Fault-tolerant sensor integration using multiresolution decomposition

L. Prasad, S. S. Iyengar, and R. L. Rao
Department of Computer Science, Louisiana State University, Baton Rouge, Louisiana 70803

R. L. Kashyap
School of Electrical Engineering, Purdue University, West Lafayette, Indiana 47907

(Received 7 May 1993)

Signal integration is an important aspect of many physical applications. It is often necessary to limit the effects of noise when data from several sensors are integrated to provide a consolidated estimate of some physical quantity being measured. This paper proposes a method of applying the idea of multiresolution to the problem of efficient integration of abstract sensor estimates when the number of sensors is very large and a large number of sensor faults are tame. The idea essentially consists of constructing a simple function from the outputs of the sensors in a cluster and resolving this function at various successively finer scales of resolution to isolate the region over which the correct sensors lie. We develop an optimal $O(N\log N)$ algorithm, where N is the total number of sensors, that implements this idea efficiently. This proposed application will result in speeding up computations involved in reducing the measure of the integrated output estimate by giving rise to an alternative method of narrowing down the region containing the correct value of the parameters being measured by the sensors.

PACS number(s): 02.90.+p

I. INTRODUCTION

Signal integration has been shown to have wide ranging applications in areas such as radar tracking and target detection. This includes the problem of fault-tolerant integration of information from multiple sensors, mapping, and modeling the environment space and task level complexity issues of the computational model. Further, these techniques have to be robust in the sense that even if some of the sensors are faulty, the integrated output should still be as reliable as possible.

We have proposed in this paper an alternative method of sensor integration using techniques of multiresolution decomposition. Multiresolution decomposition may be described as signal analysis in frequency channels of constant bandwidth on a logarithmic scale.

The cumulative signal from all the sensors is analyzed at various resolutions, starting from a coarse resolution and proceeding to successively finer scales. In a coarse-to-fine strategy, a minimum of detail necessary for recognition is processed. The approximation of a signal f at a resolution r is defined as an estimate of f derived by uniformly sampling f, r times per unit length. Tanimoto and Pavlidis [7] have developed efficient algorithms to compute the approximation of a function at different resolutions.

In this paper, we propose a method of applying the idea of multiresolution to the problem of fault-tolerant integration of abstract sensor estimates when the number of sensors is very large and a large number of sensor faults are tame.[1] The key idea is the construction of a simple function from the outputs of all the sensors and resolution of this function at various scales to isolate the region over which the correct sensors lie. We give an optimal algorithm which implements this idea efficiently.

The remainder of the paper is organized as follows. Section II describes the sensor integration in a distributed environment. Section III describes the multiresolution decomposition method. In Sec. III A we develop a technique of fault-tolerant sensor integration by applying the idea of multiresolution analysis to the overlap function $\mathcal{O}(x)$. Section III B describes the selection of robust peaks. In Sec. III C, we give an algorithm which implements our analytical method. The experimental results are given in Sec. IV. Then, we conclude this paper in Sec. V.

II. THE SENSOR INTEGRATION PROBLEM

A distributed sensor network consists of spatially distributed sensors that detect and quantify a certain phenomenon via its changing parameters. These readings are sent at regular intervals of time to processing units that integrate the readings from clusters of sensors and give outputs whose nature is much the same as the inputs of the individual sensors. Output from processors representing clusters of sensors are later integrated to get a complete picture of the spatially distributed phenomenon. However, before integration is performed at the processor level, it is necessary to have reliable estimates at each processor. Each sensor in a cluster measures the same parameter. It is possible that some of these sensors are faulty. Hence it is desirable to make use of this redundancy of the readings in the cluster to obtain a correct estimate of the parameters being observed. In short, a fault-tolerant technique of sensor integration is sought [1].

[1]Sensor faults are described in detail in Sec. II.

Marzullo [2] has addressed the problem of fault-tolerant integration of abstract interval estimates and has generalized his estimates to multidimensional sensors [3]. An illustration of this is given in Fig. 1. We [4] have proposed a method of obtaining sensor estimates with high reliability by considering the problem when the number of sensors was large and most of the faulty sensors were tamely faulty. We have also generalized our technique to multidimensional sensors in [5].

The technique we developed in [4] and [5] was a polling technique which computed the intersections of sensor outputs and the associated reliability measures. Since the number of these intersections is very large, the method is not useful for real-time applications.

In order to obtain a method of fault-tolerant sensor integration that is more feasible for real-time applications, we analyze a function called the *overlap function* introduced in [4]. We describe our approach to one-dimensional sensors although it generalizes easily to higher dimensions.

A. Preliminaries

We review the relevant definitions from [4] for convenience.

Definition 1: An *abstract sensor* is a sensor that reads a physical parameter and gives out an abstract interval-estimate I_s which is a bounded and connected subset of the real line R.

Definition 2: A *correct sensor* is an abstract sensor where the interval estimate contains the actual value of the parameter being measured. If the interval estimate does not contain the actual value of the parameter being measured, it is called a *faulty sensor.*

Definition 3: Let sensors $S_1, \ldots, S_N$ feed into a processor P. Let the abstract interval estimate of S_j be I_j $(1 \le j \le N)$, the closed interval $[a_i, b_i]$ with end points a and b. Define the *characteristic function* χ_j of the jth sensor S_j, $1 \le j \le N$ as follows:

$$\chi_j : R \to \{0,1\} \chi_j(x) = \begin{cases} 1, & \forall x \in I_j, \quad \forall 1 \le j \le N \\ 0, & \forall x \notin I_j, \quad \forall 1 \le j \le N \end{cases} . \tag{1}$$

Figure 2 illustrates the characteristic function of the interval $[a_i, b_i]$.

Definition 4: Let $\mathcal{O}(x) = \sum_{j=1}^{N} \chi_j(x)$ be the "overlap function" of the N abstract sensors. For each $x \in \mathbb{R}, \mathcal{O}(x)$ gives the number of sensor intervals in which x lies; that is, the number of intervals overlapping at the x.

Definition 5: A sensor is *tamely faulty* if it is a faulty sensor and if its output overlaps with that of a correct sensor.

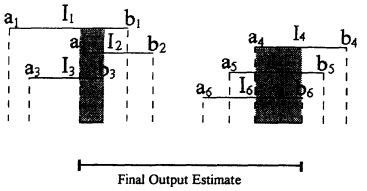

FIG. 1. Integration of interval estimates.

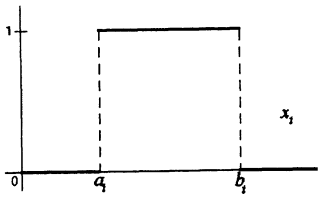

FIG. 2. Characteristic function of the interval $[a_i, b_i]$.

B. Comments on tame faults

If we tolerate at most f faults among N sensors, then by taking all $(N-f)$ intersections of the N sensor interval estimates, we are assured that the correct value of the parameter lies in one of these intersections. Marzullo [2] computes the integrated output as the smallest connected interval containing all the $(N-f)$ intersections. However, when the number of sensors is large and the number of faults cannot be strictly bounded, the $(N-f)$ intersections tend to be scattered widely over the real line, giving poor output estimates. In order to improve the output estimate in these cases, we must be able to further evaluate the $(N-f)$ intersections to choose the "best possible" intersection which contains the correct value with high reliability.

In the method proposed here we assume, as before, that the number of sensors is very large, that most faults are tame, and that there is no bound on the number of faults.

As the sensors are sampled synchronously at various time intervals, we order the sensors *a priori* by labeling them, dynamically maintain their overlap function $\mathcal{O}(x)$, and analyze it at various scales to obtain successively smaller regions which contain the correct value of the parameter observed.

The function $\mathcal{O}(x)$ is the sum of the characteristic functions of the abstract interval estimates. The value of $\mathcal{O}(x)$ at any point x is the number of intervals overlapping at the point x. The structure of the function is fairly simple.

Since there are finitely many sensors and each sensor is represented by an abstract interval estimate of bounded length, $\mathcal{O}(x)$ has compact support. By definition, $\mathcal{O}(x)$ is a non-negative function. It has several "crests" in its profile representing regions of maximal overlap of intervals (see Fig. 3).

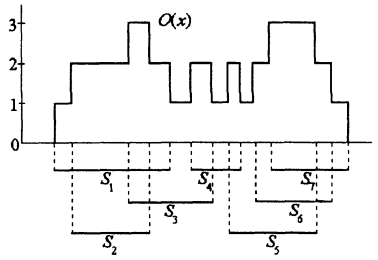

FIG. 3. The overlap function $\mathcal{O}(x)$ for a set of 7 sensors.

III. MULTIRESOLUTION DECOMPOSITION

Given a sequence of increasing resolutions $\{r_j\}_{j\in Z}$, the details of a function $f(x)$ at the resolution r_j are defined as the difference of information between the approximations of $f(x)$ at the resolution r_j+1 and the approximation at the resolution r_j.

A multiresolution representation also provides a simple hierarchical framework for interpreting the signal content. For instance, it is hard to recognize that a small rectangle inside an image is the window of a house, if we did not previously recognize the house "context." It is therefore natural to first analyze image details at a coarse resolution and then increase the resolution.

The approximation of a signal $f(x)$ at a resolution r is defined as an estimate of $f(x)$ derived from r measurements per unit length. These measurements are computed by uniformly sampling at a rate r the function $f(x)$ smoothed by a low-pass filter whose bandwidth is proportional to r. In order to be consistent when the resolution varies, these low-pass filters are derived from a unique scaling function which is dilated by the resolution factor r.

Starting at the coarsest resolution, we select those crests with the highest peaks (wavelet components with the largest amplitude) and choose the crest with the widest spread. At the next higher resolution, this crest is again inspected for crests within it with highest amplitudes and among these crests, the one with the widest spread is retained for similar analysis at the next resolution. This procedure results in isolating those regions of the real line over which $\mathcal{O}(x)$ has a maximum value, corresponding to high overlap degree. Figure 7 illustrates this procedure. $\mathcal{O}(x)$ in Fig. 6 is processed using multiresolution decomposition. The advantage of this procedure is significant from the point of view of computational speed, since the coarse-to-fine processing leads to elimination of large regions of the support of $\mathcal{O}(x)$ at each resolution.

We will rigorously formulate this heuristic in the next section in order to obtain a real-time algorithm that dynamically maintains $\mathcal{O}(x)$ and obtains the narrowed output estimate. The maintenance of $\mathcal{O}(x)$ requires $O(N\log N)$ time where N is the number of sensors. This follows directly from the fact that sensor intervals need to be sorted first according to their beginning and end points—a process that has a computational complexity of $\Omega(N\log N)$. We will verify that this method leads to results comparable to our earlier results and those of Marzullo by simulating sensor failures.

In our model of abstract sensors we assume that (i) a large number of sensor faults are tame, and that (ii) the length of each interval estimate is bounded below by l and above by L where $l<L$ and l, L are positive real numbers. Figure 4 describes the regions about the correct parameter value c where the faulty sensors cluster. A very large interval estimate is too inaccurate to be of any value and hence may be discarded. On the other hand, a very small interval estimate would not be amenable for fault-tolerance analysis. A minimum tolerance of $\pm l/2$ is built into the abstract sensors, and so we may assume that the width of each interval is at least l. These

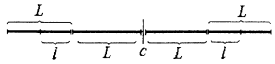

FIG. 4. The regions about the correct parameter value c where the faulty sensors cluster has $L=2l$.

two assumptions imply that the tame faults cluster in a bounded neighborhood around the correct value of the measured parameter. When the number of faulty sensors are significant, since most faults are tame, this results in overlaps of the faulty sensors amongst themselves and boosts the value of $\mathcal{O}(x)$ in the neighborhood of the correct value of the parameter, thus reinforcing the $(N-f)$ intersection containing the correct value.

Let T be the number of tamely faulty sensors. These may range in width from l to L. A tamely faulty sensor must intersect with a correct sensor. Therefore its end point nearest to the correct value c must lie within a distance of at most L from c. Thus at most $(1+\lfloor L/l\rfloor)$ tamely faulty sensors can be accommodated on either side of c with no two of them overlapping. That is, at least $\lceil T/2(1+\lfloor L/l\rfloor)\rceil$ tamely faulty sensors overlap over a region of width at least l within a distance of at most $2L$ from c.

When the number of intersections of tamely faulty sensors is $\lceil T/2(1+\lfloor L/l\rfloor)\rceil$, the width of this intersection is actually at least $2(l+L)$. When the number of intersections is T then this results in a peak with spread of at least l. This clustering reinforces the width and height of the "correct" $(N-f)$ intersection by adding in its neighborhood a peak of area Tl at least. In general, this results in a taller and wider peak in the neighborhood of c. The wildly faulty sensors, on the other hand, are random in their location on the real line and being uncorrelated, tend not to cluster in any small neighborhood. Thus the $(N-f)$ intersections resulting from them have shorter and narrower peaks representing them in $\mathcal{O}(x)$.

A. Multiresolution of the overlap function

If S_i $(1\le i\le N)$ are N abstract sensors with their interval estimates $[a_i,b_i]$ $(1\le i\le N)$ having characteristic function χ_i $(1\le i\le N)$ such that

$$\chi_i(x)=\begin{cases}1 & \text{if } x\in[a_i,b_j]\\ 0 & \text{if } x\notin[a_i,b_i]\end{cases}, \tag{2}$$

then the overlap function $\mathcal{O}(x)$ of these N sensors is given by $\mathcal{O}(x)=\sum_{i=1}^{N}\chi_i(x)$.

For each j, $\mathcal{O}(x)$ can be sampled at regular intervals $1/2^j$ to obtain the jth resolution of $\mathcal{O}(x)$ at scale $1/2^j$ as a linear combination of a set of functions obtained by scaling and translating a single function.

Let

$$\sigma(x)=\begin{cases}1 & \text{if } 0\le x\le 1\\ 0 & \text{otherwise}\end{cases} \tag{3}$$

(see Fig. 5.) Let $\alpha\in R$ and $j\in Z$. Consider the functions

$$\{\sigma(2^j(x-\alpha)-n\}_{n=-\infty}^{\infty} \text{ where } \sigma(2^j(x-\alpha)-n)=\begin{cases}1 & \text{if } \alpha+n/2^j\le x<\alpha+(n+1)/2^j\\ 0 & \text{otherwise .}\end{cases} \tag{4}$$

Without loss of generality, we may assume $0\le a<1/2^j$. Note that

$$[\alpha+n/2^j,\alpha+(n+1)/2^j)\cap[\alpha+(n+1)/2^j,\alpha+(n+2)/2^j)=\emptyset \tag{5}$$

and

$$\bigcup_{N=-\infty}^{\infty}[\alpha+N/2^j,\alpha+(N+1)/2^j]=R\ . \tag{6}$$

The jth resolution of $\mathcal{O}(x)$ with respect to the functions $\{\sigma(2^j(x-\alpha)-n)\}_{n=-\infty}^{\infty}$ is denoted by $\mathcal{O}_\alpha^j(x)$, and is given by

$$\mathcal{O}_\alpha^j(x)=\sum_{n=-\infty}^{\infty}\mathcal{O}(\alpha+2^{-j})\sigma(2^j(x-\alpha)-n)\ . \tag{7}$$

Since $\mathcal{O}(x)$ has compact support, the above summation is actually over finitely many n. If the interval estimates of the sensors S_i are $[a_i,b_i]$ $(1\le i\le N)$ and $a=\min_{1\le i\le N}\{a_i\}$ and $b=\max_{1\le i\le N}\{b_i\}$, then

$$\mathcal{O}_\alpha^j(x)=\sum_{n=\lfloor 2^j(a-\alpha)\rfloor}^{\lceil 2^j(b-\alpha)\rceil}\mathcal{O}(\alpha+n2^{-j})\sigma(2^j(x-\alpha)-n)\ , \tag{8}$$

where $\lceil x\rceil$ denotes the smallest integer greater than or equal to x. Thus $\mathcal{O}_\alpha^j(x)$ is obtained from $\mathcal{O}(x)$ by sampling $\mathcal{O}(x)$ at the points $\{\alpha+n2^{-j}\}_n$ (see Fig. 6). $\mathcal{O}_\alpha^j$ is a function whose features are of size $1/2^j$ or greater. To study the effect of sampling in the above manner it is sufficient to study the sampling of the characteristic function of an arbitrary interval $[a,b]$, since $\mathcal{O}(x)$ is a linear combination of characteristic functions.

Consider the test function

$$g(x)=\begin{cases}1 & \text{if } x\in[a,b]\\ 0 & \text{if } x\notin[a,b]\ .\end{cases} \tag{9}$$

Case (i). $b-a\ge 1/2^j$, i.e., $g(x)$ is a feature bigger than the scale width, $a\in[\alpha+n/2^j,\alpha+(n+1)/2^j)$ for some n, $b\in[\alpha+m/2^j,\alpha+(m+1)/2^j)$, and $n<m$. Thus

$$g_\alpha^j(x)=\sum_{r=-\infty}^{\infty}g(\alpha+r2^{-j})\sigma(2^j(x-\alpha)-r)=g(\alpha+n2^{-j})\sigma(2^j(x-\alpha)-n)+\cdots+g(\alpha+m2^{-j})\sigma(2^j(x-\alpha)-m)$$

if $a=\alpha+n/2^j$

or

$$g(\alpha+(n+1)2^{-j})\sigma(2^j(x-\alpha)-n-1)+\cdots+g(\alpha+m2^{-j})\sigma(2^j(x-\alpha)-m) \quad \text{if } a>\alpha+n/2^j\ .$$

Therefore we have

$$g_\alpha^j(x)=\begin{cases}1 & \text{if } \alpha+n/2^j\le x<\alpha+(m+1)/2^j\\ & \text{if } a=\alpha+n/2^j\\ 0 & \text{otherwise ,}\end{cases} \tag{10}$$

and

$$g_\alpha^j(x)=\begin{cases}1 & \text{if } \alpha+(n+1)/2^j\le x<\alpha+(m+1)/2^j\\ & \text{if } a>\alpha+n/2^j\\ 0 & \text{otherwise .}\end{cases} \tag{11}$$

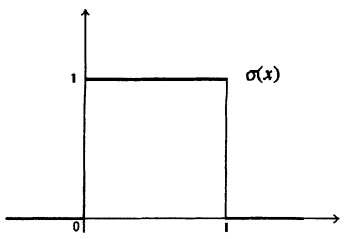

FIG. 5. Characteristics of scaling and translating a single function.

Thus there are two things which may happen independently to the support of $g(x)$: (i) it may shrink on the left by at most $1/2^j$, and (ii) it may extend on the right by at most $1/2^j$ (see Fig. 7).

We will see later that we have to correct for a positive shrinkage of the support of a feature, so as not to lose any information (correct value of the parameter measured). This is done by resolving over a region bigger than the one at hand by $1/2^j$ on the left. The extension ("smearing") of support will decrease with further resolution, and does not pose a problem.

Case (ii). $b-a<1/2^j$. [$g(x)$ is a feature smaller than the scale width]. If $a,b\in[\alpha+n/2^j,\alpha+(n+1)/2^j)$ for some n, then $g_\alpha^j(x)=0\forall x$, i.e., the feature will not appear at scale $1/2^j$. If $a\le 1/2j\le b$ for some n then

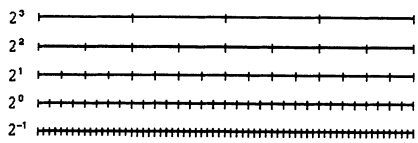

FIG. 6. Sampling at various scales.

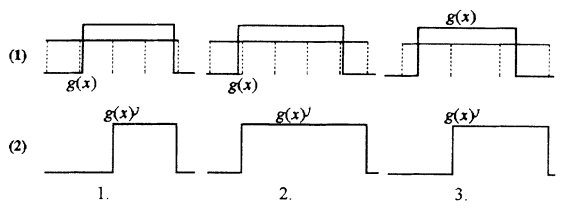

FIG. 7. Computational characterization of support of $g(x)$.

$$g_\alpha^j(x)=\begin{cases}1 & \text{if } \alpha+n/2^j\le x<\alpha+(n+1)/2^j\\ 0 & \text{otherwise},\end{cases} \tag{12}$$

that is, $g(x)$ will appear as a feature of size $1/2^j$ shifted to the right by at most $b-a$. This will diminish in size with further resolution, and $g(x)$ will be recovered by correction to the left and resolution.

We note that changing a will not produce any advantage insofar as sampling $\mathcal{O}(x)$ is concerned since location of the sampling $\{\alpha+n/2^j\}$ with respect to $\mathcal{O}(x)$ is arbitrary. We may thus conveniently let $\alpha=0$ and henceforth sample $\mathcal{O}(x)$ at points $\{n/2^j\}$ to obtain the jth resolution. Thus

$$\mathcal{O}^j(x)=\sum_{n=\lceil 2^j a\rceil}^{\lceil 2^j b\rceil}\mathcal{O}[n2^{-j}\sigma(x2^j-n)]j\in Z\ . \tag{13}$$

While considering resolution of $\mathcal{O}(x)$, we have to choose the scale appropriately. Too large a scale will provide no useful information about the structure of $\mathcal{O}$, while too small a scale would not isolate the features important to us, by bringing in unnecessary detail. Further, since each sensor has width at least l, it is desirable to start off with a scale smaller than (or the same order as) l, i.e., choose $j=\lceil\log_2(1/l)\rceil$. Thus each sensor will figure as a feature at least as big as $1/2^j$.

The fluctuations in $\mathcal{O}(x)$ occur at the points a_i,b_i $(1\le i\le N)$ which are the end points of the interval estimates. If a is the least of the a_i and b is the largest of the b_i, then the average number of fluctuations per unit length is given by $2N/(b-a)$. So in order to capture all the fluctuations we would have to resolve at least to a level $j>\log_2[2N/(b-a)]$.

B. Selection of robust peaks

At the jth level of resolution $\mathcal{O}_\alpha^j$ can be looked upon as a series of juxtaposed peaks. In other words, consider the sequence $\{\mathcal{O}(n/2^j)\}$. This sequence is a concatenation of several bitonic sequences, each of which increases first and then decreases (for details about bitonic sequences, see [6]). Each bitonic sequence which increases first and then decreases corresponds to a peak in $\mathcal{O}_\alpha^j$. We wish to isolate those peaks which are the tallest and have the widest spread, for it is in the region over which these peaks lie that the correct value of the parameter being measured is most likely to be found. Since the characteristic function of each sensor adds an area numerically equal to the sensor's width to the area under $\mathcal{O}(x)$, a good measure of the robustness of a peak is the area under it.

At the jth resolution consider the sequence $\{\mathcal{O}(n/2^j)\}_\alpha$. This is a finite sequence since the support of $\mathcal{O}$ is finite. Let there be p peaks (or p bitonic sequences) in $\mathcal{O}^j$. Thus the sequence $\{\mathcal{O}(n/2^j)\}$ can be rewritten as

$$\left\{\mathcal{O}\left[\frac{n_0}{2^j}\right],\mathcal{O}\left[\frac{n_0+1}{2^j}\right],\mathcal{O}\left[\frac{n_1}{2^j}\right],\mathcal{O}\left[\frac{n_1+1}{2^j}\right],\ldots,\mathcal{O}\left[\frac{n_{p-1}+1}{2^j}\right],\mathcal{O}\left[\frac{n_p}{2^j}\right]\right\}, \tag{14}$$

where the subsequence $\{\mathcal{O}[(n_{k-1}+1/)2^j],\ldots,\mathcal{O}(n_k/2^j)\}$ is the kth bitonic sequence from the left. Therefore the area under this peak is given by $1/2^j\sum_{n=n_{k-1}}^{n_k}\mathcal{O}(n/2^j)$. Since the factor $1/2^j$ is common to the areas of all peaks at the jth resolution, we may make the area "scale free" by dropping this factor and writing the area of the kth peak at level j as

$$\mathcal{A}^j(k)=\sum_{n=n_{k-1}}^{n_k}\mathcal{O}(n/2^j)\ . \tag{15}$$

We then select the peak with the largest area and ignore the other peaks. The function $\mathcal{O}$ is further resolved over the regions over which these largest peaks occur, and the process is repeated until a satisfactory region of the real line is isolated as the most likely candidate for containing the correct value of the parameter being measured by the sensors. However, before resolving a certain selected peak further, we correct the region over which the resolution is to be carried out by adding a segment of length $1/2^j$. Figures 8–12 show the property of the coarse grain to fine grain scheme for isolating robust peaks.

If at the jth resolution the kth peak is selected as the peak with the largest area under it, then the region over which the resolution of $\mathcal{O}(x)$ is performed again is $[(n_{k-1}+1)/2^j,n_k/2^j]$ with a correction of length $1/2^j$ at the left.

Therefore, $\mathcal{O}(x)$ is resolved over $[n_{k-1}/2^j,n_k/2^j]$. This process is continued until the interval to be resolved is smaller than the maximum acceptable width for output estimate, or when further resolution does not reduce interval width. The final corrected region with the largest value of $\mathcal{O}$ over it is accepted as the final output estimate.

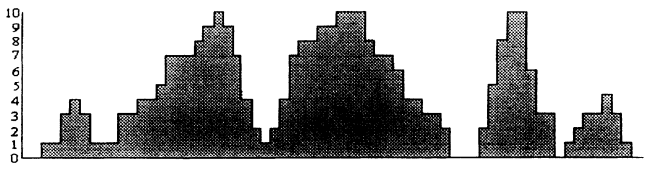

FIG. 8. $\mathcal{O}(x)$, shaded region indicates portion to be resolved.

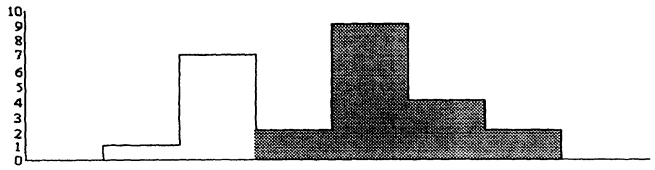

FIG. 9. $\mathcal{O}^{-2}(x)$, shaded region and region of width 4 at left to be resolved.

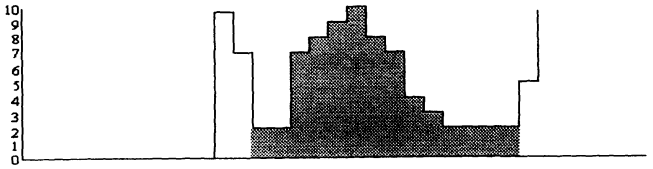

FIG. 11. $\mathcal{O}^{0}(x)$, shaded region and region of width 4 at left to be resolved.

C. Multiresolution algorithm

The inputs to the algorithm are the end points a_i, b_i of the interval estimate $[a_i,b_i]$ of the sensors S_i, $1 \le i \le N$, and the lower and upper bounds of resolution $j_0 = \lceil \log_2(1/l) \rceil$ and $j_1 > \lceil \log_2\{2N/\text{supp}[O(x)]\} \rceil$. The algorithm is shown in Fig. 13.

Procedure RESOLVE, which resolves $\mathcal{O}(x)$ to obtain an approximation of $\mathcal{O}$ at the jth resolution over a given interval, is shown in Fig. 14. It yields the indices of two points at the jth resolution, over which the largest or most prominent peak occurs, at the jth resolution. Figures 8–12 indicate the various stages during the execution of this algorithm graphically.

Procedure RESOLVE involves only scanning and hence is linear in the number of sensors N. Since the average density of fluctuation in $\mathcal{O}$ is $2N/\text{supp}[\mathcal{O}(x)]$, the level of resolution j required to capture almost all the measures of $\mathcal{O}$ is given by $j > \log_2\{2N/\text{supp}[O(x)]\}$.

If we assume that the parameter being measured by the sensors is known to lie between certain bounds, then $j > \log_2 N + C$ where C is some constant. Thus procedure RESOLVE will be called, on an average, $O(\log N)$ times. Hence the average computational complexity of the algorithm is $O(N \log N)$.

IV. EXPERIMENTAL VALIDATION

We have implemented the algorithm described in Sec. III C of this paper and have developed a parameter-driven simulator to evaluate the performance of the algorithm. Briefly, the simulator parses the input parameters which include the number of sensors, the numbers of tamely faulty and wildly faulty sensors, the upper and lower limits for the sensor interval widths, and the correct value. It then constructs randomly generated intervals for the sensors according to the scheme described in the following section. The overlap function is computed from a sorted list of the extreme points of all the sensors.

The overlap function is then sampled at increasing levels of resolution with the sampling frequency gradually increasing. The algorithm terminates when only one peak remains or when an arbitrarily fixed resolution is reached.

The overall complexity of the simulator is dominated by the sorting operation that needs to be performed over the endpoints of the sensor intervals. Thus, the computational complexity of the simulator is $O(N \log(N))$ where N is the total number of interval end points. This complexity is optimal in the sense that sorting $O(N)$ values cannot be completed in less than $O(N \log(N))$ time.

A. Construction of sensor intervals

The correct valued sensors (these intervals contain the correct physical value being estimated) are constructed as

$$s_i = L - \omega\eta, \quad e_i = s_i + \omega \tag{16}$$

where s_i is the start point of the interval, and e_i is the end point of the interval. ω is the width of the sensor interval estimate and is constrained to lie between $\omega_{\min}$ and $\omega_{\max}$. L in the equation above is the correct value of the physical quantity estimated. η is a random number uniformly distributed over (0.0,1.0]. For different sensors, ω is a random number uniformly distributed over $[\omega_{\min},\omega_{\max}]$. The intervals for tamely faulty sensors are constructed as

$$s_i = L + t_{\text{off}}\eta, \quad e_i = s_i + \omega \tag{17}$$

where t_{off}, a parameter of tame faults, denotes the critical distance from the correct value within which one of the end points of each tamely faulty sensor interval must lie.[2] Intervals for wildly faulty sensors are constructed as

$$s_i = L + t_{\text{off}} + \eta(L_{\max} - t_{\text{off}} - \omega), \quad e_i = s_i + \omega \tag{18}$$

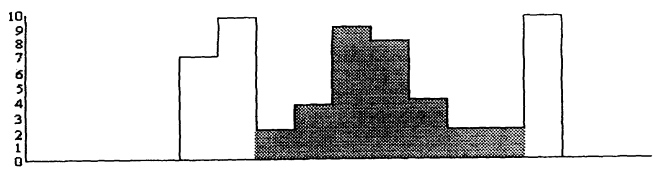

FIG. 10. $\mathcal{O}^{-1}(x)$, shaded region and region of width 4 at left to be resolved.

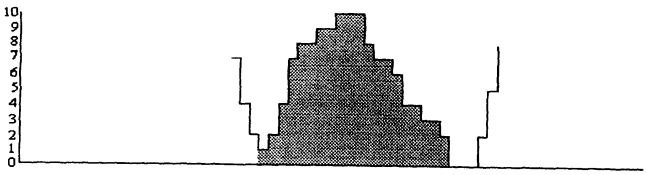

FIG. 12. $\mathcal{O}^{1}(x)$, shaded region and region of width $\frac{1}{2}$ at left to be resolved over. At this point, we may terminate resolution and choose the interval over which this peak attains a maximum as the final output.

[2]Both of the end points of wildly faulty sensor intervals must lie at a distance of at least t_{off} from L.

begin

1. Form the array of ordered pairs: $[(a_1,1),(b_1,-1),(a_2,1),(b2,-1),\ldots,(a_N,1),(b_N,-1)]$
2. Sort this array in increasing order with respect to the first components of the ordered pairs to obtain the array $[(\alpha_1,\sigma_1),(\alpha_2,\sigma_2),\ldots,(\alpha_{2N},\sigma_{2N})]$, where each α_i is some a_j or b_j, $\alpha_i, \leq \alpha_{i+1}1 \leq i \leq 2N$, and
$$\sigma_i = \begin{cases} 1 & \textit{if } \alpha_i \textit{ is an } a_j \\ 0 & \textit{if } \alpha_i \textit{ is a } b_j \end{cases}.$$
3. Construct the array $[(-\infty,0),(\alpha_1,\sigma_1),\ldots,(\alpha_i,\sum_{j=1}^{i}\sigma_j),\ldots,(\alpha_{2N},\sum_{j=1}^{2N}\sigma_j),(\infty,0)]$ representing the overlap function $O(x)$. Note that $\sigma_1 = +1, \sum_{j=1}^{i}\sigma_j = 0$, and $O(x) = \sum_{j=1}^{i}\sigma_j \ \forall\ x\ \alpha_i \leq x < \alpha_{i+1}$, $0 \leq i \leq 2N$ where $\alpha_0 = -\infty$ and $\alpha_{2N+1} = \infty$). Set $n_{j_0} = \lfloor 2^{j_0}\alpha_1 \rfloor$ and $n'_{j_0} = \lceil 2^{j_0}\alpha_{2N} \rceil$
4. **while** $j_0 \leq i < j_1$ **do** $RESOLVE[i; n_{i-1}, n'_{i-1}]$; $i \leftarrow i+1$; **enddo**
Sample $O(x)$ between $n_{j_1}/2^{j_1}$ and $n'_{j_1}/2^{j_1}$ to obtain $O^{j_1}(x)$, the approximation of $O(x)$ at the j-th resolution. Choose that subinterval of $[n_{j_1}/2^{j_1}, n'_{j_1}/2^{j_1}]$ over which $O^{j_1}(x)$ attains a maximum (or takes values greater than a specified value) and accept this subinterval as the integrated output estimate of the N sensor estimates

end.

FIG. 13. The multiresolution algorithm.

where $L_{\max}$ is the maximum value any sensor can estimate. Faulty sensor intervals (both wild and tame) are located randomly on both sides of the correct value L.

B. Computation and sampling of the overlap function

The overlap function is described in detail in Sec. III A. The array

$$\left[(-\infty,0),(\alpha_1,\sigma_1),\ldots,\left[\alpha_i,\sum_{j=1}^{i}\sigma_j\right],\ldots,\left[\alpha_{2N},\sum_{j=1}^{2N}\sigma_j\right],(\infty,0)\right]$$

represents the overlap function $O(x)$. In the above equation, α_i is an extreme point of some interval (either s_j or e_j for some j) such that $\alpha_i \leq \alpha_{i+1}$, $1 \leq i \leq 2N$, and σ_i takes the value one if α_i is a start point of some interval and takes the value negative one if α_i is an end point of some interval.

The overlap function is sampled at increasing sampling frequencies as described in Sec. III B. The simulation begins with the interval whose end points are $n_{j_0} = 2^{j_0}\alpha_1$ and $n'_{j_0} = 2^{j_0}\alpha_{2N}$. These end points are replaced by the end points of the peak with the largest area under it, found at the current sampling rate for resolution at the next higher level of resolution. The simulation then continues until no further refinement is possible or until a user-specified resolution is reached.

procedure $RESOLVE[j; n_{j-1}, n'_{j-1}]$
begin

1. Resolve $O(x)$ at scale 2^{-j} by sampling it over the interval $[(n_{j-1}-1)/2^{j-1}, n'_{j-1}/2^{j-1}]$ at the points $(2n_{j-1}-2)/2^j, (2n_{j-1}-1)/2^j, \ldots, 2n'_{j-1}/2^j$ to obtain $O^j(x)$, the approximation of $O(x)$ at the j-th resolution, represented by the array
$$[(-\infty,0),\ldots,(n/2^j,\sum_{j=1}^{k_n}\sigma_j),\ldots,(\infty,0)],\ (2n_{j-1}-2 \leq n \leq 2n'_{j-1})$$
and $O^j(x) = \sum_{j=1}^{k_n}\sigma_j$ where $n/2^j \leq x \leq (n+1)/2^j$ and $(n/2^j - \alpha_{k_n}) < 1/2^j$
2. Choose n_j and n'_j, where $n_j < n'_j$ and $2n_{j-1}-2 \leq n_j \leq 2n'_{j-1}$ such that $\{O(n/2^j\}_{n=n_j}^{n'_j}$ is a contiguous bitonic subsequence of $\{O(n/2^j)\}_{n=2n_{j-1}-2}^{2n'_{j-1}}$, which first increases and then decreases, and which has the largest sum i.e., the subsequence with the maximum sum among all such bitonic subsequences

end.

FIG. 14. The RESOLVE procedure used by the multiresolution algorithm.

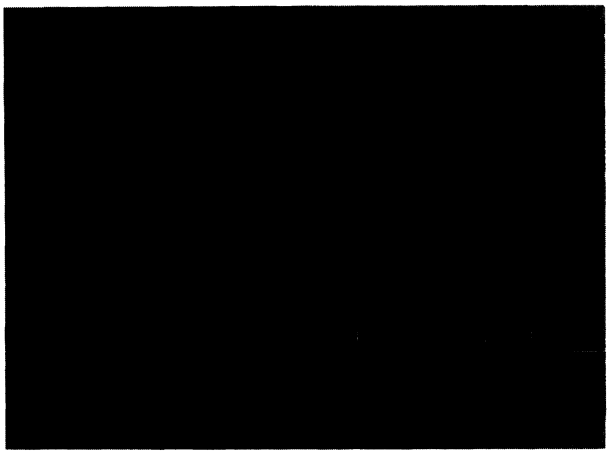

FIG. 15. 10 sensors, 5 wild faults, 3 tame faults.

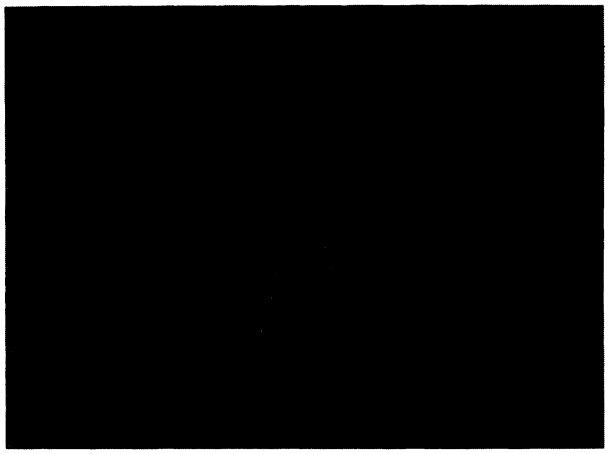

FIG. 17. 100 sensors, 10 wild faults, 10 tame faults.

Figures 15–19 show the results of the application of our algorithm to several instances of the sensor interval integration problem. The overlap function is shown as the whitest lines. The vertical line in the middle of each figure is the correct physical value being estimated by the sensors. The sampled overlap function is shown as dark grey lines. The peak of the sampled overlap function with the largest area under it, is shown delimited by the outer vertical lines. These lines move closer and closer at finer scales of resolution as seen in the figures. The actual sensor intervals are also shown in some figures overlaid with the overlap function; these appear in Fig. 16.

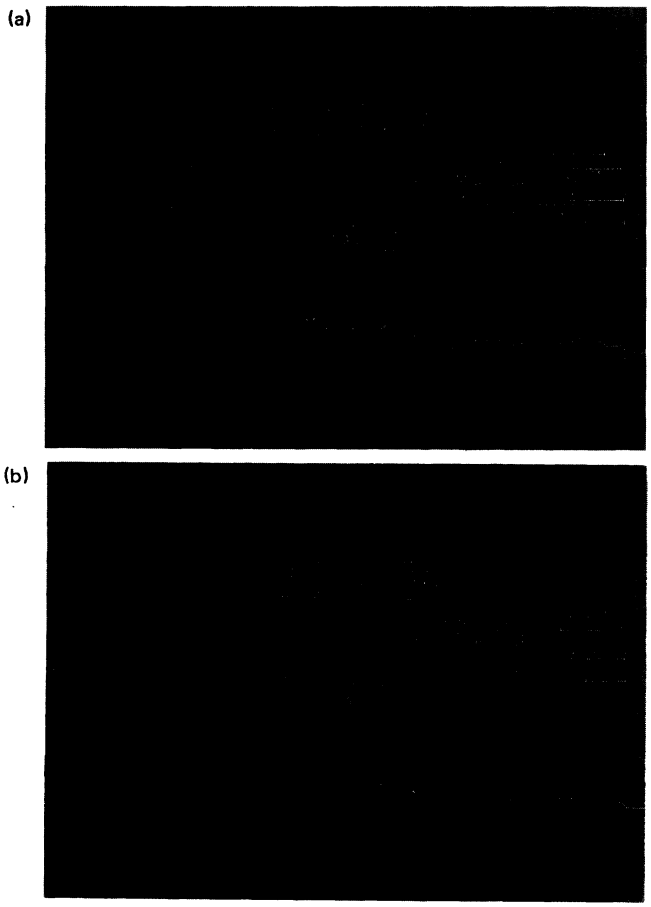

FIG. 16. (a) 100 sensors, 50 wild faults, 30 tame faults. (b) 100 sensors, 50 wild faults, 30 tame faults.

C. Simulation results

Tables I–IV show the regions of interest in the overlap function $\mathcal{O}(x)$ as the sampling frequency is varied from coarse to fine. Where a rise or fall of the sampled function is so sharp that it was completely missed at the current resolution level, the region of interest at the previous level of resolution is itself used at the next level. For example, in Table I, the three rows marked *medium* under sampling frequency exhibit this behavior.

Tables II and III both show the intermediate stages of resolution for 100 sensors but with different proportions of faulty and correct sensors. It is clear from these tables that as the proportion of correct sensors increases, the central peak (containing the correct physical value—see Figs. 15 and 16) sustains at different levels of resolution.

Figure 15 graphically shows the information summarized in Table II. As seen from the figure and the table, there are only two broad scales in the overlap function—a coarse scale (first two rows of table) and a fine scale. The correct sensor intervals all contain the correct value—the vertical center line. The tamely faulty sensor intervals are clustered at the top of the two figures located not too far from the correct value; the wildly faulty sensor intervals are located on either side of the correct value a significant distance away from it.

TABLE I. 200 sensors; 100 wildly faulty, 50 tamely faulty.

Sampling frequency	Resolution level	Sensor interval		
		Start point	End point	Width
coarse	3	−0.449 277	0.673 841	1.123 118
medium	4	−0.191 211	0.128 103	0.319 314
medium	5	−0.191 211	0.128 103	0.319 314
medium	6	−0.191 211	0.128 103	0.319 314
fine	7	−0.022 714	0.003 010	0.025 724

TABLE II. 100 sensors; 50 wildly faulty, 30 tamely faulty.

Sampling frequency	Resolution level	Sensor interval Start point	End point	Width
coarse	3	−0.675 956	0.663 665	1.339 621
medium	4	−0.675 956	0.663 665	1.339 621
medium	5	−0.115 485	0.074 100	0.189 585
fine	6	−0.115 485	0.074 100	0.189 585

TABLE III. 100 sensors; 10 wildly faulty, 10 tamely faulty.

Sampling frequency	Resolution level	Sensor interval Start point	End point	Width
coarse	3	−0.627 078	0.379 362	1.006 440
medium	4	−0.627 078	0.379 362	1.006 440
medium	5	−0.627 078	0.379 362	1.006 440
fine	6	−0.627 078	0.379 362	1.006 440

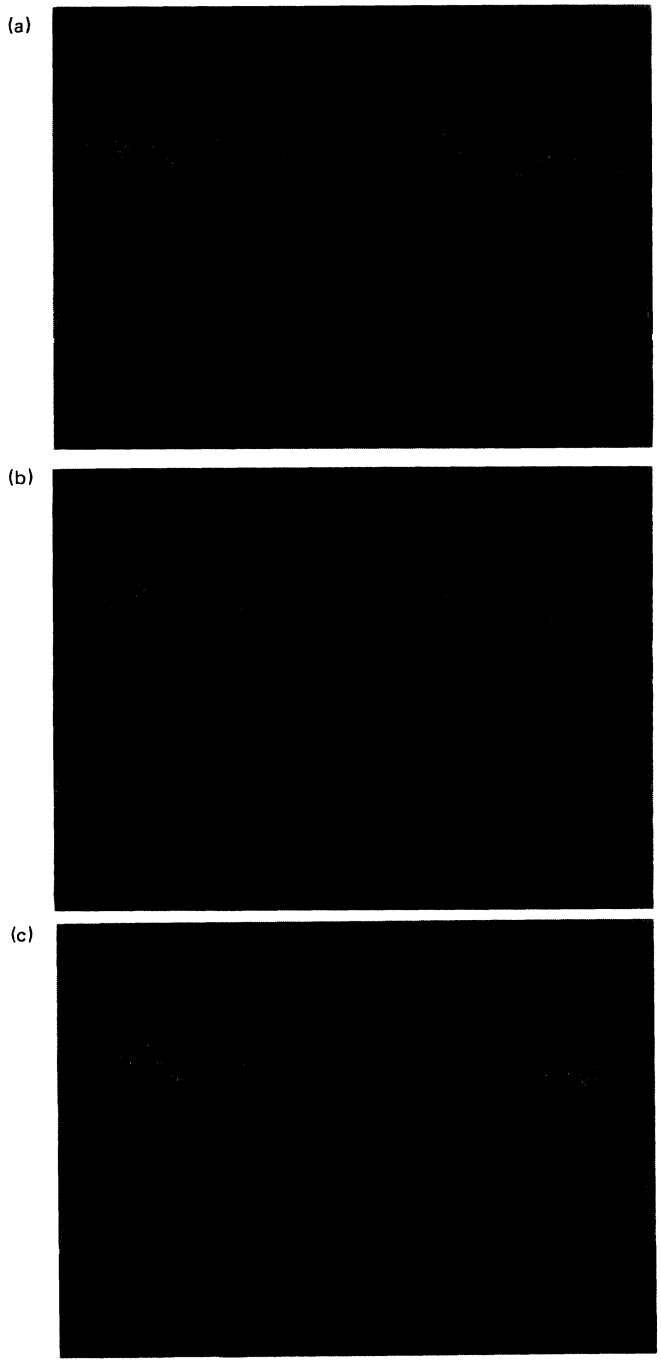

FIG. 18. (a) 1000 sensors, 800 wild faults, 100 tame faults. (b) 1000 sensors, 800 wild faults, 100 tame faults. (c) 1000 sensors, 800 wild faults, 100 tame faults.

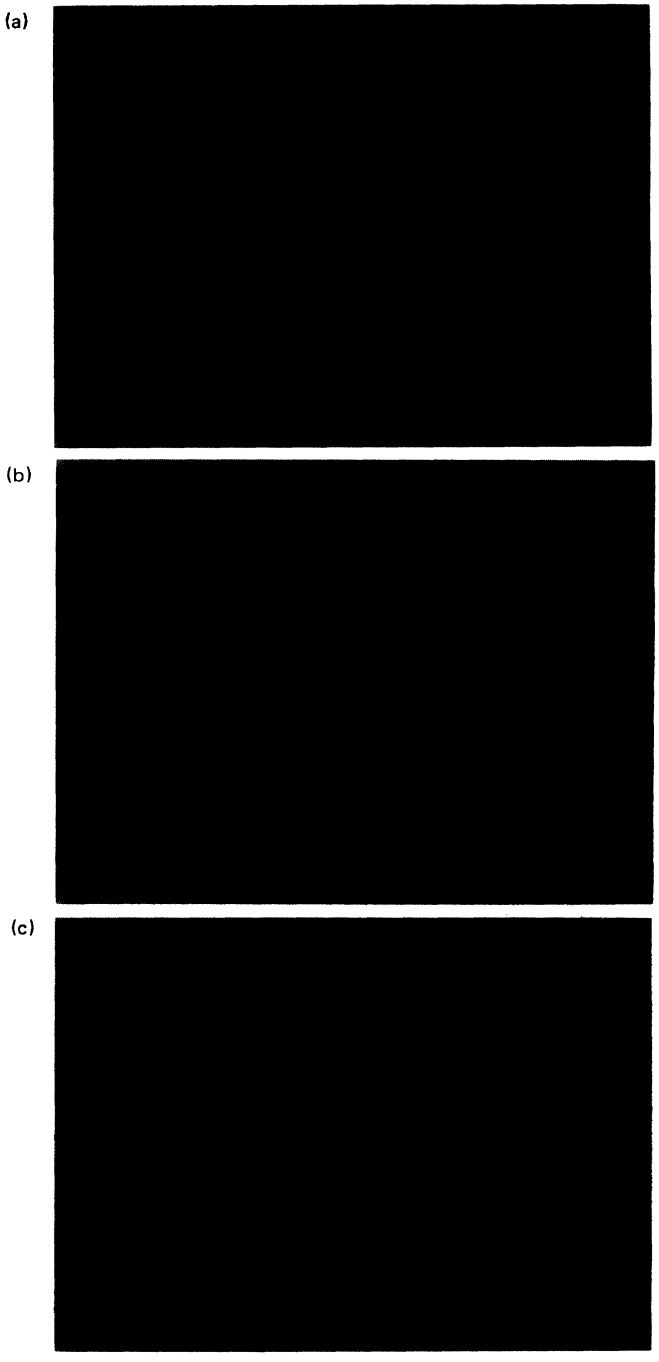

FIG. 19. (a) 200 sensors, 100 wild faults, 50 tame faults. (b) 200 sensors, 100 wild faults, 50 tame faults. (c) 200 sensors, 100 wild faults, 50 tame faults.

TABLE IV. 1000 sensors; 800 wildly faulty, 100 tamely faulty.

Sampling frequency	Resolution level	Sensor interval Start point	End point	Width
coarse	3	0.252 377	1.006 563	0.754 186
coarse	4	0.252 377	1.006 563	0.754 186
medium	5	−0.047 790	0.203 101	0.250 891
medium	6	−0.047 790	0.203 101	0.250 891
medium	7	−0.016 398	0.021 779	0.038 177
fine	8	−0.012 628	−0.006 013	0.006 615
fine	9	−0.012 628	−0.006 013	0.006 615

As an illustration of the extreme case behavior of our algorithm, we include two figures (16 and 17). Figure 16 shows the overlap function corresponding to 100 sensors; 10 wildly faulty and 10 tamely faulty (see Table III for data). Notice that the most significant peak is sustained at every intermediate level of resolution—a phenomenon that faithfully repeats whenever the fraction of correct valued sensors among all the sensors is relatively high.

What happens when most of the sensors are faulty is illustrated in Fig. 17. Table IV contains the results of the simulation with 1000 sensors. Notice that the final interval computed by the algorithm does not include the correct value due to the very large fraction of wildly faulty sensors. It is relevant to mention one *caveat* here. The actual form of the final solution is also dependent upon the seed of the random number generator used to start the *Monte Carlo* simulation. Thus, it is not unusual for the final solution of one particular simulation experiment, with almost all of the sensors being faulty, to contain the correct value.

Figure 18 shows the sampled overlap function, at various stages of resolution, corresponding to the data in Table I. Figure 18(b) shows that the region of the overlap function sampled does not change throughout the medium frequency range as also evidenced by the three rows in Table I marked *medium* under the sampling frequency column.

V. CONCLUDING REMARKS

We have applied the concept of multiresolution to one-dimensional sensor estimates to obtain a fault-tolerant integrated estimate of the parameter being measured by the sensors. This is done by computing the overlap function of the sensors, and resolving this function at increasingly finer dyadic scales to obtain a sequence of functions, each of which consists of a series of peaks. In each of these functions, the highest peak with the largest spread is chosen and further resolved at a finer scale to obtain the next function. This process is repeated finitely many times up to a certain scale and the sensor estimate is taken to be the region over which the maximum value of the largest peak is attained.

This method helps isolate the neighborhood of the correct value of the parameter being measured by taking advantage of the fact that the maximum intersection of intervals occurs about the correct value, and this corresponds to the highest peaks, and further, the tamely faulty sensors cluster around the correct value, contributing to the height and spread of the peak under which the correct value lies. At each resolution, only the relevant details of $\mathcal{O}(x)$ are considered, increasing the efficiency of the computation.

The underlying idea in our method, the recognition and isolation of the most prominent and robust peaks in a region and the consequent elimination of narrower and less prominent peaks as "errors," can indeed be used elsewhere to isolate the important characteristics of a signal and remove the "noise" in a computationally efficient manner. This method can be generalized, with some modifications, to multidimensional sensors and signals.

ACKNOWLEDGMENTS

The authors wish to thank Dr. R. N. Madan for his comments and suggestions during the period of this research. This research was supported in part by Louisiana Board of Regents Grant No. LEQSF-RD-A-04, 1990, and by ONR Grant No. N000014-91-J1306.

[1] J. Crowley, Technical Report, CMU-R1-TR-82-7, Robotic Inst., CMU, 1987 (unpublished).
[2] K. Marzullo, ACM Trans. Comput. Syst. **8**, 284 (1990).
[3] P. Chew and K. Marzullo (unpublished).
[4] L. Prasad, S. S. Iyengar, R. L. Kashyap, and R. N. Madan, IEEE Trans. Syst. Man Cybernet. **SMC-21**, 1082 (1991).
[5] L. Prasad and S. S. Iyengar (unpublished).
[6] M. J. Quinn, *Designing Efficient Algorithms for Parallel Computers* (McGraw-Hill, New York, 1987).
[7] S. L. Tanimoto and T. Pavlidis, Comput. Vision Graphics Image Processing **4**, 104 (1975).

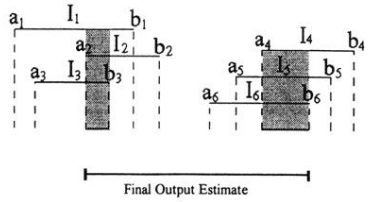

FIG. 1. Integration of interval estimates.

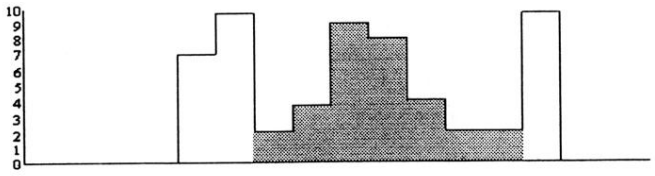

FIG. 10. $\mathcal{O}^{-1}(x)$, shaded region and region of width 4 at left to be resolved.

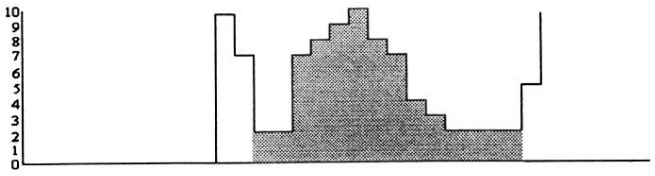

FIG. 11. $\mathcal{O}^0(x)$, shaded region and region of width 4 at left to be resolved.

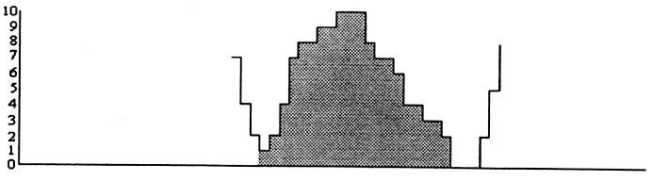

FIG. 12. $\mathcal{O}^1(x)$, shaded region and region of width $\frac{1}{2}$ at left to be resolved over. At this point, we may terminate resolution and choose the interval over which this peak attains a maximum as the final output.

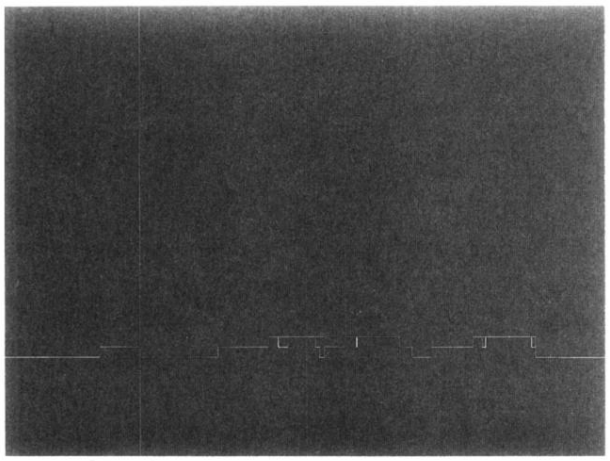

FIG. 15. 10 sensors, 5 wild faults, 3 tame faults.

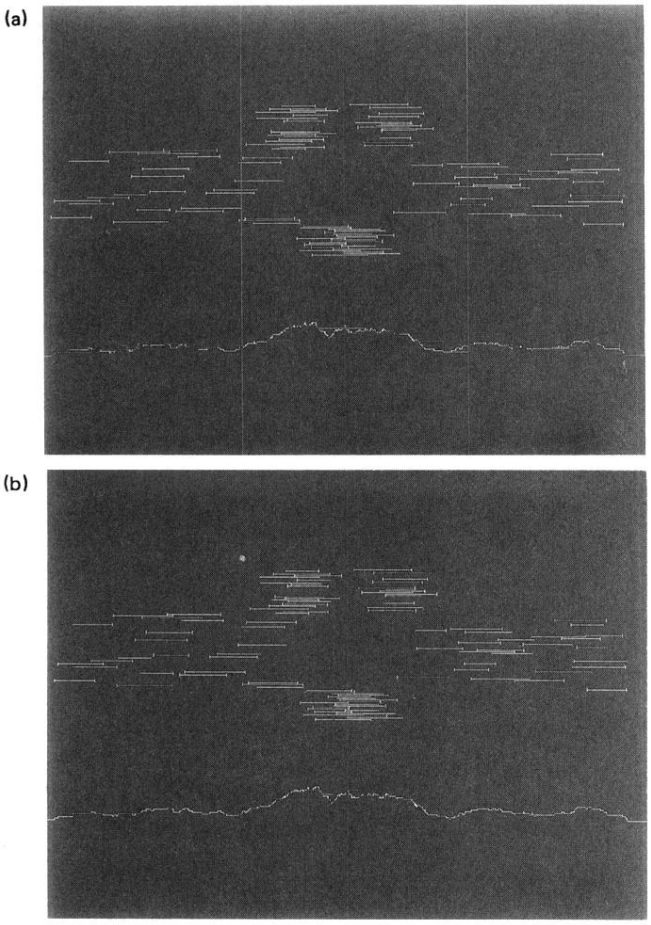

FIG. 16. (a) 100 sensors, 50 wild faults, 30 tame faults. (b) 100 sensors, 50 wild faults, 30 tame faults.

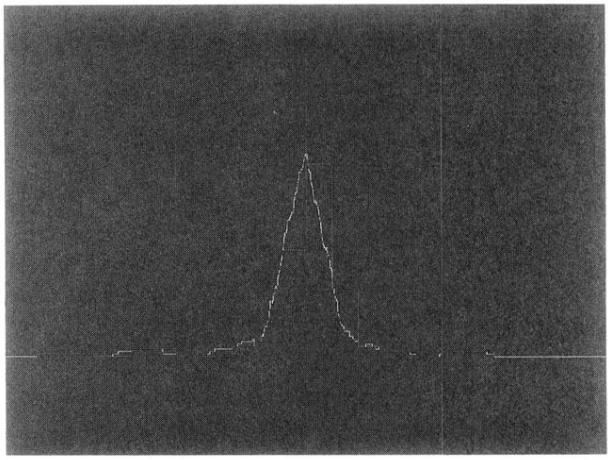

FIG. 17. 100 sensors, 10 wild faults, 10 tame faults.

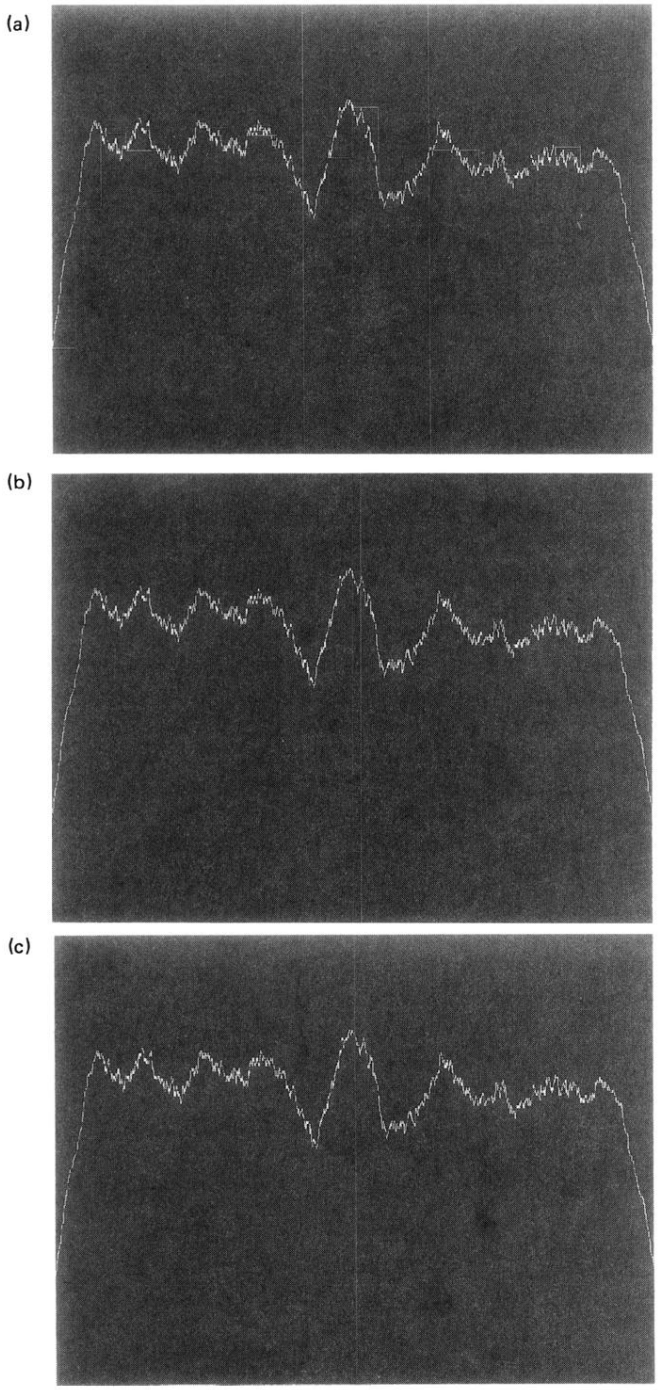

FIG. 18. (a) 1000 sensors, 800 wild faults, 100 tame faults. (b) 1000 sensors, 800 wild faults, 100 tame faults. (c) 1000 sensors, 800 wild faults, 100 tame faults.

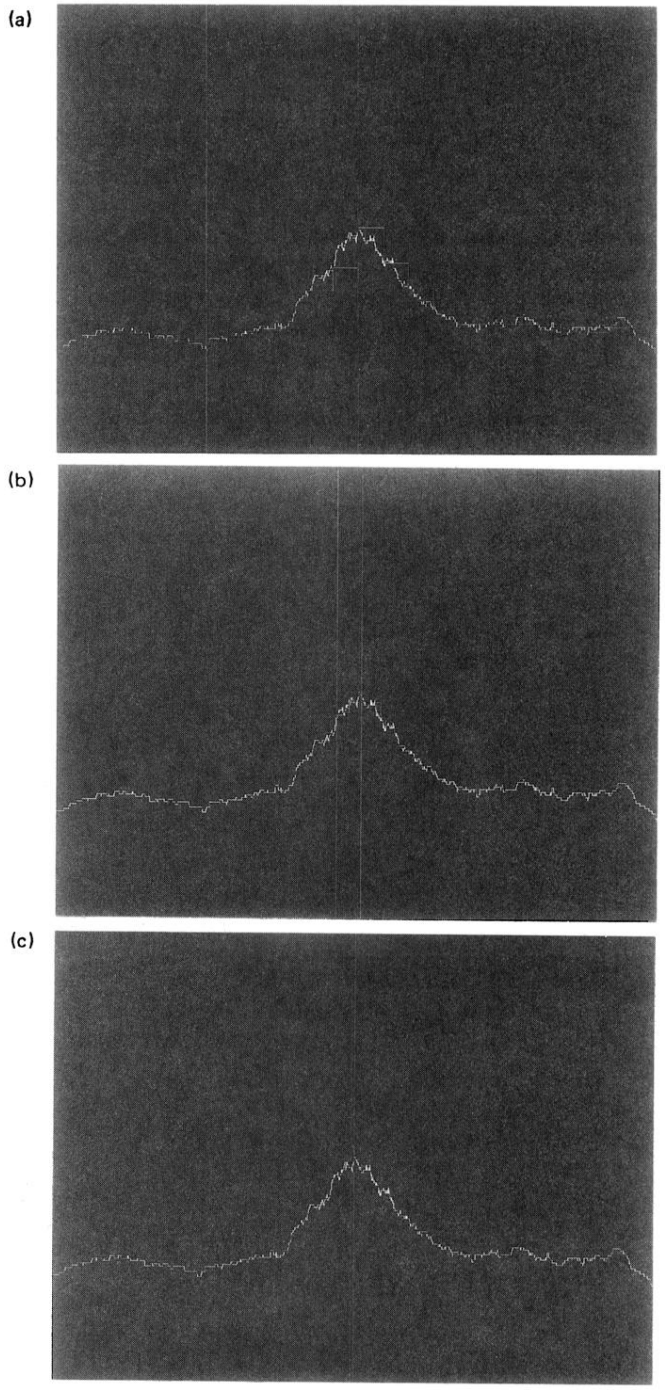

FIG. 19. (a) 200 sensors, 100 wild faults, 50 tame faults. (b) 200 sensors, 100 wild faults, 50 tame faults. (c) 200 sensors, 100 wild faults, 50 tame faults.

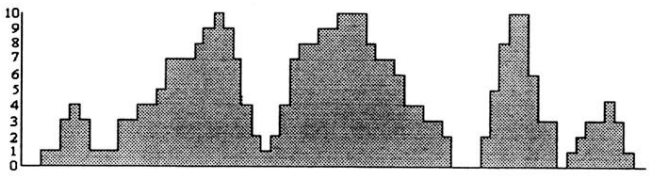

FIG. 8. $\mathcal{O}(x)$, shaded region indicates portion to be resolved.

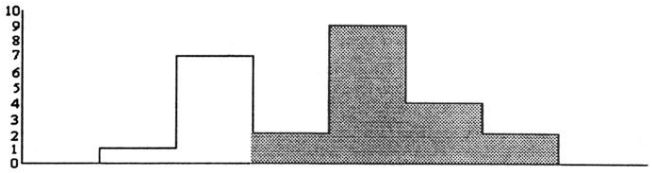

FIG. 9. $\mathcal{O}^{-2}(x)$, shaded region and region of width 4 at left to be resolved.