[11] E. T. Jaynes, "Information theory and statistical mechanics," in *Statistical Physics*, K. W. Ford, Ed. New York: Benjamin, 1963, pp. 181-218.
[12] T. Seidenfeld, "Why I am not an objective Bayesian: Some reflections prompted by Rosenkranz," *Theory and Reason*, vol. 11, pp. 413-440, 1979.
[13] S. Kullback, *Information Theory and Statistics.* New York: Wiley, 1959.
[14] J. A. Edward and M. M. Fitelson, "Notes on maximum entropy processing," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 232-234, Mar. 1973.
[15] A. Van den Bos, "Alternative interpretation of maximum entropy analysis," *IEEE Trans. Inform. Theory*, vol. IT-17, July 1971.
[16] B. R. Frieden, "Restoring with maximum likelihood and maximum entropy," *J. Opt. Soc. Amer.*, vol. 62, no. 4, pp. 511-518, 1972.
[17] F. A. Graybill, *Matrices with Applications in Statistics*, 2nd ed. Belmont, CA: Wadsworth, 1983.

David R. Farrier was born in Plymouth, England on February 12, 1950. He received the B.Sc. (first class honours) and Ph.D. degrees from Southampton University, Southampton, England, in 1971 and 1976, respectively, both in electrical engineering.

From 1972 to 1975, he was a Research Assistant at Southampton University, studying adaptive signal processing. From 1975 to 1979 he was employed by the United Kingdom Atomic Energy Authority, where he was concerned with the modeling of a nuclear process plant. In 1979 he returned to the Department of Electrical Engineering, Southampton University, as a Lecturer. His current interests are adaptive signal processing, detection, estimation, and control.

Median Filtering by Threshold Decomposition

J. PATRICK FITCH, EDWARD J. COYLE, AND NEAL C. GALLAGHER, JR., MEMBER, IEEE

Abstract—**Median filters are a special class of ranked order filters used for smoothing signals. Repeated application of the filter on a quantized signal of finite length ultimately results in a sequence, termed a root signal, which is invariant to further passes of the median filter. In this paper, it is shown that median filtering an arbitrary level signal to its root is equivalent to decomposing the signal into binary signals, filtering each binary signal to a root with a binary median filter, and then reversing the decomposition. This equivalence allows problems in the analysis and the implementation of median filters for arbitrary level signals to be reduced to the equivalent problems for binary signals. Since the effects of median filters on binary signals are well understood, this technique is a powerful new tool.**

I. Introduction

A median filter maps a class of input signals into an associated set of root sequences. Each of these root signals is by definition invariant to additional filter passes and is the result of repeated filter passes on one or more of the input signals [1]. One effective use of median filters has been the reduction of high frequency and impulsive noise in digital images without the extensive blurring and edge destruction associated with linear filters [2]. Other applications include the smoothing of noisy pitch contours in speech signals and data compression using the root signal properties combined with a block truncation coding (BTC) technique [3], [4]. In practice, processing and delay times, possible coding schemes, and output signal space characteristics can all be influenced by both the number of roots and the number of filter passes necessary to reach a root.

The implementation of a standard median filter requires a simple nonlinear digital operation. To begin, take a sampled signal of length L; across this signal slide a window that spans $2N + 1$ points. The filter output at each window position is given the same position as the sample point at the center of the window and is set equal to the median value of the $2N + 1$ signal samples in the window. Start up and end effects are accounted for by appending N samples to both the beginning and the end of the sequence. The front appended samples are given the value of the first signal sample; similarly, the rear appended samples receive the value of the last sample of the signal. In Fig. 1 we present an example where a median filter of window width 5, $N = 2$, is applied to a signal with three levels. The output is given for each pass of the median filter until a root signal is reached. Recall that the basic idea is to rank the samples in the window and select the median value as the filter output.

Manuscript received October 21, 1983; revised June 1, 1984. This work was supported by the National Science Foundation under Grants ECS-8306235 and ECS-8203754.

J. P. Fitch is with the Engineering Research Division, Lawrence Livermore National Laboratory, Livermore, CA 94550.

E. J. Coyle and N. C. Gallagher, Jr. are with the School of Electrical Engineering, Purdue University, West Lafayette, IN 47907.

0096-3518/84/1200-1183$01.00 © 1984 IEEE

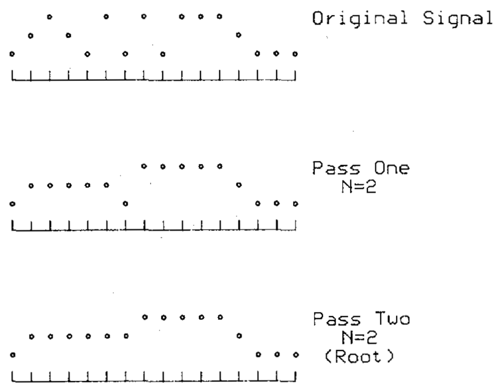


Fig. 1. Three-level signal filtered to a root in two passes by a window width 5 standard median filter.

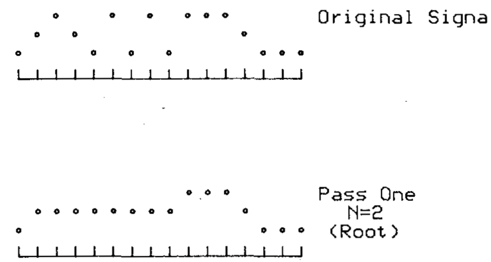


Fig. 2. Three-level signal filtered to a root by one pass of a window width 5 recursive median filter.

Recursive median filtering [6] is a modification of the standard process in which the center sample is replaced by the computed median before we shift the window to the next point. Therefore, the N leftmost samples in a window are computed medians. Fig. 2 illustrates the results of applying a window width 5 recursive median filter to the same signal as was used in the previous figure. Note that only one pass is required to reach a root with recursive median filters. Also, for this signal the root of the recursive filter is different from the root of the standard filter appearing in the previous figure.

As can be seen from the preceding discussion, median filtering is a highly nonlinear operation. The traditional tools of linear analysis–the representation of signals as the superposition of sinusoids and the separate filtering of each sinusoid–cannot be used to analyze the effects of this nonlinear operation on arbitrary signals. An example of the difficulty caused by the loss of these tools is that many properties of the median filter which can be easily obtained for binary signals are not readily extended to signals with an arbitrary number of levels. To make such extensions more natural, we need a decomposition and a set of simple signals which perform the same function for median filters that superposition and the set of sinusoids perform for linear filters.

The primary contribution of this paper is the development of the decomposition and the set of simple signals which work for median filters. The decomposition is called threshold decomposition and the simple signals are the well understood binary signals [4]-[6]. With this decomposition, a k-level signal is reduced to the sum of $k - 1$ binary signals, each of the binary signals is median filtered separately, and then the results are recombined via superposition to get the output of the filter.

In Section II, we introduce the algorithms for decomposition, filtering, and reconstruction of k-level signals when the desired filtering operation is the standard median filter operation. The proof of the equivalence of the two operations is provided. In Section III, the same process is proposed for the recursive median filter, and the same results are obtained. It is also noted that these methods apply to any order rank operation [6], not just the median. Section IV provides some examples to illustrate the new filtering process. Section V is a discussion of the results and some conclusions.

II. Median Filtering by Threshold Decomposition

We assume throughout this paper that the input signal is a discrete sequence of length L which takes on the value $a(m)$ at position m, $1 \leqslant m \leqslant L$, and that for each m, $a(m)$ is quantized to one of the k values $0, 1, \cdots, k-1$. Let $y_S(m)$ be the output at position m of a standard median filter with window width $2N+1$ applied to the input sequence $a(m)$.

Define the level i threshold decomposition of the original signal at point m to be

$$t_0^i(m) = \begin{cases} 1 & \text{if } a(m) \geqslant i \\ 0 & \text{if } a(m) < i \end{cases} \tag{1}$$

with $1 \leqslant m \leqslant L$ and $1 \leqslant i \leqslant k-1$. Applying the standard median filter [1] to these thresholded values gives another binary sequence

$$x_S^i(m) = \text{median } \{t_0^i(m-N), \cdots, t_0^i(m), \cdots, t_0^i(m+N)\} \tag{2}$$

where start up and end effects are accounted for by repeating the first or last value of the signal as described in the Introduction.

The relation of the binary valued, median filtered threshold sequence to the output $y_s(m)$ of the median filter applied to the original k-level signal is provided by the following.

Lemma 1: There exists a mapping $f(\cdot)$ from the set of binary median filtered sequences $x_s^i(m)$, $1 \leqslant i \leqslant k-1$ to the signal space of k-leveled signals such that $y_s(m) = f(x_s^i(m)$, $1 \leqslant i \leqslant k-1)$.

Proof by construction: Define $I(A)$ to be the indicator function of the event A, that is

$$I(A) = \begin{cases} 1 & \text{if } A \text{ is true} \\ 0 & \text{if } A \text{ is false.} \end{cases}$$

Examine the output of one filter pass on the binary level i threshold decomposition sequence at any position m, $1 \leqslant m \leqslant L$

$$x_s^i(m) = \text{median } \{t_0^i(m-N), \cdots, t_0^i(m), \cdots, t_0^i(m+N)\} \tag{3}$$

$$= \begin{cases} 1 & \text{if } \displaystyle\sum_{n=-N}^{N} t_0^i(m+n) \geqslant N+1 \\ 0 & \text{if } \displaystyle\sum_{n=-N}^{N} t_0^i(m+n) \leqslant N \end{cases}$$

$$= I\left\{ \sum_{n=-N}^{N} t_0^i(m+n) \geqslant N+1 \right\} \tag{4}$$

$$= I\{\text{at least } N+1 \text{ elements in } t_0^i(m-N), \cdots, t_0^i(m+N) \text{ equal } 1\}$$
$$= I\{\text{at least } N+1 \text{ elements in } a(m-N), \cdots, a(m+N) \text{ are} \geqslant i\}. \tag{5}$$

From (5) we see immediately that if $x_s^p(m) = 1$, then for at least $N+1$ positions in the window the signal value $a(n)$, $m - N \leqslant n \leqslant m + N$, is greater than or equal to p. But this implies that at least $N+1$ $a(n)$'s in the window are greater than or equal to q for all $q \leqslant p$. That is, $x_s^q(m) = 1, 1 \leqslant q \leqslant p$. This gives us the first property of the threshold decomposition.

Property 1: If $x_s^p(m) = 1$ then $x_s^q(m) = 1$ for $1 \leqslant q \leqslant p$.

We can now describe a mapping to the y's from the x's. For any m such that $1 \leqslant m \leqslant L$, we have

$$y_s(m) = \text{median}\,(a(m-N), \cdots, a(m), \cdots, a(m+N))$$
$$= \max\,(0, i\text{: at least } N+1\ a\text{'s in window are} \geqslant i)$$
$$= \max\,(0, i\text{: } I(\text{at least } N+1\ a\text{'s in window are} \geqslant i) = 1). \tag{6}$$

The combination of (5) and (6) yields

$$y_s(m) = \max\,\{0, i\text{: } x_s^i(m) = 1\}$$

and by Property 1 of the decomposition

$$y_s(m) = \sum_{i=1}^{k-1} x_s^i(m) \tag{7}$$
$$= f\{x_s^i(m), 1 \leqslant i \leqslant k-1\}. \qquad \square$$

In loose terms, the function $f(\cdot)$ stacks the binary signals $x_s^i(m)$ on top of one another starting with $i = 1$. The value of the output at position m is then the highest level at position m in the stack at which a 1 appears.

The function $f(\cdot)$ constructed above is shown to be the inverse of the threshold decomposition by the following lemma.

Lemma 2: The binary sequences $t_s^i(m)$, $1 \leqslant m \leqslant L$, and $1 \leqslant i \leqslant k-1$, obtained by thresholding $y_s(m)$, $1 \leqslant m \leqslant L$, are identical to the binary sequences $x_s^i(m)$, $1 \leqslant m \leqslant L$, and $1 \leqslant i \leqslant k-1$, from Lemma 1.

Proof:

$$t_s^i(m) = \begin{cases} 1 & \text{if } y_s(m) \geqslant i \\ 0 & \text{else.} \end{cases}$$

Using (7) from the proof of Lemma 1, we obtain

$$t_s^i(m) = \begin{cases} 1 & \text{if } \sum_{q=1}^{k-1} x_s^q(m) \geqslant i \\ 0 & \text{else.} \end{cases}$$

Property 1 allows a reduction to the following:

$$t_s^i(m) = \begin{cases} 1 & \text{if } x_s^i(m) = 1 \\ 0 & \text{else} \end{cases} \tag{8}$$
$$= x_s^i(m). \qquad \square$$

The above results show how one pass of a median filter over the input signal is equivalent to first thresholding the signal, then filtering each threshold sequence, and finally reconstructing the output using the function $f(\cdot)$. This operation can clearly be repeated for each pass of the median filter until a root signal is obtained. A simple inductive argument shows that the intermediate reconstructions can be omitted. These results are summarized in the following theorem.

Theorem 1: The root signal associated with a window width $2N+1$ median filter can be obtained by thresholding the original signal, filtering the resulting binary signals to roots, and then mapping these binary roots back to the k-level root signal using the function $f(\cdot)$ constructed in Lemma 1.

This result provides a new tool for both the implementation and the analysis of the median filtering operation applied to arbitrary level signals.

III. Recursive Median Filtering by Threshold Decomposition

Recursive median filters are known to have different properties than standard median filters [6]. For instance, the output after one pass of a recursive median filter is invariant to additional passes of the same filter—i.e., it is a root. As was shown by the examples in the Introduction, this root may not equal the standard median filter root for the same signal. Furthermore, the output of a recursive median filter is affected not only by the window size, but also by the direction the window slides across the signal. If an algorithm similar to Theorem 1 can be found for recursive median filters, we could compare the two types of filters by analyzing the binary signal case. In the following theorem, a property analogous to the result of the preceding section is given for recursive median filters.

Theorem 2: The root signal associated with a window width $2N+1$ recursive median filter can be obtained by thresholding the original signal, applying the recursive median filter to the resulting binary signals, and then mapping these binary roots back to the k-level root signal using the function $f(\cdot)$ constructed in Lemma 1 of the preceding section.

The proof of this theorem is best presented as a series of properties for recursive median filters. Throughout this section of the paper let $y_r(m)$ denote the output at position m of a recursive median filter with window width $2N+1$ moving left to right across the input sequence $a(m)$. Define the level i threshold decomposition of the original signal at position m to be

$$t_0^i(m) = \begin{cases} 1 & \text{if } a(m) \geqslant i \\ 0 & \text{if } a(m) < i \end{cases} \tag{9}$$

with $1 \leqslant m \leqslant L$ and $1 \leqslant i \leqslant k-1$. Recursive median filtering the thresholded values gives another binary sequence

$$x_r^i(m) = \text{median}\,\{x_r^i(m-N), \cdots, x_r^i(m-1), t_0^i(m), \cdots, t_0^i(m+N)\}. \tag{10}$$

Property 1.r: $y_r(1) = f(x_r^i(1), 1 \leqslant i \leqslant k-1)$. That is, the reconstruction function, defined in (7) of the previous section as $f(\cdot)$, works for the first point in the input signal. Note that a similar property holds at the last point of the signal when a recursive median filter moving from right to left across the signal is used.

Proof: To start the filter at position 1, N points of value $a(1)$ are appended to the beginning of the signal. The output of the recursive filter at position 1 is

$$\begin{aligned} y_r(1) &= \text{median}\,(a(1), \cdots, a(1), a(2), \cdots, a(N)) \\ &= a(1) \\ &= \sum_{i=1}^{k-1} t_0^i(1). \end{aligned} \tag{11}$$

The level i threshold decomposition sequences recursively median filtered at position m are given by $x_r^i(m)$. So at position 1, after appending N points, we have

$$\begin{aligned} x_r^i(1) &= \text{median}\,\{t_0^i(1), \cdots, t_0^i(1), t_0^i(2), \cdots, t_0^i(N)\} \\ &= t_0^i(1). \end{aligned} \tag{12}$$

Substituting (12) into (11), we obtain

$$\begin{aligned} y_r(1) &= \sum_{i=1}^{k-1} x_r^i(1) \\ &= f\{x_r^i(1), 1 \leqslant i \leqslant k-1\} \end{aligned} \tag{13}$$

which shows that the function $f(\cdot)$ from (7) for the standard median filter will reconstruct the recursive median filtered threshold values at position 1 of the signal. □

Property 2.r: If $x_r^p(m) = 1$, then $x_r^q(m) = 1$, $1 \leqslant q \leqslant p$, $1 \leqslant m \leqslant L$. If the recursively filtered binary sequences were stacked according to threshold value, then the interpretation of this property is that a 1 occurring at some level implies all the binary sequences of smaller threshold levels are also 1.

Proof by induction: We first note that by Property 1.r this claim is true at position 1 of the signal. If assuming that the property holds at positions 1 through $m - 1$ implies that it is valid at position m, we then know by induction that it holds for any position of the signal. So, assume that m is fixed and $x_r^p(n) = 1$ implies $x_r^q(n) = 1$ for all q such that $1 \leqslant q \leqslant p$ and $1 \leqslant n \leqslant m - 1$. If for some n between 1 and $m - 1$ we know that $x_r^p(n) = 0$, then we cannot say whether $x_r^q(n)$ is 0 or 1 for $1 \leqslant q < p$. That is, $x_r^p(n) = 0$ implies that $x_r^q(n) \geqslant x_r^p(n) = 0$ for any $1 \leqslant q \leqslant p$. In other words, the number of 1's in the sequence $x_r^p(m - N), \cdots, x_r^p(m - 1)$ can only stay the same or increase as the parameter p is decreased to q, $1 \leqslant q \leqslant p$. This allows us to conclude that if

$$\begin{aligned} 1 &= x_r^p(m) \\ &= \text{median}\,\{x_r^p(m - N), \cdots, x_r^p(m-1), t_0^p(m), \\ &\quad \cdots, t_0^p(m + N)\} \end{aligned}$$

then

$$\begin{aligned} 1 &= \text{median}\,\{x_r^q(m - N), \cdots, x_r^q(m-1), t_0^p(m), \\ &\quad \cdots, t_0^p(m + N)\}. \end{aligned} \tag{14}$$

Similarly by the definition of the threshold decomposition, we know that $t_0^p(n) = 1$ implies that $t_0^q(n) = 1$ for $1 \leqslant q \leqslant p$ and $1 \leqslant n \leqslant L$. Therefore, the number of 1's in the window cannot decrease when we replace $t_0^p(n)$ with $t_0^q(n)$ in (14). This leads to

$$\begin{aligned} 1 &= \text{median}\,\{x_r^q(m - N), \cdots, x_r^q(m-1), t_0^q(m), \\ &\quad \cdots, t_0^q(m + N)\} \\ &= x_r^q(m) \end{aligned}$$

for $1 \leqslant q \leqslant p$. Summarizing, Property 2.r holds at position m whenever it holds for positions 1 through $m - 1$. But, as was noted at the beginning of the proof, Property 2.r always holds at position 1 of the signal. By induction on m, the proof of Property 2.r is complete. □

Property 3.r: If the first $m - 1$ positions of the signal can be successfully decomposed and reconstructed for a recursive median filter, then $x_r^i(n) = 1$ if and only if $y_r(n) \geqslant i$, $1 \leqslant n \leqslant m - 1$. That is, if

$$\begin{aligned} y_r(n) &= f\{x_r^i(n), \quad 1 \leqslant i \leqslant k-1\} \\ &= \sum_{i=1}^{k-1} x_r^i(n) \end{aligned} \tag{15}$$

for all n such that $1 \leqslant n \leqslant m - 1$ where m is any fixed integer from 2 to the signal length L, then $x_r^i(n) = 1$ if and only if $y_r(n) \geqslant i$, $1 \leqslant n \leqslant m - 1$.

Proof: Begin the proof by thresholding the recursively filtered values $y_r(n)$ for $1 \leqslant n \leqslant m - 1$. This gives for each i a sequence $t_r^i(n)$, where

$$t_r^i(n) = \begin{cases} 1 & \text{if } \; y_r(n) \geqslant i \\ 0 & \text{else.} \end{cases} \tag{16}$$

Using the assumptions of this property given in (15), we obtain

$$t_r^i(n) = \begin{cases} 1 & \text{if } \; \displaystyle\sum_{q=1}^{k-1} x_r^q(n) \geqslant i \\ 0 & \text{else.} \end{cases}$$

Now invoking Property 2.r, the previous equation reduces to

$$t_r^i(n) = \begin{cases} 1 & \text{if } \; x_r^i(n) = 1 \\ 0 & \text{else.} \end{cases} \tag{17}$$

Combining (16) and (17), we have $x_r^i(n) = 1$ if and only if $y_r(n) \geqslant i$, $1 \leqslant n \leqslant m - 1$. □

This result is similar to Lemma 2 for standard median filters. However, it is important to note that we assumed the decomposition worked for the first $m - 1$ positions.

Property 4.r: If the first $m - 1$ positions of the signal can be successfully decomposed and reconstructed for a recursive median filter, then so can the mth position.

Proof: First note that because the same hypothesis is used, any results from the proof of Property 3.r can be used in this proof. Examine the recursively filtered binary sequences at position m.

$$\begin{aligned} x_r^i(m) &= \text{median}\,\{x_r^i(m - N), \cdots, x_r^i(m-1), t_0^i(m), \\ &\quad \cdots, t_0^i(m + N)\} \end{aligned} \tag{18}$$

$$= \begin{cases} 1 & \text{if } \; \displaystyle\sum_{n=1}^{N} x_r^i(m-n) + \sum_{n=0}^{N} t_0^i(m+n) \geqslant N + 1 \\ 0 & \text{if } \; \displaystyle\sum_{n=1}^{N} x_r^i(m-n) + \sum_{n=0}^{N} t_0^i(m+n) < N + 1 \end{cases}$$

$$= I\left\{\sum_{n=1}^{N} x_r^i(m-n) + \sum_{n=0}^{N} t_0^i(m+n) \geqslant N + 1\right\}$$

$$= I\left\{\sum_{n=1}^{N} x_r^i(m-n) \geqslant N+1-\sum_{n=0}^{N} t_0^i(m+n)\right\}$$

$$= I\left\{\text{at least } N+1-\sum_{n=0}^{N} t_0^i(m+n) \text{ elements in } x_r^i(m-N), \cdots, x_r^i(m-1) \text{ equal } 1\right\}.$$

Now using Property 3.*r* we can change the filtered threshold values in the argument of the indicator function to the filtered values of the original signal denoted by the y_r's.

$$x_r^i(m) = I\left\{\text{at least } N+1-\sum_{n=0}^{N} t_0^i(m+n) \text{ elements in } y_r(m-N), \cdots, y_r(m-1) \text{ are} \geqslant i\right\}. \tag{19}$$

The analysis of the filtered version of the original signal follows

$$y_r(m) = \text{median}\,\{y_r(m-N), \cdots, y_r(m-1), a(m), \cdots, a(m+N)\} \tag{20}$$
$$= \max\,\{0, i\text{: at least } N+1 \text{ of } y_r(m-N), \cdots, a(m+N) \text{ are} \geqslant i\}.$$

Using the threshold decomposition, we know that the number of $a(m), \cdots, a(m+N)$ which are greater than or equal to i is given by $\sum_{n=0}^{N} t_0^i(m+n)$. Therefore,

$$y_r(m) = \max\left\{0, i\text{: at least } N+1-\sum_{n=0}^{N} t_0^i(m+n) \text{ elements in } y_r(m-N), \cdots, y_r(m-1) \text{ are} \geqslant i\right\}$$

$$y_r(m) = \max\left\{0, i\text{: } I(\text{at least } N+1-\sum_{n=0}^{N} t_0^i(m+n) \text{ elements in } y_r(m-N), \cdots, y_r(m-1) \text{ are} \geqslant i) = 1\right\} \tag{21}$$

and by (19), we have

$$y_r(m) = \max\,\{0, i\text{: } x_r^i(m) = 1\}.$$

Using Property 2.*r* we convert this to a sum

$$y_r(m) = \sum_{i=1}^{k-1} x_r^i(m)$$
$$= f\{x_r^i(m), \quad 1 \leqslant i \leqslant k-1\}. \qquad \square$$

Property 4.*r* states that if the threshold decomposition and reconstruction works for the first $m-1$ positions of the signal, then it works for the mth position. Using Property 1.*r* we know the function $f(\cdot)$ always works for recursive filters at position 1 of the signal. Therefore, the combination of these two properties provides an inductive proof that the threshold decomposition technique using recursive filters works at any position of the signal. The proof of Theorem 2 is complete.

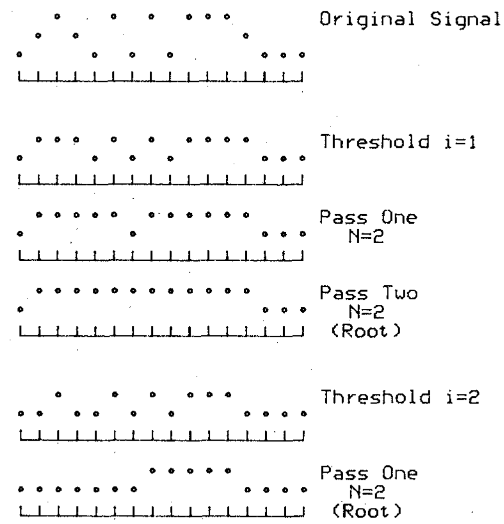


Fig. 3. The threshold decomposition of a three-level signal with the resultant standard median filter, $N = 2$, output sequences.

Since recursive median filters converge in one pass, the algorithm introduced in the preceding theorem has many practical advantages over the algorithm using the standard median filter. A simple parallel architecture with one binary recursive median filter preceded by a threshold device may be used for each level.

We note here that a simple modification of the proofs of Theorems 1 and 2 shows that the same results hold if we substitute any nth-order operation [6] for the median operation. Thus, fast implementations and analytically useful decompositions exist for these filters as well.

IV. Examples

In Fig. 3 we use the same three-level signal as in the Introduction to demonstrate the technique of threshold decomposition. After obtaining the decomposition, passes of a window width 5 standard median filter are performed until all the binary sequences are roots. As expected, the maximum number of passes to reach a root for the binary sequences equals the number of passes to reach a root with the three-level signal, as seen in Fig. 1. After each pass of the median filter we can combine the resulting binary sequences to obtain the corresponding three level signal. Comparisons between Fig. 1 and Fig. 3 verify this fact.

Similar comparisions for recursive filters can be made between Fig. 2 and Fig. 4. With a recursive filter, however, only one pass is necessary to reach a root.

V. Conclusions

The threshold decomposition and the set of binary signals perform the same function for median filters that superposition and sinusoids perform for linear filters—they allow complex problems to be decomposed into simpler problems. This has very fortunate practical and theoretical consequences.

On the theoretical side, it shows the analysis of the median filter's effect on multilevel signals is reduced to the much simpler analysis of binary signals. It is now clear that any of the properties in previous papers which were limited to binary signals can now be extended in straightforward fashion to multilevel signals. For example, it is known [5] that a significant

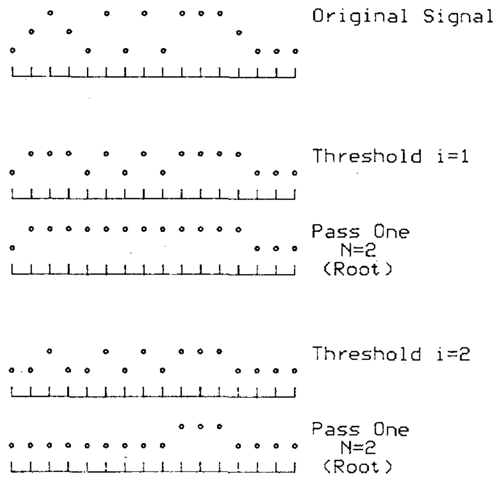


Fig. 4. The threshold decomposition of a three-level signal with the resultant recursive median filter, $N = 2$, output sequences.

reduction (by several orders of magnitude) in the number of filter passes needed to produce the desired smoothing of long signals is possible if we only require that the output of the filter be a root with a probability of 0.95. Theorem 1 implies that the same reduction in the required number of passes to a root is valid for multilevel signals as well.

On the practical side, the decomposition has an important impact on the implementation of median filters. It shows that a median filter for a multilevel signal is simply a parallel connection of filters for binary signals. Furthermore, since the output of the median filter for a binary signal is found by counting the number of 1's in the window and comparing the result to a threshold [see (4)], these filters become almost trivial to implement—complicated ranking is no longer needed. The possibility of VLSI implementation is apparent.

These theoretical results should improve our understanding of the behavior of these filters, particularly their success in the impulsive noise environment [2], [3], [7]. The practical results should lead to the use of these filters in many new real-time signal processing applications, particularly real-time image processing.

References

[1] N. C. Gallagher and G. L. Wise, "A theoretical analysis of the properties of median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 1136–1141, Dec. 1981.

[2] T. S. Huang and G. J. Yang, "Median filters and their applications to image processing," School Elec. Eng., Purdue Univ., West Lafayette, IN, Tech. Rep. EE 80-1, Jan. 1980.

[3] L. R. Rabiner, M. R. Sambur, and C. E. Schmidt, "Applications of a nonlinear smoothing algorithm to speech processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 552–557, Dec. 1975.

[4] G. A. Arce and N. C. Gallagher, Jr., "State description for the root-signal set of median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 894–902, Dec. 1982.

[5] J. P. Fitch, E. J. Coyle, and N. C. Gallagher, Jr., "Root properties and convergence rates of median filters," in *Proc. 21st Annu. Allerton Conf. Commun., Contr., Comput.*, Oct. 1983.

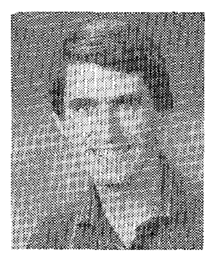
[6] T. A. Nodes and N. C. Gallagher, "Median filters: Some modifications and their properties," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 739–746, Oct. 1982.

[7] ——, "Theoretical results on the properties of median type filters," School Elec. Eng., Purdue Univ., West Lafayette, IN, Tech. Rep. EE 82-32, Sept. 1982.

J. Patrick Fitch was born in Baltimore, MD, on March 12, 1960. He received the B.S. degrees in physics and engineering science from Loyola College, Baltimore, MD, in 1981, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1984.

Since August 1984, he has been with the Engineering Research Division of Lawrence Livermore National Laboratory, Livermore, CA, where he is developing signal and image processing systems and techniques.

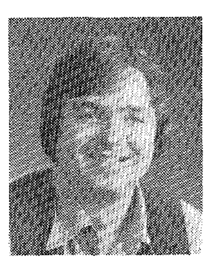
Dr. Fitch is a member of Sigma Pi Sigma, Eta Kappa Nu, Phi Kappa Phi, and Alpha Sigma Nu.



Edward J. Coyle was born in Philadelphia, PA, on April 22, 1956. He received the B.S. degree in electrical engineering from the University of Delaware, Newark, and the M.S. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 1980 and 1982, respectively.

Since August 1982, he has been an Assistant Professor of Electrical Engineering at Purdue University, West Lafayette, IN. His interests include computer networks and digital signal processing.

Prof. Coyle is a member of Phi Kappa Phi, Tau Beta Pi, and Eta Kappa Nu.



Neal C. Gallagher, Jr. (S'72–M'75) received the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 1974.

After being a member of the faculty of Case Western Reserve University, Cleveland, OH, he joined Purdue University, West Lafayette, IN, in 1976, where he is currently Professor of Electrical Engineering. He has consulted for government and industry in the areas of signal processing and holographic optics. His interests include numerical analysis, digital processing, source coding, and optical information processing.

Prof. Gallagher is Past President of the Central Indiana, Central Illinois, Chicago, and South Bend Section of the IEEE Information Theory Group.