Weighted D-Filtering

Wen-Rong Wu and Amlan Kundu

***Abstract*—In this paper, we have proposed a new type of filter which has the most desirable properties of an image smoothing filter. These properties are 1) robust smoothing efficiency, 2) edge preservation, and 3) thin-line detail preservation. The new filter computes its output as the median of weighted averages, instead of plain averages as used in the Hodges–Lehman D-filter, of symmetrically placed order statistics. One particular weighting scheme is considered in details for experiments. The experimental and comparison results are included verifying the useful properties of the proposed filter.**

I. Introduction

Image smoothing and restoration have wide applications in image processing and robot vision [1]. Median filtering as proposed in [2] has been widely applied to image enhancement [3]–[5]. The median filter, however, has relatively poor smoothing efficiency, and it cannot preserve thin line details (TLD). Various approaches based on robust estimation theory and rank estimation are proposed [6], [7] to achieve higher smoothing efficiency without sacrificing the edge preserving characteristics of the median filter. A number of other filters such as multilevel median filter [1], [8], FIR hybrid median filter [1], etc., are proposed to preserve TLD of the image.

In this paper, our objective is to design a filter that has three very important characteristics: 1) good noise smoothing efficiency; 2) good edge preservation; and 3) good TLD preservation. The new filter, called the weighted D-filter, is considered in the next section. The simulation results and the conclusions are described in Section III.

II. Weighted D-Filter

It is known that the characteristic of the central pixel is very important for edge and detail preservation. The filter should also have good smoothing efficiency over the flat regions of the image. In the following, we propose a general scheme for designing a robust edge and detail preserving filter.

1) Start with a sliding window encompassing a datalength n and a robust noise smoothing filter. The sliding window is not necessarily a square window. In the absence of edges or details, such a filter gives very good smoothing of the noise.
2) Design a subsample that selects only those pixels, from the pixels of the sliding window, whose gray values are in some neighborhood of the central pixel. Over a flat region, such a subsample should substantially include all the pixels of the window.
3) Apply the filter of Step 1 to the pixels of the window with appropriately more weights to the pixels of the subsample.

Design of Subsample

The subsample should select all the pixels whose gray values are "close" to the central pixel. One possible design is given below.

Manuscript received June 8, 1989; revised October 31, 1990. This paper was recommended by Associate Editor E. J. Coyle.

W.-R. Wu is with the Microelectronic and Information Science and Technology Research Center, National Chiao-Tung University, Taiwan, ROC.

A. Kundu is with the Department of Electrical Engineering, State University of New York, Buffalo, NY 14260.

IEEE Log Number 9042416.

Pick the central pixel value X_0 of an $n \times n$ region. Then select every X_i of the region in the range $[X_0 - q \leqslant X_i \leqslant X_0 + q]$ as part of the subsample. The parameter q is chosen for the optimal performance of the algorithm, and is closely related to σ, the noise standard deviation.

Since for TLD, X_0 belongs to the "thin line," such subsampling with the proper choice of q only picks the pixels on the "thin line." When an edge passes through the window, X_0 belongs to the edge. Thus the relationship $X_0 - q \leqslant X_i \leqslant X_0 + q$ is likely to be satisfied by the pixels belonging to the edge. There is one major drawback in the subsampling scheme discussed above. What will happen if the central pixel is corrupted by an outlier? To safeguard against such cases one needs to estimate X_0 by means of a robust estimate. This makes the design of subsample very much complicated. An easier but less accurate approach is to have a default option. Whenever X_0 is corrupted by an outlier, the filter should behave as a plain smoothing filter as required by the flat regions of the image. This is reasonable as the flat regions are more common than the edges or the thin-line signals in a natural image. In the following, we propose a filtering scheme that incorporates the subsampling scheme and is based on Hodges–Lehman D-filter [10]. This filter has an in-built tolerance to such outlier presence [10].

Filter Design

Step 1: We choose the D-filter [10] as described next. The noise smoothing and the outlier tolerance properties of D-filter are described in [10].

Hodges–Lehman D-Filter [10]: Let X_i; $1 \leqslant i \leqslant n$ be a sample from a population with distribution $F(x,\theta)$ and density $f(x,\theta)$ where $f(\cdot)$ is symmetric about zero, continuous, and strictly positive on the convex support of $F \cdot [x\colon 0 < F(x) \leqslant 1]$. Denote $X_{(1)}, \cdots, X_{(n)}$ as the order statistics of the sample. Let $n = 2m$ or $2m-1$. In either case, we define

$$D_n(x_1,\cdots,x_n) = \underset{1 \leqslant i \leqslant m}{\text{median}} \left(X_{(i)} + X_{(n-i+1)} \right) \cdot 1/2. \quad (1)$$

We observe the usual convention of letting D_n be the midpoint of the interval of medians, if there is an ambiguity. D-filtering as given by (1) assigns the same weight to both the symmetrically placed order statistics. However, one of the symmetrically placed order statistics could fall in the subsample range, while the other may not. Assignment of equal weight in such cases is responsible for edge smearing and destruction of thin line details. In the proposed design, the weighting scheme is changed.

Step 2—Subsample Design: We choose the subsample design scheme as described before.

Step 3: In this step, the filter of Step 1 needs to be modified to assign more weights to the pixels in the subsample. Consider a generalized version of the D-filter, called the weighted D-filter, as given below.

Weighted D-Filter:

$$D = \underset{i}{\text{median}} \left(X_{(i)} \cdot w(i) + X_{(n-i+1)} \cdot u(n-i+1) \right);$$

$$w(i) + w(n-i+1) = 1 \quad (2a)$$

where $0 \leqslant w(i), w(n-i+1) \leqslant 1$.

For the D-filter $w(i) = w(n-i+1) = 0.5$. For the weighted D-filter, $w(i)$ and $w(n-i+1)$ can have any value in the range $0-1$. It is intuitively clear that $w(i)$ should have a small value (close to zero) when $X_{(i)}$ does not belong to the subsample; and $X_{(n-i+1)}$ belongs to the subsample. Also, in this case $w(n-i+1)$ should have a large value close to one. On the other hand, if both $X_{(i)}$ and $X_{(n-i+1)}$ belong to the subsample, both $w(i)$ and

$w(n-i+1)$ should have equal weight of 1/2. So, we choose

$$w(i)=1/2; \quad w(n-i+1)=1/2 \quad \text{when } R_1=R_2=1 \text{ or } R_1=R_2=0 \tag{2b}$$

$$w(i)=1; \quad w(n-i+1)=0 \quad \text{when } R_1=1 \text{ and } R_2=0 \tag{2c}$$

$$w(i)=0; \quad w(n-i+1)=1 \quad \text{when } R_1=0 \text{ and } R_2=1. \tag{2d}$$

Here, R_i's are defined as logical variables given by the following equations.

$$R_1=1 \quad \text{when } X_0-q \leqslant X_{(i)} \leqslant X_0+q, \quad \text{otherwise } R_1=0 \tag{3a}$$

$$R_2=1 \quad \text{when } X_0-q \leqslant X_{(n-i+1)} \leqslant X_0=q, \quad \text{otherwise } R_2=0. \tag{3b}$$

It is clear from (2c) and (2d) that, in the presence of edge or TLD, the observations inside the subsample are given the most weight (one). For the slowly varying region or quasi-constant region, the weighting scheme is given by (2b), as the subsample range is likely to contain all the data. The filtering in this case is plain D-filtering as desired. When the central pixel is corrupted by an impulse, the other weighting scheme given by (2b) will be effective. In this case, most of the observations will be outside the subsample. The weighting scheme ensures that, in this extreme case, the output is essentially the plain D-filter output, and the impulse is rejected. This is the so-called default option, which is reasonable as most of the image consists of quasi-constant or slowly varying regions.

Window Selection: For the weighted D-filter to be effective both in edge and detail preservation, the proper choice of window is absolutely essential. We first choose an $m \times m$ square window. There are four diagonals that can be drawn through the central pixel of this window. For the $m \times m$ window, all the pixels on these four diagonals are selected for filtering. Thus the window selection is similar to that of the multilevel median filter [8].

III. Experimental Results and Conclusions

First, we evaluate the filter for edge preservation. For this purpose, we propose a quantitative criterion for edge preservation.

Definition: Let a sample of size n consist of two samples of size n_1 and n_2 with distributions $f(\cdot,\theta_1)$ and $f(\cdot,\theta_2)$, respectively, such that $n=n_1+n_2$, and $|\theta_1-\theta_2|>3\sigma$. The distribution $f(\cdot)$ is considered to be symmetric and medium-tailed with standard deviation σ. An estimate is called practically edge preserving if it estimates θ_1 when $n_1>n_2$; and

$$\left(\frac{1}{N}\sum_{i=1}^{i=N}\left(\theta_1-\theta_1'(i)\right)^2\right)^{1/2}<\sigma \tag{4}$$

where $\theta_1'(i)$ is the estimate of θ_1 for the ith experiment, and N is the number of times the experiment is carried out. Naturally, N should be large ($\geqslant 25$). In a recent paper, Peterson *et al.* [9] essentially used the same concept to evaluate the edge preserving nature of certain filters. Their method is based on numerical evaluation of the filter output distribution when the input is an edge signal as assumed in our definition. Our definition, though not based on such information as output distribution of the filters, qualitatively leads to the same answer.

Fig. 1 shows the edge preserving characteristic of the weighted D-filter. For the experiment reported in Fig. 1, $n=25$, $N=25$, $\theta_1=100$, $\theta_2=180$, and $\sigma=16$. The effect of parameter q is clearly evident from Fig. 1. For values of q in the range $3\sigma-\sigma$, the filter is edge preserving.

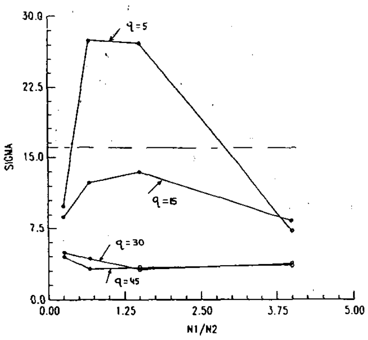

Fig. 1. Experimental results on the edge preserving characteristic of the weighted D-filter.

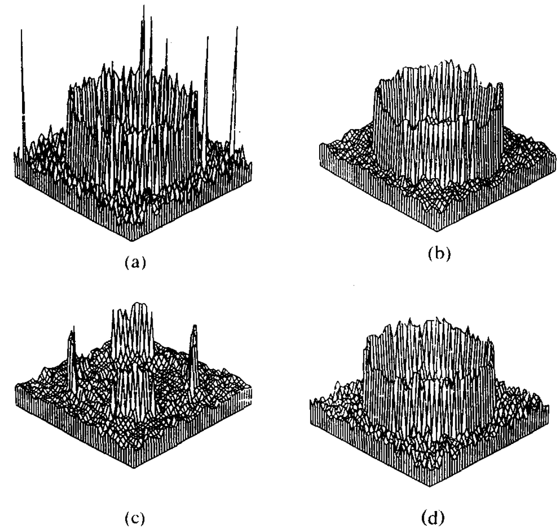

Fig. 2. (a) Synthetic thin ring image with 1% impulse corruption and zero mean additive Gaussian noise. (b) Weighted D-filtered image. (c) Median filtered version of Fig. 2(a). (d) Multilevel median filtered version of Fig. 2(a).

Next, our focus is on TLD preservation. We propose the following quantitative measures, denoted as M_1, M_2, and M_3, for TLD preservation.

Consider a thin ring image as shown in Fig. 2. Let the thin ring image be $s(i,j)$ and the filtered (or corrupted) image be $r(i,j)$. Indicate the set of (i,j), which belongs to the thin ring, by R and the set of the neighborhood of $((i,j)\in R)$ by S. Or,

$$S=\{(i,j),\|(i,j)-(i',j')\|\leqslant\beta,\ \text{where}(i',j')\in R,\ \text{and } \|\cdot\| \text{ is the Euclidean distance}\}.$$

The parameter β in S should be chosen small enough to include only the immediate neighborhood of R. The measures M_1, M_2, and M_3 are now defined as follows:

$$M_1=\frac{\sum_{(i,j)\in R}|r(i,j)-s(i,j)|}{N} \tag{5}$$

where N is the number of points in R. M_1 measures the average absolute deviation of the intensity of the filtered (or

TABLE I
COMPARISON OF M_1, M_2, AND M_3 FOR DIFFERENT FILTERS; $\alpha = 8$, $\beta = 3$

	M_1	M_2	M_3
Noisy	6.355	0.6842	104.9
MLM	5.836	0.7105	40.02
Weighted-*D*	4.631	0.8289	18.78
Median	28.24	0.1711	265.4

corrupted) thin line signal with respect to the original. For ideal filter performance, M_1 should be zero.

$$M_2 = \frac{\sum_{(i,j)\in R} I(|r(i,j) - s(i,j)|)}{N} \tag{6a}$$

where $I(\cdot)$ is an indicator function defined as

$$I(x) = \begin{cases} 1, & \text{if } x \leqslant \alpha \\ 0, & \text{otherwise.} \end{cases} \tag{6b}$$

This measure gives the fraction of pixels in the filtered thin ring with intensity values within a prescribed range from the original. Ideally, this measure should be one. For the computation of this measure, the parameter α should be chosen in the vicinity of σ, the standard deviation of the additive noise.

$$M_3 = \frac{\sum_{(i,j)\in S} (r(i,j) - s(i,j))^2}{L} \tag{7}$$

where L is the number of points in S. M_3 is the measure for average squared error over a small neighborhood encompassing the thin ring. Ideally, this measure should be zero. If the filtering process cannot reconstruct the thin ring at its original position, this measure is likely to have very high values.

Fig. 2(a) shows the noise-corrupted synthetic image of a one-pixel wide thin edge in the shape of a disk. The height of the uncorrupted thin ring is 50. The additive noise is white Gaussian with mean 0 and standard deviation 8. The image size is 50×50. The weighted D-filtered version is given by Fig. 2(b). The filtering is done using a 3×3 window in a raster scan fashion with parameter $q = 24$. Fig. 2(c) and (d) show the median and MLM filtered version of Fig. 2(a), respectively. A 3×3 window is used for the median filter, and a 5×5 window is used for the MLM filter. Comparing Fig. 2(b) with Fig. 2(d), it is easy to see that the weighted D-filter is as effective as the MLM filter in preserving thin-line details. Also, as shown in Fig. 2(c), the performance of the median in relation to TLD preservation is very poor. The measures M_1, M_2, and M_3 for Fig. 2(a)–(d) are given in Table I.

It is clear from the table that the median filter cannot preserve thin line structures. The multilevel median [8] and the weighted D-filter can substantially preserve the thin line details as they both pay attention to the characteristic of the central pixel.

Finally, the smoothing efficiency of weighted D-filter is compared with that of plain D-filter, which is known to have a very good smoothing efficiency for a number of distributions. It is intuitively clear that the weighted D-filter, with the proper choice of the parameter q, does retain most of the smoothing efficiency of the D-filter as it mimics the D-filter over the flat regions. However, to rigorously answer this question, we compare the performance of the D and the weighted D estimators in the following manner: We consider a number of different distributions with constant mean. For each distribution, 2000 samples are generated. The window size is selected as 9. With a moving window of size 9 and the samples, the signal value at each point is estimated by means of the D and the weighted D estimators. The variances of these estimators are computed and chosen as the performance measure. In Table II we summarize the results. RE stands for the relative efficiency which is defined as $\text{Var}(D)/\text{Var}(\text{weighted } D)$. PI stands for the probability of inclusion, which is calculated as:

TABLE II(a)
RE AND PI FOR WEIGHTED D-FILTER; $q = 3.2\times(s.d.)$

	Parameter	RE	PI
Laplacian	$\mu = 6$	0.9552	0.9648
	$\mu = 11$	0.9162	0.9647
Gaussian	$\sigma = 8$	0.9357	0.9765
	$\sigma = 15$	0.9432	0.9764
Uniform	$m = 14$	0.9994	0.9942
	$m = 26$	0.9925	0.9942

TABLE II(b)
RE AND PI FOR WEIGHTED D-FILTER; $q = 2.8\times(s.d)$

	Parameter	RE	PI
Laplacian	$\mu = 6$	0.8646	0.9554
	$\mu = 11$	0.8438	0.9552
Gaussian	$\sigma = 8$	0.9209	0.9662
	$\mu = 15$	0.9135	0.9662
Uniform	$m = 14$	0.9548	0.9823
	$m = 26$	0.9429	0.9821

$$PI = \int_{-\infty}^{+\infty} f(y)\,dy \int_{y-q}^{y+q} f(x)\,dx \tag{8}$$

where q is the parameter used in weighted D-filter, and $f(\cdot)$ is the underlined distribution of the data. Since the weighted D-filter practically includes a segment of the sample, rather than the whole sample, for filtering, PI gives the average rate of sample inclusion.

From Table II, we conclude that if we choose the parameter q in the range 2.8σ–3.2σ, the PI index is close to one and the loss of smoothing efficiency, as compared to the D-filter, is relatively small. The filter is also tried on a natural image. The detailed experimental results are reported in [12]. The results with the natural image are very good for the weighted D-filter.

On the basis of our study, the following comments are in order.

1) If we choose the parameter q properly, i.e., in the range 2.5σ–3σ, and if the additive noise is moderate, i.e., σ is less than 6% of the dynamic range, the weighted D-filter works nicely as a robust edge and detail preserving smoothing filter. For large noise, a bigger value of q is required to smooth the flat regions. However, a very large value of q means that the weak edges or the thin-line signals may not be preserved.
2) The weighted D-filter uses three possible values for the weight: 0, 0.5, and 1.0. It is conceivable that a bigger set of values, or even a continuous set of values in the range 0–1 might improve the filter performance even more. A bigger set of weight values also requires an elaborate scheme for weight assignment. Investigations are underway in this direction.
3) In a recent paper, Gandhi *et al.* [11] has developed an important approach for the design of edge and detail preserving smoothing filters incorporating winsorization both in rank and temporal domains. The temporal winsorization, as described in [11] for 1-D signals, can be formalized with respect to two temporal (or spatial) coordinates to design filters for image signals. It is interesting to note that in the weighted D-filter, the rank winsorization depends on the central pixel value. The central pixel

provides the temporal index. Thus the filter uses a temporal index based rank winsorization scheme.

References

[1] A. Nieminen, P. Heinonen, and Y. Neuvo, "A new class of detail-preserving filters for image processing," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 9, pp. 74–90, Jan. 1987.

[2] J. W. Tukey, *Exploratory Data Analysis*. *Reading, MA*: Addison-Wesley, 1971.

[3] T. S. Huang, Ed., *Two-Dimensional Digital Signal Processing II: Transforms and Median Filters*. New York: Springer-Verlag, 1981.

[4] N. C. Gallagher, Jr., 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.

[5] J. P. Fitch, E. J. Coyle, and N. C. Gallagher, Jr., "Root properties and convergence rates of median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 230–239, Feb. 1985.

[6] A. C. Bovik, T. S. Huang, and D. C. Munson, Jr., "Edge-sensitive image restoration using order-constrained least square methods," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1253–1263, Oct. 1985.

[7] Y. H. Lee and S. A. Kassam, "Generalized median filtering and related nonlinear filtering techniques," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 672–683, June 1985.

[8] G. R. Arce and R. E. Foster, "Detail preserving ranked-order based filters for image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 83–98, 1989.

[9] L. Peterson, Y. H. Lee, and S. A. Kassam, "Some statistical properties of alpha-trimmed mean and standard type M filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 707–713, 1988.

[10] A. Kundu and W. R. Wu, "Double-window Hodges-Lehman *D* filter and hybrid *D* median filter for robust image smoothing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1293–1298.

[11] P. P. Gandhi, I. Song, and S. A. Kassam, "Nonlinear smoothing filters based on rank estimates of location," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1359–1379, Sept. 1989.

[12] W. R. Wu and A. Kundu, "Weighted *D* Filtering," in *SPIE Conf. Nonlinear Image Processing*, vol. 1247, pp. 100–111, Santa Clara, CA, 1990.

Parallel Architectures for Multirate Superresolution Spectrum Analyzers

Seth D. Silverstein, William E. Engeler, and John A. Tardif

Abstract —**This work develops parallel architectures for implementing matrix based superresolution spectral estimation algorithms for situations that require high levels of resolution commensurate with large coherent apertures and large sample orders. The featured architectures couple matrix based superresolution algorithms together with front-end multirate decimation preprocessor. This procedure creates parallel pseudo-apertures corresponding to different sub-bands of the temporal (or spatial) frequency spectrum. The overall superresolution of the large aperture is maintained. Simulations applying the large array architectures to the Tufts–Kumaresan reduced rank modified covariance algorithm and the linear minimum free energy/regularized form of the modified covariance algorithm are given for 1024-element coherent apertures.**

Manuscript received February 19, 1990; revised November 29, 1990. This paper was recommended by Associate Editor Y. F. Huang.

The authors are with the General Electric Research and Development Center, Schenectady, NY 12301.

IEEE Log Number 9042417.

I. Introduction

This work introduces parallel architectures for implementing matrix based superresolution spectral estimation algorithms for situations that require high levels of resolution commensurate with large coherent apertures and large sample orders. Algorithms involving large order matrices are computationally burdensome and often suffer from stability problems associated with ill-conditioning. The parallel algorithms/architectures discussed and simulated in this work demonstrate an efficient algorithm that preserves the potential Rayleigh resolution of the full aperture, and reduces matrix orders to levels where calculations are feasible.

There is a number of techniques used to avoid large matrix problems. Unfortunately, most of these approaches compromise the potential system resolution. For example, the division of a long coherent aperture into nonoverlapping subapertures, each with sample orders that are small enough to make matrix operations feasible, reduces the Rayleigh resolution to that of the shorter length subapertures. Another technique involves reducing the order of parametric models to levels small enough to suppress instabilities. The arithmetic instabilities that are manifested in spurious peaks are caused by large noise-induced fluctuations in the small eigenvalues of the autocorrelation matrices. This latter method also significantly degrades resolution.

The featured architectures are based upon a simple application of the sampling theorem. To prevent aliasing, a baseband signal of bandwidth B, $f \in [-B/2, +B/2]$, must be sampled at a rate greater than the Nyquist rate, which is equal to the baseband bandwidth B. Here frequency is used in a generic sense. It relates to temporal frequency for time series applications, while for spatial phased arrays, the spatial frequency is equal to one-half of the sine of the bearing angle.

Consider a uniformly sampled coherent aperture consisting of N complex samples, $\{x(n)\}$, each separated by the Nyquist interval $1/B$. The total length of the coherent aperture is N/B, and the Rayleigh resolution is B/N. By digital filtering, divide the spectrum up into K sub-bands of equal bandwidth. The filtering operation gives K sets of signals $\{y_l(n)\}$ (all of order N). The sub-bands can be sampled at the reduced rate $B \to B/K$, which corresponds to a decimation from $N \to N/K = Q$ samples. The decimated "pseudo"-arrays that describe the sub-band spectra are uniformly sampled with sampling intervals equal to K/B. Each of the sub-band pseudo-arrays have the same total length as the original aperture, $KQ/B = N/B$, which implies that the sub-band Rayleigh resolution $\rho_{\text{Ray}} = B/N$ *is unchanged* from that of the original total array. All these processes can be performed in parallel on each of the sub-bands, and the results can be fed in parallel into a bank of superresolution processors. These pseudo-arrays now have sample orders that are sufficiently reduced to make the necessary matrix operations practical.

The effectiveness of the large order array algorithm will be demonstrated by single snapshot (single realization of a time series) simulations featuring two noise suppressing superresolution algorithms: the Tufts–Kumaresan (T–K) reduced rank modified covariance algorithm [1], and the linear minimum free energy extension (LMFE) of the modified covariance algorithm [2], [3]. Both of these algorithms are formulated with forward and backward smoothed forms of the sample covariance, which is often referred to as a *modified covariance matrix*. In single snapshot applications of MUSIC subspace type algorithms, spatial smoothing is necessary to increase the rank of the signal covariance matrix to a level commensurate with the actual number of sources. Simulations demonstrating these large array preprocessors for single snapshot applications of spatially

0098-4094/91/0400-0449$01.00