GEOPHYSICS, VOL. 78, NO. 3 (MAY-JUNE 2013); P. V79–V86, 18 FIGS., 1 TABLE.
10.1190/GEO2012-0232.1

Noise reduction by vector median filtering

Yike Liu[1]

ABSTRACT

The scalar median filter (SMF) is often used to reduce noise in scalar geophysical data. We present an extension of the SMF to a vector median filter (VMF) for suppressing noise contained in geophysical data represented by multidimensional, multicomponent vector fields. Although the SMF can be applied to each component of a vector field individually, the VMF is applied to all components simultaneously. Like the SMF, the VMF intends to suppress random noise while preserving discontinuities in the vector fields. Preserving such discontinuities is essential for exploration geophysics because discontinuities often manifest important geologic features such as faults and stratigraphic channels. The VMF is applied to synthetic and field data sets. The results are compared to those generated by using SMF, *f-x* deconvolution, and mean filters. Our results indicate that the VMF can reduce noise while preserving discontinuities more effectively than the alternatives. In addition, a fast VMF algorithm is described for reducing computation time.

INTRODUCTION

Constructing subsurface images and estimating rock properties within the earth are essential tasks for exploration geophysics. Many types of data (e.g., seismic or electromagnetic) are acquired and processed for probing the subsurface structures and rock properties. These data are often acquired remotely from targets and may suffer from severe noise contamination. In many cases, noise in data must be reduced by data processing before useful information could possibly be extracted from the raw measurements.

Many practical methods have been developed for noise reduction in geophysics. Examples include the mean filter (mean values calculated in moving windows), *f-k* filter, *f-x* deconvolution (Canales, 1984; Marfurt, 2006), Radon transform (Sacchi and Porsani, 1999; Sacchi et al., 2004), edge-preserving smoothing (Luo et al., 2002), and scalar median filter (SMF) (Mi and Margrave, 2000). Compared to other methods, the SMF filtering technique often produces less smearing among adjacent samples after noise attenuation. For example, the SMF can remove an abnormal impulse from seismic records without smearing the impulse into its nearby samples as the mean or *f-k* filters do. The SMF can also be used to separate the up- and downgoing wavefields in vertical seismic profile (VSP) data because it often reduces interference between these two wavefields.

Despite the fact that numerous types of geophysical data are naturally represented by vector fields, the vector median filter (VMF) is seldom employed in data processing. On the other hand, SMF is commonly used in exploration geophysics. Examples of vector geophysical data include electromagnetic data, multicomponent seismic wavefields, dip and azimuth of events in 3D seismic images, fracture orientations, move-out slopes in migrated common-image gathers, amplitude and phase spectra of seismic traces, and windowed segments of seismic traces. Because each component of a vector field reflects a certain property of a common objective, there may be some coherent relation between these components. Therefore, it can be beneficial to process all components of a vector field simultaneously, rather than treat each component as an independent scalar field.

We introduce the VMF to reduce noise in vector fields. The VMF was first recommended for image processing, and many papers have been published on this subject (Astola et al., 1990; Plataniotis et al., 1998; Caselles et al., 2000; Lukac and Smolka, 2003; Spence and Fancourt, 2007; Liu et al., 2009). Astola et al. (1990) demonstrate that the VMF can reduce noise while preserving sharp boundaries in color images, where each pixel is represented by three (i.e., red, green, blue) component vectors. For comparison, it has been demonstrated that applying the SMF to each color component individually produces an inferior result. In this paper, we demonstrate similar benefits in applying the VMF to attenuate noise in a dip/azimuth vector field derived from a migrated seismic image. First, we introduce the concepts of the median value, median vector, SMF, and VMF, along with a fast VMF algorithm. Then, we describe

Manuscript received by the Editor 20 June 2012; revised manuscript received 30 December 2012; published online 10 April 2013; corrected version published online 24 April 2013.
[1]Chinese Academy of Sciences, Institute of Geology and Geophysics, Beijing, China. E-mail: ykliu@mail.igcas.ac.cn.

three properties of the SMF and VMF. Finally, we apply the VMF to synthetic and field seismic data to demonstrate that the VMF can effectively reduce noise while preserving discontinuities.

THEORY

Median vector

Let us start by defining the well-known scalar median value. Given a set of scalars $S_i = \{a_i\}\ i = 1, 2, \ldots, N$, if the set is sorted into ascending (or descending) order, then the value of a member in the middle of the sorted set is the median. This sorting-based definition is intuitive and easy to understand, but hard to extend to a set of vectors. To allow for an extendable definition, we redefine the scalar median value based on a minimum-distance concept. The median member a_m, according to the minimum-distance definition, is the member whose distance to all other members in the set is smallest. This definition can be expressed as

$$a_m = \underset{a_m \in S_i}{\operatorname{argmin}} \sum_{i=1}^{N} \|a_m - a_i\|_L, \tag{1}$$

where i is the summation index, N is the number of members in the set, and L denotes the order of the norm. Any proper norm (e.g., L_1, L_2 or L_∞) is eligible to be used in this definition.

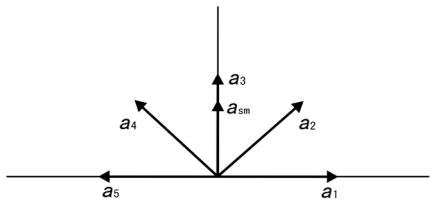

Figure 1. Comparison of the vector median vector and the scalar median vector. The median vector $\mathbf{a}_3$ is a member of the input set of vectors, whereas the scalar median vector $\mathbf{a}_{sm}$ found by taking the scalar median of each component of vectors in the input set is not a member of the input set.

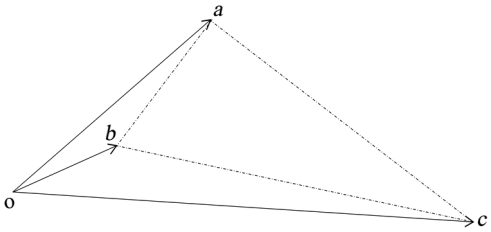

Figure 2. The median vector depends on the choice of norm for the three vectors a, b, and c. If the L_2 norm is used, the median is a, because the Euclidean distance from a to c is smaller than the distance from b to c. For the L1-case, the L1-distance a to c is larger than b to c because the horizontal distance from a to b is less than the vertical distance.

It is easy to prove (see Appendix A) that the median value defined by equation 1 is identical for any scalar data set, regardless of what type of norm or sorting is chosen. In addition to the proof, the equivalence of these two definitions can also be demonstrated using a simple example. For a five-member set $\{1, 2, 3, 4, 5000\}$, the median value is three, according to the sorting-based definition. For the minimum-distance definition, the summed L_1-norm distances from the first member to all other members is $\|1-1\| + \|2-1\| + \|3-1\| + \|4-1\| + \|5000-1\| = 5005$. Similarly, the summed distances from the second, third, fourth, and fifth members are 5002, 5001, 5002, and 19,990, respectively. Here, the minimum distance is 5001 and is associated with the third member. Therefore, the median value obtained from this definition is three, the same as that based on the sorting approach.

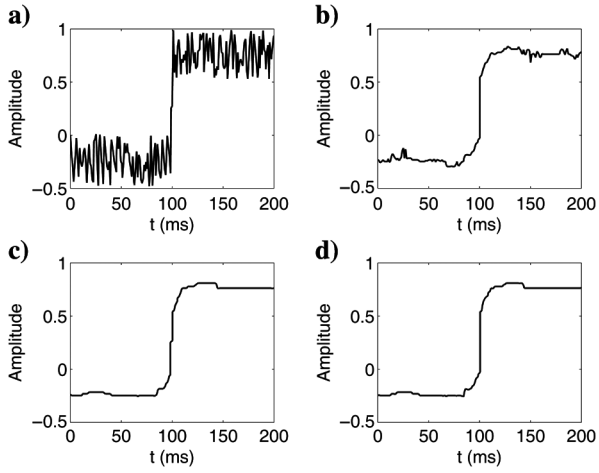

Figure 4. A 1D scalar function (a) is filtered using one iteration (b), five iterations (c), and fifty iterations (d) of the SMF.

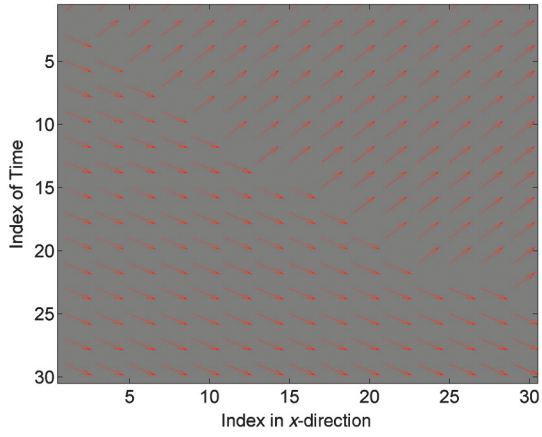

Figure 5. Two geologic structural trends indicated with vectors along northeast and southeast.

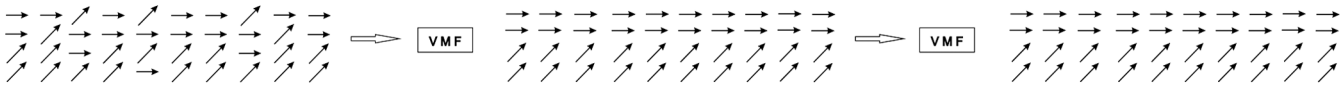

Figure 3. A 2D vector field (a) is filtered using the VMF (b). Note that the discontinuity between the second and third rows of vectors is preserved, while the erroneous vectors have been removed. Applying the VMF a second time (c) results in no further change to the vector field shown in (b).

Equation 1 can be easily extended to define the median vector, which is the vector $\mathbf{a}_m$ for which

$$\mathbf{a}_m = \underset{\mathbf{a}_m \in s_i}{\operatorname{argmin}} \sum_{i=1}^{N} \|\mathbf{a}_m - \mathbf{a}_i\|_L. \quad (2)$$

Equation 2 is the same as equation 1, except that the scalar a_i is now replaced by the vector $\mathbf{a_i}$, where bold letters indicate vectors and $S_i = \{\mathbf{a_i}\}$ is a vector set. Indeed, equation 1 can be considered a special case of equation 2 when the vector has only one component. A simple example of a median vector for a five-member 2C vector set is depicted in Figure 1. There, the input vectors $\mathbf{a}_1$ through $\mathbf{a}_5$ are unit vectors with coordinates (−1, 0), (−0.707, 0.707), (0,1), (0.707, 0.707), and (1,0). Using equation 2, we can find that $\mathbf{a}_3 = (0, 1)$ is the median vector because the L_2-norm summed distance from $\mathbf{a}_3$ to all other vectors is the smallest. In contrast, if we attempt to calculate the median vector $\mathbf{a}_{sm}$ by finding the scalar median among components of the vectors, the mixed "median" vector $\mathbf{a}_{sm}$ will be (0, 0.707). Note that $\mathbf{a}_{sm}$ is not a member of the five-member input vector set. Creating a member not originally in the input may blur sharp boundaries.

VMF

Based on the median vector defined in equation 2, we define the VMF. The VMF is similar to the mean filter, which smooths data by taking the mean within a windowed subset of the data. Instead of finding the mean for every windowed subset, the VMF finds the median vector. Given a set of vectors $S_i = \{a_{i-j}, a_{i-j+1}, \ldots, a_{i+j-1}, a_{i+j}\}$ where j is the window half-width, the median vector is defined as

$$\mathbf{a}_{m_i} = \underset{\mathbf{a}_m \in S_i}{\operatorname{argmin}} \sum_{k=i-j}^{k=i+j} \|\mathbf{a}_m - \mathbf{a}_k\|_L. \quad (3)$$

This definition is applicable to SMFs and VMFs.

The distance in equations 2 or 3 can be computed using any norm. When uncorrelated noise is present in the data, the L_1 norm typically produces the best results. However, when there is coherent noise in the data, other norms (often, the L_2 norm) produce better results in many practical applications (Plataniotis et al., 1998). Figure 2 illustrates that the median vector depends on the choice of norm. In the figure, there are three vectors a, b, and c.

In this example, if the L_2 norm is used, the median vector is **a** because the Euclidean distance $|bc| > |ac| > |ab|$. For the L1-case, the L1-distance $|ac| > |bc|$ because the horizontal distance from **a** to **b** is less than the vertical distance. It is useful to note that SMF results are independent of the norm order (proved in Appendix A), whereas the VMF output depends on the norm order.

It is possible that more than one vector satisfies the minimum summed distance criterion. In this case, we introduce an additional condition to remove this ambiguity. For example, we might choose — among vectors having equal minimum summed distance — the one closest to the input vector at the center of the moving window.

Fast computation of the VMF

Computing median vectors for a large data set can be expensive. Given a moving window with half-width $W/2$ (where W is an even

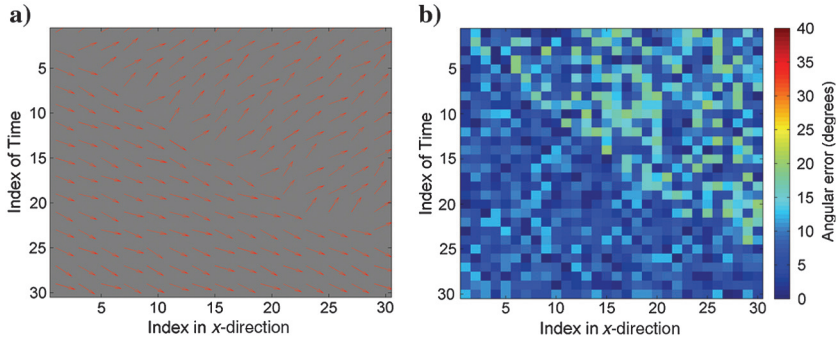

Figure 6. (a) The vector field of Figure 5 after applying random rotations of up to ±19.79 northeast and ±12.53 southeast. (b) The difference between the rotated and unrotated angles.

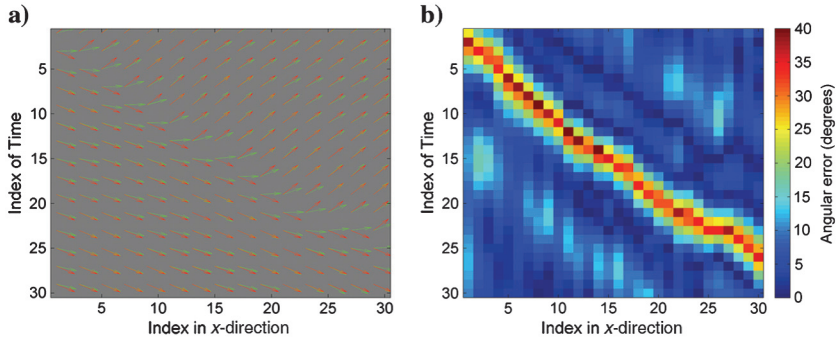

Figure 7. A comparison of the vector fields between the unrotated (red) (Figure 5) and after filtering the rotated vector fields of Figure 6a by the *f-x* deconvolution filter (green) (a), and their dip difference (b).

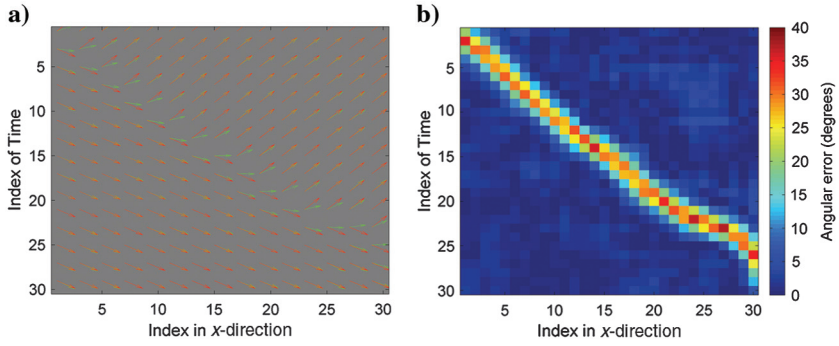

Figure 8. A comparison of the vector fields between unrotated (red) (Figure 5) and after filtering the rotated vector fields of Figure 6a by the mean filter (green) (a), and their dip difference (b).

number), then $O(W)$ operations are required to compute the summed distance from one vector to all others. Thus, computing all summed distances requires $O(W^2)$ operations. For a 1D function with N vectors, the computational cost is $O(N * W^2)$. In contrast, to compute one output value using a mean filter requires only $O(N)$ operations if the filter is implemented properly.

Fortunately, the computational cost of the VMF for a 1D function can be reduced from $O(N * W^2)$ to $O(N * W)$ using the fast VMF algorithm proposed by Astola et al. (1990). This cost reduction is achieved by reusing the summed distances computed for previous windows. We demonstrate the fast VMF concept with a simple 1D example, but the same principle can be applied to higher dimensions.

We revised the fast algorithm to make it easier to understand and implement. Let $\mathbf{a}_j^k$ denote the jth vector in the kth subset or window containing $W+1$ vectors, and s_j^k denote the summed distance to all other vectors in the kth window

$$s_j^k = \sum_{i=1}^{W+1} \|\mathbf{a}_j^k - \mathbf{a}_i^k\|_L, \quad j = 1, 2 \ldots, W+1, \quad \text{and}$$
$$k = 1, 2, \ldots, N. \tag{4}$$

Then, write s_j^{k+1} in terms of s_j^k:

$$\begin{cases} s_j^{k+1} = s_{j+1}^k + \|\mathbf{a}_j^{k+1} - \mathbf{a}_{W+1}^{k+1}\|_L - \|\mathbf{a}_{j+1}^k - \mathbf{a}_1^k\|_L, j = 1, \ldots W, \text{and} k = 1, \ldots, N \\ s_{W+1}^{k+1} = \sum_{i=1}^{W+1} \|\mathbf{a}_{W+1}^{k+1} - \mathbf{a}_i^{k+1}\|_L \end{cases}. \tag{5}$$

Note that vector $\mathbf{a}_j^{k+1}$ is the same as $\mathbf{a}_{j+1}^k$. With this in mind, the interpretation of equation 5 is simple — it removes the contribution of $\mathbf{a}_1^k$ from s_j^k and adds the contribution of $\mathbf{a}_{W+1}^{k+1}$ to s_j^{k+1}. The fast algorithm described by equation 5 requires only $2 * (W+1)$ operations to calculate all sums in the $(k+1)$th window. Therefore, the total cost is $O(N * W)$. Equation 5 is applicable to the SMF and the VMF, and the computational costs of both filters are of the same order.

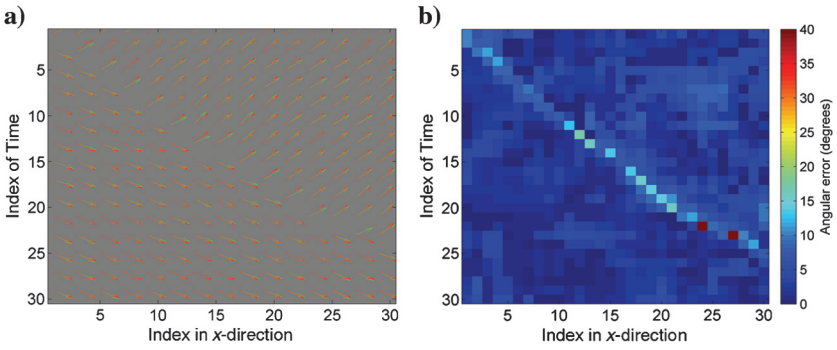

Figure 9. A comparison of the vector fields between unrotated (red) (Figure 5) and after filtering the rotated vector fields of Figure 6a by the SMF filter (green) (a), and their dip difference (b).

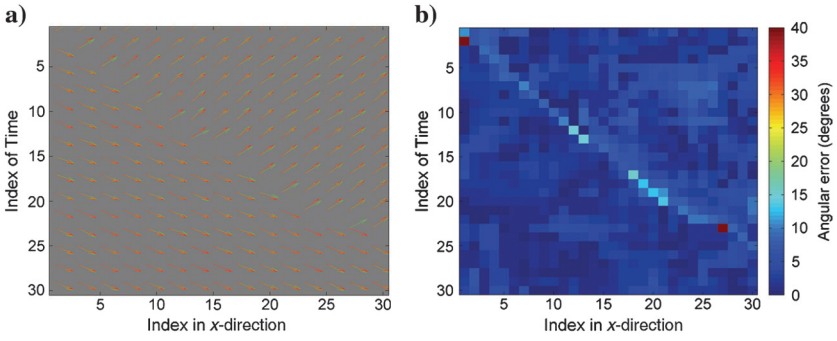

Figure 10. A comparison of the vector fields between unrotated (red) (Figure 5) and after filtering the rotated vector fields of Figure 6a by the L1-VMF filter (green) (a), and their dip difference (b).

VMF PROPERTIES

The VMF possesses three important properties: closure, edge preservation, and iteration invariance. These properties allow the VMF to preserve discontinuities while attenuating noise.

Property 1: Closure

The closure property requires that the median vector is a member of the input set. This means that the VMF will never output a vector that does not exist in the original input set of vectors. This property does not apply to the mean filter. For example, for a set $\{1 \quad 2 \quad 6\}$, the median value is 2, wheras the mean value is 3, which does not exist in the original set. This closure property may seem trivial, but it leads to the next two important properties.

Property 2: Edge preservation

The most attractive feature of the VMF is that it reduces noise while preserving edges, or discontinuities, in the data. To illustrate this property, Figure 3 shows an example simulating a geologic unconformity. In the figure, the first two rows of vectors are mostly horizontal, whereas the last two rows are nearly vertical. The VMF with a window size 3×3 is applied to the 2D vector field shown in Figure 3a. The window moves from left to right and from top to bottom to cover the whole area and outputs

Table 1. Differences in rms dip (degree).

Filter	L1-VMF	L2-VMF	SMF	*f-x* deconvolution	Mean	Input (without filtering)
rms	4.752	4.873	4.941	12.28	8.668	9.550

the median vector at the center of each window (shown in Figure 3b). Notice that the erroneous vectors in Figure 3a have been removed and the boundary of the unconformity is preserved.

There are other ways to reduce the noise in Figure 3a. Perhaps the simplest is to calculate the mean vector by separately averaging each component of vectors within a moving window. However, as the mean vector is a mix of different vectors, the output will contain other vectors beside the two kinds of vectors shown in Figure 3b and the sharp boundary will be blurred. Another possible way to reduce noise is to apply the SMF to each component of the vector field separately. However, others (e.g., Astola et al., 1990) have shown that the VMF produces better results than applying multiple SMFs to individual vector components.

Property 3: Iteration invariance

Because the output vector of a VMF is one of the input vectors, in some circumstances, the output of the VMF will be identical to the input. Such an input is referred to as the root function of the VMF.

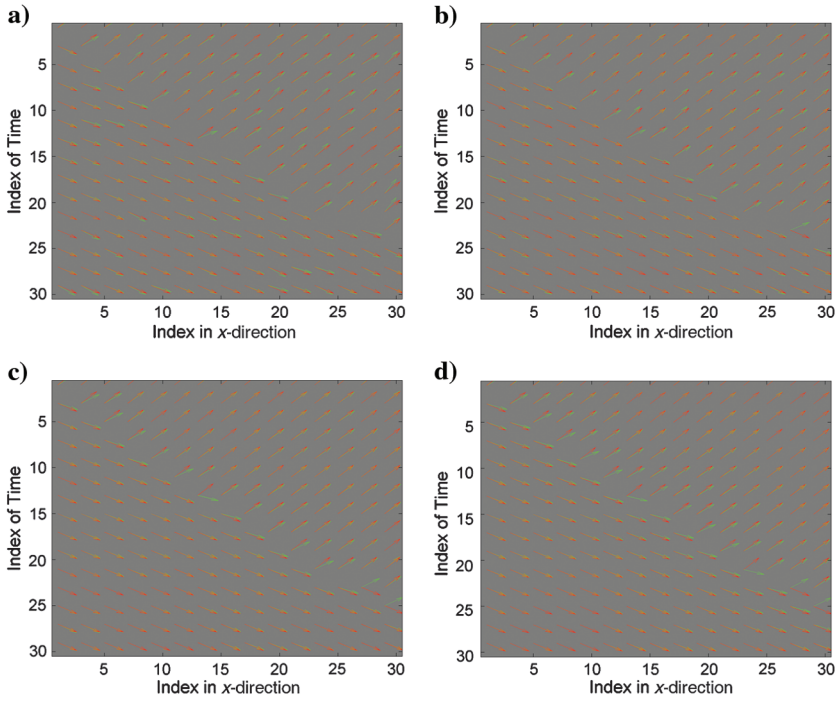

Figure 11. (a) through (d) show the output of applying L_2-VMF with window size of 3, 5, 9, and 15 samples, respectively, to the vectors with random noise (Figure 6a) where green shows the filtering results and red is the unrotated vectors (Figure 5).

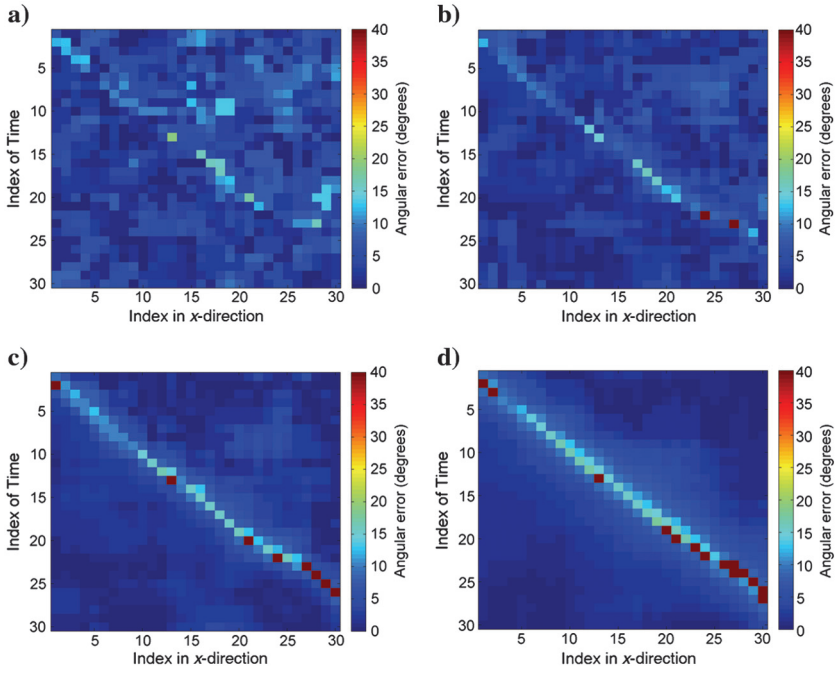

Figure 12. Dip difference after filtering using the L_2-VMF with window sizes of 3 (a), 5 (b), 9 (c), and 15 (d) samples in each axis.

Figure 4 demonstrates the iteration invariance of the SMF for a 1D scalar field. Because the input function is 1D, a SMF (a special case of VMF) is used. As a 1D median filter, a 21-sample window moves from left to right, and a median value is generated at the center of each window. After one (Figure 4b) and five (Figure 4c) iterations, the SMF has reduced the noise in the input function and maintained a step-like function. More iterations (Figure 4d) after the fifth one result in no further change to the result shown in Figure 4c. This property, known as iteration invariance, is inherited by the VMF. The first iteration of VMF applied to the input vector field shown in Figure 3a produces the noise-free vector field (Figure 3b). Figure 3c shows no further change after applying the VMF again to the vector field shown in Figure 3b. The edge-preserving property of the VMF makes it a desirable tool for noise reduction while maintaining discontinuities. Moreover, because the output of the VMF becomes iteration invariant, the VMF can be applied recursively until a stable result (i.e., the root function) is obtained.

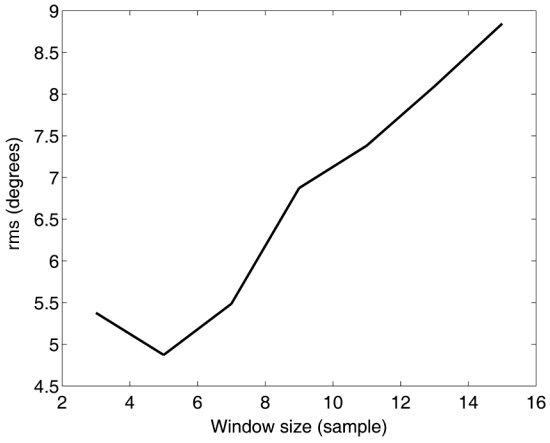

Figure 13. The quality of VMF noise reduction and edge preservation as a function of window size. The quality measure is the rms difference between the filtered result and the correct vector field.

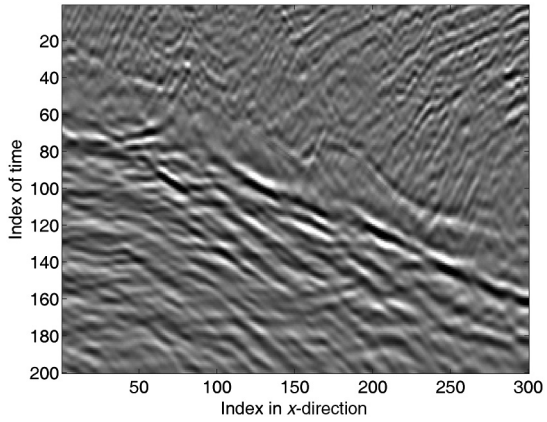

Figure 14. A migrated seismic amplitude section, which is used as the input for computing dip vectors of seismic events at every sampled location. Note there is an abrupt dip change around the center of the figure.

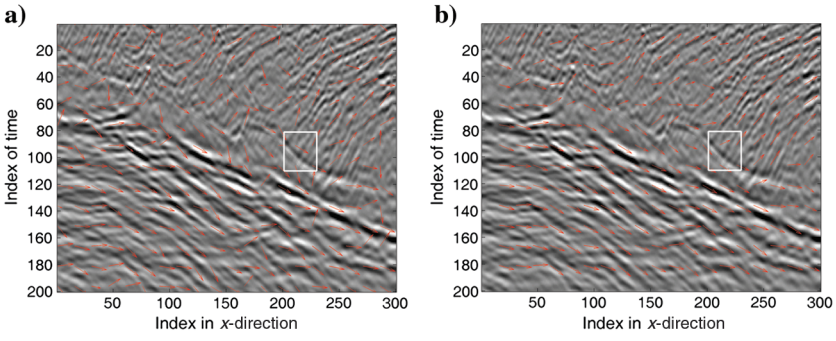

Figure 15. A subset of the dip vectors before (a) and after (b) applying the L_1-VMF.

NUMERICAL EXAMPLES

To evaluate the effectiveness of the VMF, we designed a synthetic 2D vector field to simulate a geologic unconformity (Figure 5) based on the prototype of real data. The vector field consists of two distinguishing sets of vectors — the northeast and southeast vectors. To introduce noise, the orientations of the northeast vectors are randomly rotated up to ± 19.79 and the southeast vectors $\pm 12.53°$ from their original angles. The vector field after the random rotations is shown in Figure 6a; and the difference between the rotated and unrotated angles is shown in Figure 6b.

Figures 7, 8, 9, and 10 display, respectively, the results after applying *f-x* deconvolution, a mean filter, a SMF and a L1-VMF to the rotated data shown in Figure 6a. The window size and filter length used in the *f-x* deconvolution are 100 and 40, respectively. The window size for the other approaches is 5×5. The filtered results are compared to the unrotated vectors shown in Figure 5. We kept the length of the vectors constant and measured the angle errors between the nonrotated and rotated vectors. As seen in Figures 7b and 8b, the angle differences are reduced significantly in areas away from the unconformity boundary after applying the *f-x* deconvolution, or mean filter; however, large differences still remain near the boundary. These large errors can also be seen in Figures 7a and 8a where the green arrows gradually change from southeast to northeast. The unconformity boundary is poorly preserved by *f-x* deconvolution, or mean filtering.

Figures 9 and 10 are, respectively, the results of applying the SMF and L1-VMF. Figures 9b and 10b show that the unconformity boundary is preserved better than in Figures 7b and 8b. For a more objective comparison, we introduce a quantitative root mean square (rms) index between angles of the filtered and intact vectors. This index serves as a simple indication of the effectiveness of various approaches. The calculated rms dip differences are listed in Table 1. The rms of the *f-x* deconvolution result is even larger than the rms of the input data. This may be due to the mixture of northeast and southeast vectors near the unconformity.

Figure 11a, 11b, 11c, and 11d shows the output after applying the L_2-VMF with square window sizes of 3, 5, 9, and 15, respectively. Note that for the region away from the unconformity, more noise is attenuated as the window size becomes larger. In contrast, the discontinuity is better preserved for smaller window sizes (Figure 12). Figure 13 summarizes the rms difference

as a function of window size. The minimum rms is reached when the window size is 5×5. In this example, and for this criterion, the window size of 5 optimizes the noise attenuation and edge preservation. In general, the optimal window size is data-dependent.

APPLICATION TO 2D FIELD SEISMIC DATA

Figure 14 shows a 2D migrated vertical seismic section, which is a portion of an inline section of a 3D migrated cube. The displayed section has 300 traces and 200 time samples. The trace interval is 25 m and the temporal sample rate is 4 ms. Using structure tensors (Fehmers and Hocker, 2003; Luo et al., 2006), we computed the local dip vectors of seismic events at every sample location. The dip vectors are not physical dips, but are defined in terms of samples. In other words, they are normalized using a velocity of 25 m/0.004 s = 6250 m/s. The computed dip vectors are overlain on the section of Figure 15a. The dip vectors are parallel to local events in the image, and the length of the dip vector varies according to the lateral coherency of nearby traces. Figure 15b shows the result of applying the L_1-VMF with a window size of 13 to the vector field shown in Figure 15a.

Figure 16 shows a zoom of the box marked on Figure 15a. Figure 17 shows the result of applying a mean filter, *f-x* deconvolution, SMF, L_2-VMF, and L_1-VMF, where the windows size is 5×5, except a window size of 100 and filter length of 40 are used for *f-x* deconvolution. The mean filter, *f-x* deconvolution, and SMF are applied to each component of the vector field separately, whereas the VMFs are applied to all components. The white lines in Figure 17a through 17e indicate an interpreted unconformity, across which we expect the dip vectors to change abruptly. Note that the green arrows in Figure 17a and 17e change gradually near the unconformity, which indicates that the mean filter and *f-x* deconvolution are blurring the discontinuity. In contrast, the results of the SMF (Figure 17b), L_2-VMF (Figure 17c) and L_1-VMF (Figure 17d) show better preservation of the discontinuity in the vector field.

As the window size becomes larger, the computation time for VMF increases accordingly. Figure 18 depicts the computation time for applying the L_2-VMF to the data shown in Figure 15a. Although the fast VMF algorithm is used, the computation cost for the VMF is still very high. The computation time as measured by using a MATLAB code on a workstation is dependent on window size and the number of components. Nevertheless, an algorithm based on the fast VMF has been developed and applied to 3D prestack data for separating blended seismic wavefields generated by simultaneous sources (Huo et al., 2009). Note that the number of vectors in the window is the square of the window size for 2D cases, so the computing time increase depicted in Figure 18 seems much faster than a linear speedup with window size.

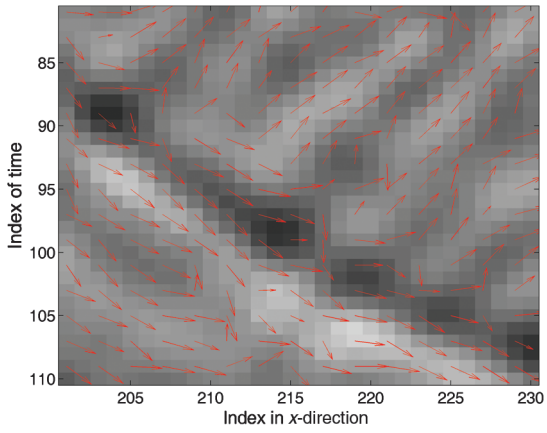

Figure 16. Zoom of the computed dip vectors of Figure 15a.

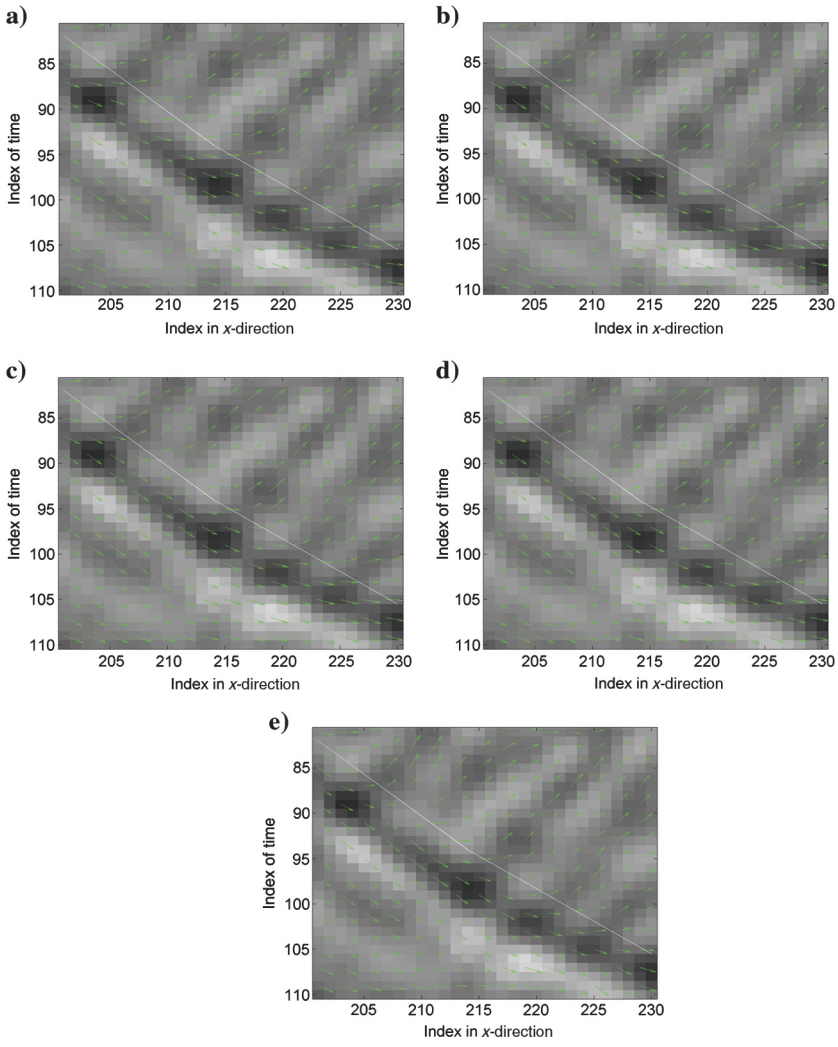

Figure 17. Field data after filtering by mean filter (a), SMF (b), L_2-VMF (c) and L_1-VMF (d), *f-x* deconvolution (e), respectively.

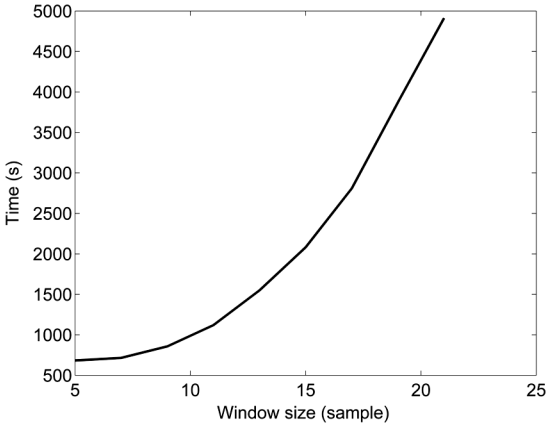

Figure 18. The L_2-VMF computation time versus window size.

CONCLUSIONS

We have extended the SMF to the VMF for reducing random noise in geophysical vector fields, while preserving discontinuities. A fast algorithm for implementing the VMF is described. The VMF and SMF effectively reduce noise in vector fields while preserving discontinuities. The result of applying VMF to synthetic data is superior to applying SMF as measured by rms error. This demonstrates that VMF is slightly better at noise reduction because VMF uses the correlation between components. A field data example demonstrates that all median filters preserve the discontinuity much better than does the mean filter or *f-x* deconvolution.

ACKNOWLEDGMENTS

This work has been partially supported by National Nature Science Foundation of China (Grant No. 40930421, 41074091) and the National Oil and Gas Projects of China (2011ZX05008-006).

APPENDIX A

EQUIVALENCE OF TWO DEFINITIONS FOR 1D DATA

In this appendix, we show that median values based on the minimum-distance and sorting are equivalent. For a data set with $2P + 1$ scalar values, we sort the set into ascending order, i.e., a_j, $j = 1, 2, \ldots, 2P + 1$, and denote the median value derived from the sorting set by a_{P+1}. To prove the equivalence, we need to show that the distance from a_{P+1} to the rest of the values is the smallest. As a first step, the following shows that the summed distance from a_{P+2} to all other values is larger than the distance to a_{P+1}

$$\sum_{j=1}^{2P+1} \|(a_j - a_{P+2})\| = \sum_{j=1}^{P+1} \|(a_{P+2} - a_j)\| + \sum_{j=P+2}^{2P+1} \|(a_j - a_{P+2})\|. \quad \text{(A-1)}$$

Because a_j has been sorted into ascending order and a member with larger index is greater or equal to all others with smaller indices, we can dismiss the norm symbol and rewrite the equation as following

$$\begin{aligned}
&\sum_{j=1}^{P+1}\{(a_{P+2} - a_{P+1}) + (a_{P+1} - a_j)\} \\
&\quad + \sum_{j=P+2}^{2P+1} \{(a_j - a_{P+1}) - (a_{P+2} - a_{P+1})\} \\
&= \left\{\sum_{j=1}^{P+1} \|(a_j - a_{P+1})\| + (P+1)\|(a_{P+2} - a_{P+1})\|\right\} \\
&\quad + \left\{\sum_{j=P+2}^{2P+1} \|(a_j - a_{P+1})\| - (P)\|(a_{P+2} - a_{P+1})\|\right\} \\
&= \sum_{j=1}^{2P+1} \|(a_j - a_{P+1})\| + \|(a_{P+2} - a_{P+1})\|. \quad \text{(A-2)}
\end{aligned}$$

The first term in equation A-2 is the summed distance from the sorting-based median value; therefore, the summed distance from a_{P+2} is larger than that from a_{P+1}. This proof can be extended to any member a_j, and the norm could be in any order because a_j are scalars.

REFERENCES

Astola, J., P. Haavisto, and Y. Neuvo, 1990, Vector median filter: Proceedings of the IEEE, **78**, 678–689, doi: 10.1109/5.54807.

Canales, L. L., 1984, Random noise reduction: 54th Annual International Meeting, SEG, Expanded Abstracts, 525–527.

Caselles, V., G. Sapiro, and D. H. Chung, 2000, Vector median filters, inf-sup operations, and coupled PDE's theoretical connections: Journal of Mathematical Imaging and Vision, **12**, 109–119, doi: 10.1023/A:1008310305351.

Fehmers, G. C., and C. F. W. Hocker, 2003, Fast structural interpretation with structure-oriented filtering: Geophysics, **68**, 1286–1293, doi: 10.1190/1.1598121.

Huo, S., Y. Luo, and P. Kelamis, 2009, Simultaneous sources separation via multi-directional vector-median filter: 79th Annual International Meeting, SEG, Expanded Abstracts, 31–35.

Liu, Y., Y. Luo, and Y. Wang, 2009, Vector median filter and its applications in geophysics: 79th Annual International Meeting, SEG, Expanded Abstracts, 3342–3347.

Lukac, R., and B. Smolka, 2003, Application of the adaptive center-weighted vector median framework for the enhancement of cDNA microarray images: International Journal of Applied Mathematics and Computer Science, **13**, 369–383.

Luo, Y., M. Marhoon, S. A. Dossary, and M. Alfaraj, 2002, Edge-preserving smoothing and applications: The Leading Edge, **21**, 136–158, doi: 10.1190/1.1452603.

Luo, Y., Y. E. Wang, N. M. AlBinHassan, and M. Alfaraj, 2006, Computation of dips and azimuths with weighted structural tensor approach: Geophysics, **71**, no. 5, V119–V121, doi: 10.1190/1.2235591.

Marfurt, K. J., 2006, Robust estimates of 3D reflector dip and azimuth: Geophysics, **71**, no. 4, P29–P40, doi: 10.1190/1.2213049.

Mi, Y., and G. F. Margrave, 2000, Median filtering in Kirchhoff migration for noisy data: 70th Annual International Meeting, SEG, Expanded Abstracts, 822–825.

Plataniotis, K. N., D. Androutsos, and A. N. Venetsanopoulos, 1998, Color image processing using adaptive vector directional filters: IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, **45**, 1414–1419, doi: 10.1109/82.728854.

Sacchi, M. D., and M. Porsani, 1999, Fast high resolution parabolic Radon transform: 89th Annual International Meeting, SEG, Expanded Abstracts, 1477–1480.

Sacchi, M. D., D. J. Verschuur, and P. M. Zwartjes, 2004, Data reconstruction by generalized deconvolution: 74th Annual International Meeting, SEG, Expanded Abstracts, 1989–1992.

Spence, C., and C. Fancourt, 2007, An iterative method for vector median filtering: IEEE International Conference, **5**, V265–V268.