filter using different appending strategies. The following example illustrates this behavior.

Example 2:

Suppose $X = 111011$ is to be filtered by the type-1 stack filter $S_f(\cdot)$ based on the positive Boolean function of (5.2). Then, under S1, the signal X will be filtered to 111000 in two passes; whereas, under S2-0, the signal X will be filtered to 000000 in three passes.

□

6. Conclusion

The convergence behavior of type-0 through type-3 stack filters has been investigated. It was shown that stack filters of type-0 through type-2 all possess the convergence property; that is, they filter any input signal to a root after consecutive passes of the filter under any appending strategy.

A counter example was given to show that not all type-3 stack filters have this convergence property.

We next investigated the rate of convergence for convergent stack filters. We showed that stack filters of type-0 will take at most a single pass to filter any input signal to a root. The rate of convergence of type-1 and type-2 stack filters was shown to be linear in the length of the input signal.

7. References

[1] N.C. Gallagher, Jr., and G.L. Wise, "A theoretical analysis of the properties of median filters," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* vol. ASSP-29, no. 6, pp. 1136-1141, December 1981.

[2] J.P. Fitch, E.J. Coyle, and N.C. Gallagher, Jr., "Root properties and convergence rates of median filters," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* vol. ASSP-33, no. 1, pp. 230-240, February 1985.

[3] ____, "Some convergence properties of median filters," *IEEE Trans. on Circuits And Systems,* vol. CAS-33, no. 3, pp. 276-286, March 1986.

[4] ____, "Median filtering by threshold decomposition," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* vol. ASSP-32, no. 6, pp. 1183-1188, December 1984.

[5] E.J. Coyle, J.-H. Lin and M. Gabbouj, "Optimal stack filtering and the estimation and structural approaches to image processing," *IEEE Trans. Acoustics, Speech, Signal Processing,* vol. ASSP-37, pp. 2037-2066, Dec. 1989.

[6] P.D. Wendt, E.J. Coyle, and N.C. Gallagher, Jr., "Stack filters," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* vol. ASSP-34, no. 4, pp. 898-911, August 1986.

[7] M. Gabbouj and E.J. Coyle, "Minimum mean absolute error stack filtering with structural constraints and goals," *IEEE Trans. on Acoustics, Speech and Signal Processing,* vol. ASSP-38, pp. 955-965, June 1990.

[8] P.-T. Yu and E.J. Coyle, "Convergence behavior and N-roots of stack filters," *IEEE Trans. on Acoustics, Speech, and Signal Processing.* vol. ASSP-38, pp. 1529-1544, Sept. 1990.

[9] ____, "The classification and associative capability of stack filters," *Proc. of the 27th Annual Allerton Conf. on Comm., Control, Computing,* University of Illinois at Urbana-Champaign, IL, Sept. 1989; and submitted to the *IEEE Trans. on Acoustics, Speech, and Signal Processing.*

[10] P.D. Wendt, "Nonrecursive and recursive stack filters and their filtering behavior," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* vol. ASSP-38, pp. 2099-2107, December 1990.

[11] ____, "Convergence properties of multi-dimensional stack filters," *Proc. of the 1990 SPIE/SPSE Symp. on Electronic Imaging,* Feb. 11-16, 1990, Santa Clara, CA.

[12] M. Gabbouj, P.-T. Yu and E.J. Coyle, "Convergence behavior and root signal sets of stack filters," *Circuits, Systems and Signal Processing, Special issue on Median and Morphological Filtering, to appear.*