A COMPARATIVE STUDY OF THE COMPLEXITY OF MULTIVARIATE MEDIANS

Mauro Barni and Vito Cappellini

Department of Electronic Engineering, University of Florence
Via di S.Marta, 3, 50139 - Firenze, ITALY
e-mail barni@cosimo.die.unifi.it

ABSTRACT

Multivariate median filters represent a powerful tool for edge preserving noise removal from multichannel digital images. However, the usability of such filters in practical applications is often limited because of their high computational complexity, all the more that a comprehensive analysis of the complexity of the various classes of multivariate medians is still missing. In this work, the complexity of many multivariate extensions of the median filter is briefly discussed. Both theoretical analysis and experimental results show that the computational complexity depends mainly on the strategy adopted to sort multivariate data. The use of marginal ordering leads to the fastest algorithms, filters relying on reduced ordering have an intermediate behavior, whereas those based on aggregate ordering are by far the most complex.

1. INTRODUCTION

Considerable attention has recently been paid to the extension of scalar ranked-order filters to the multichannel case; to this aim, several schemes have been proposed trying to define a multichannel vector median operator with properties similar to those of its scalar counterpart. A problem often arising with multivariate median filters is computational complexity; all the more that, though the noise removal capabilities of these filters have been thoroughly investigated, a comprehensive analysis of their complexity has not been carried out yet.

Virtually all the multichannel order-statistics filters proposed so far are based on three different ranking schemes. According to the classical definition by Astola et al. [1], the aggregate ordering technique [2] is used: input samples are ordered on the basis of the sum of the distances to all the other points in the window. The output of the Vector Median filter (VM filter) is chosen as the first ordered sample, i.e. the point for which the sum is minimum. VM filters are a natural extension of scalar median filters, however, their high computational load prevents their use in many applications.

Filters based on multivariate reduced ordering (R-VM filters) have been first introduced by Hardie and Arce [3]: samples are ordered according to the distance to a properly chosen central point and the first ordered sample is the output of the filter. In [3] the effectiveness of this class of filters is carefully analyzed and its good filtering capabilities highlighted for the special case in which the sample mean vector is chosen as the central point. As to complexity, R-VM filters represent a considerable improvement with respect to VM filters, since a lower number of distances must be computed at each window location.

The simplest ranking scheme is marginal ordering [2], which consists in the independent ordering of the input sample components. The use of marginal ordering leads to the definition of the Marginal ordering Vector Median filter, (M-VM filters, [4]), which corresponds to the componentwise application of the scalar median filter. The M-VM filter and its generalized versions couple good filtering capabilities and low complexity [4], however, they can not be used in applications where closed operations are mandatory.

In this note, the problem of determining the complexity of filters belonging to the above three classes is briefly addressed.

Throughout the rest of the paper, an $n \times n$ filter window will be assumed; let also $N = n \times n$ be the number of samples contained in the filter window, with each sample being a point in R^p. The complexity of the filters will be computed in terms of the number of square roots, products, additions (actually algebraic sums, i.e. additions and subtractions), and comparisons that must be performed at each window location, and it will be considered as a function of n. Finally, we will suppose filtering is carried out by scanning the image by rows, from left to right.

2. CLASSICAL VM FILTERS

Given N samples $\{\vec{x}_1 ... \vec{x}_N\}$ in R^p the output of the vector median operator is defined as [1]

$$\begin{cases} \vec{x}_{VM} \in \{\vec{x}_1, \vec{x}_2 ... \vec{x}_N\} \\ \sum_{i=1}^{N} \| \vec{x}_{VM} - \vec{x}_i \| \leq \sum_{i=1}^{N} \| \vec{x}_j - \vec{x}_i \| \quad j = 1 \,..\, N \end{cases} \tag{1}$$

where $\| \ \|$ is a norm in R^p. The direct application of equation (1) requires the evaluation of the distances between all the possible couples of samples, i.e. $(n^2-1)n^2/2 = O(n^4)$ vector distances. By using a running algorithm faster operations can be achieved. More specifically, at each window location, the vector median operator is carried out by means

This work was supported by the "Progetto Finalizzato Beni Culturali - C.N.R." (Italian Finalized Project on Cultural Heritage - National Research Council).

of a 3-step process: i) the distances between the points leaving the filter window and those which were already inside it at the previous step are computed; ii) for each point, the sum of the distances to all the other points is updated by subtracting the distances to the points leaving the window and by adding those to the ones entering it; the sums relative to the new points are computed from scratch; iii) the point with the minimum sum is chosen. The first step requires the computation of $(n^2-n)n+n(n-1)/2$ distances; for the second one, $2n(n^2-n)$ algebraic sums are necessary to update the old sums and $n(n^2-1)$ additions are needed to initialize the sums of the new samples; finally, n^2-1 comparisons must be performed to choose the sample for which the sum of distances is minimum.

To determine the actual number of elementary operations required by a VM filter, the metric adopted to compute distances must be taken into account. Such numbers are summarized in Table I, whereas the overall complexity of VM filters is given in Table II. By focusing on asymptotic complexities, it turns out that the VM_2 filter has the highest complexity, since it requires the computation of $O(n^3)$ square roots, $O(n^3)$ multiplications and $O(n^3)$ additions. The complexity of the VM filter based on squared Euclidean metric is $O(n^3)$ too, but only multiplications and additions are involved. Finally, the VM_1 filter exhibits the lowest complexity, since only $O(n^3)$ additions and $O(n^3)$ comparisons must be evaluated.

2.1. Fast VM filters

To reduce the complexity of VM filters several fast algorithms have been proposed [5] [6] [7]. In [5] a fast algorithm is described for the squared Euclidean VM filter (F-VM_2^2 filter). The algorithm relies on the fact that the sample minimizing the sum of the squared Euclidean distances is the point closest to the window centroid $\vec{x}_a$ (the sample mean vector). This suggests the possibility of choosing the VM by simply computing the distance between each sample and $\vec{x}_a$. Once again the algorithm consists of 3 steps: i) evaluation of $\vec{x}_a$; ii) computation of the distance between the samples and $\vec{x}_a$; iii) choice of the minimum distance sample. If a running algorithm is used to compute the sample mean vector, $2pn$ additions and p divisions are needed to perform the first step. The choice of the point closest to the centroid can be carried out through the computation of n^2 squared Euclidean distances, i.e. pn^2 multiplications, $2pn^2$ additions and n^2-1 comparisons. Henceforth, the complexity of the F-VM_2^2 algorithm is $O(n^2)$ both in terms of multiplications and additions, which constitutes a considerable improvement with respect to the $O(n^3)$ complexity of classical algorithms.

VM filters based on 1-norm and 2-norm are preferable to the VM_2^2 filter because of their superior edge preserving capabilities. With regard to the Euclidean VM filter, no fast algorithm has been proposed so far, instead, attempts trying to speed it up rely on the approximate fast computation of the Euclidean distance (AF-VM_2 filter [6]. The overall scheme of the algorithm is the same as that of classical implementations, the only difference being the number of elementary operations required to compute the approximate distance. An accurate description of the approximation used to speed up the computation of the Euclidean norm is outside the scope of this brief note, here it only needs saying that the general form of the approximated norm is

$$\| \vec{x} \|_2 \simeq \| \vec{x} \|_{2,app} = \sum_{i=1}^{p} a_i \mid x_{(i)} \mid \tag{2}$$

where the a_i's are suitable constant and $x_{(i)}$ indicates the i-th ordered components of vector $\vec{x}$. If sorting of components is achieved by means of the quicksort algorithm (an $O(p \log p)$-comparisons algorithm), it can be readily seen that the asymptotical complexity of the AF-VM_2 filter is $O(n^3)$ multiplications, additions and comparisons, that is the same of the VM_2 filter except for the evaluation of $O(n^3)$ square roots.

As concerns 1-norm vector median filtering, a fast algorithm has been proposed recently in [7]. Again, the exact description of the algorithm and the discussion of its complexity would be too a long task to fit in this brief note, however, the basic idea the algorithm relies on is very simple and will be outlined below. Given a point $\vec{x}$ in R^p, the sum of distances between $\vec{x}$ and the window samples is considered as a cost functional $f(\vec{x})$. Let $\vec{x}_m$ be the point where $f(\vec{x})$ assumes its absolute minimum. If $\vec{x}_m$ is known and the difference $d_i = f(\vec{x}_i) - f(\vec{x}_m)$ can be computed easily for each point $\vec{x}_i$ of the filter window, then a fast algorithm is achieved by minimizing d_i instead of $f(\vec{x})$. In this way, in fact, only n^2 differences must be computed thus leading to an $O(n^2)$ algorithm. Of course, this is true only if the time spent to compute $\vec{x}_m$ is negligible with respect to the overall computation time. Indeed, this is always the case, since the non-constrained minimization of $f(\vec{x})$ corresponds to the componentwise application of the scalar median [5], which is a very fast operator [8]. For a discussion of how the d_i's can be calculated the reader is referred to [7], here it is only important to point out that the 1-norm fast median filter (F-VM_1 filter) can be split into 2 parts: first the scalar median is componentwise applied, then the differences d_i's are computed and the minimum one is selected. With regard to the first part, it can be shown that only $O(n)$ comparisons are required [8]. The second part, instead, requires the computation of n^2 d_i's. In [7] it is also demonstrated that each d_i can be computed by means of $O(p)$ additions only, thus yielding an overall asymptotic complexity of $O(n^2)$ additions and comparisons. Indeed, the F-VM_1 filter represents a significant improvement with respect to the classical implementation of VM filters.

3. MARGINAL VM-FILTERS

The simplest way to extend the ranking of a set of samples to the R^p case is marginal ordering [2] [4]. In marginal ordering, multivariate samples are ordered independently along each dimension, that is:

$$\begin{aligned} &x_{(1)_1,1} \le x_{(2)_1,1} \le \dots \le x_{(N)_1,1} \\ &x_{(1)_2,2} \le x_{(2)_2,2} \le \dots \le x_{(N)_2,2} \\ &\dots \\ &x_{(1)_p,p} \le x_{(2)_p,p} \le \dots \le x_{(N)_p,p} \end{aligned} \tag{3}$$

Table 1: Number of elementary operations required to evaluate a distance in R^p. Absolute values and comparisons are considered togheter

	Distance metric			
Operation	2-norm	squared 2-norm	1-norm	approx. 2-norm
Square roots	1	-	-	-
Mult. & div.	p	p	-	p
Additions	$2p$	$2p$	$2p$	$2p$
Comparisons	-	-	p	$p + O(\log p)$

Table 2: Complexity of VM filters. For the F-VM$_1$ case the signal is assumed to be uniformly distributed between 0 and a

	Asymptotic complexity			
Filter type	Square roots	multiplications	additions	comparisons
VM$_2$	$O(n^3)$	$O(n^3)$	$O(n^3)$	$O(n^2)$
VM$_2^2$	-	$O(n^3)$	$O(n^3)$	$O(n^2)$
VM$_1$	-	-	$O(n^3)$	$O(n^3)$
F-VM$_2^2$	-	$O(n^2)$	$O(n^2)$	$O(n^2)$
AF-VM$_2$	-	$O(n^3)$	$O(n^3)$	$O(n^3)$
F-VM$_1$	-	-	$O(n^2)$	$O(n^2)$

where $x_{(i)_j,j}$ is the j-th component of the i-th ranked samples, where ranking is performed with respect to the j-th component. Marginal ordering Vector Median filtering (M-VM filter) is obtained by selecting for each component the $(\nu+1)$-th ranked sample $(2\nu+1=N)$, i.e. by component-wise applying the scalar median filter

$$\vec{x}_{M-VM} = \left(x_{(\nu+1)_1,1}, x_{(\nu+1)_2,2}, \ldots x_{(\nu+1)_p,p}\right) \qquad (4)$$

Note that the output of the M-VM filter may not correspond to any of the input samples. The complexity of the M-VM filter is easily derived by noting that it corresponds to p applications of the scalar median. On the other hand, in the common case in which sample values are integer numbers, a very fast algorithm can be used to implement the scalar median filter (running median [8]) . According to it, the histogram of the points inside the window is built and the median of the histogram is chosen as the filter output. More specifically, at the beginning of each row the histogram is built from scratch, whereas at the other locations the new histogram is obtained by updating the old one according to the values of the points entering and leaving the window. Also the median of the histogram is not computed from scratch at each window location, instead the number of points lying on the left of the old median is continuously updated, and the median position is moved to the left or to the right according to the number of points currently on its left. By noting that at each new location n points out of n^2 are changed, and that only comparisons are needed to update both the median and the window histogram, it can be argued that the complexity of the M-VM filter is equal to $O(n)$ comparisons, thus leading to very fast operations.

Many modifications to the basic M-VM filter have been proposed to achieve better performance in presence of Gaussian noise: e.g. the α-trimmed Vector Mean filter (M-αVM filter,) the Vector Modified Trimmed Mean filter (V-MTM filter), the Vector Double-Window Modified trimmed Mean (V-DW-MTM filter) [4], however, truly speaking most of these filters can not be considered as real vector medians, and will not be considered further.

4. REDUCED VM-FILTERS

The last class of multivariate median filters is obtained by considering reduced ordering (Reduced ordering Vector Median filters, R-VM filters) [3]. In reduced ordering, multivariate samples are ordered according to their distance to a given central point. Many different schemes can be defined according to the choice of the central point and to the metric employed to calculate distances to it. Let us begin with the R-VM$_{2,mean}$ filter, that is an R-VM filter which uses the sample mean as the central point and the (squared) Euclidean metric to compute distances (note that, since distances to the central point are only used to sort samples, the squared Euclidean metric can be used instead of the linear one, which is more complex to calculate). For each window location the sample mean has to be computed first, by using a running algorithm this can be accomplished by means of $2np$ additions and p divisions. In addition, pn^2

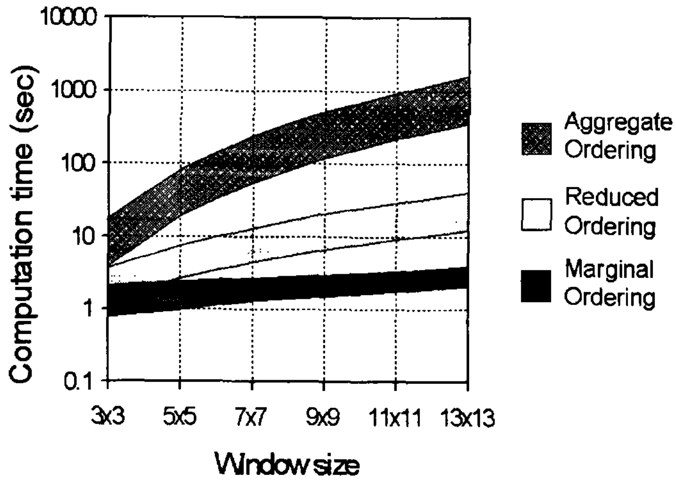

Figure 1: Computation time of some multivariate median filters. The shaded areas depicts the range where the computation time of filters based on different sorting schemes lies. Data refer to 256 × 256 RGB images.

multiplications, $2pn^2$ additions and n^2-1 comparisons must be performed to compute distances and to choose the minimum distance sample. Similar considerations hold for the R-VM$_{1,mean}$ filter, i.e. a filter which orders samples according to their distance to the window sample mean. $2np$ additions and p divisions are required to update the sample mean, while $2pn^2$ additions, pn^2 absolute value and n^2-1 comparisons are needed to choose the minimum distance point. When the marginal median is used as the central point, the R-VM$_{2,med}$ and the R-VM$_{1,med}$ filters result. In both cases, the first step consists in the application of the M-VM operator, whose complexity has been shown to be equal to $pO(n)$ comparisons. Then, n^2 distances must be evaluated and the point for which the distance is minimum selected. As before, this calls for the computation of pn^2 multiplications, $2pn^2$ additions and n^2-1 comparisons for the R-VM$_{2,med}$ case, and $2pn^2$ additions, pn^2 absolute values and n^2-1 comparisons in the R-VM$_{1,med}$ case. An interesting solution for the choice of the central point has been proposed by Tang et al. [9], which introduced the VR$_1$ filter as the sample in the window which minimizes the sum of distances to $\vec{x}_a$, $\vec{x}_m$ and $\vec{x}_{n^*}$:

$$\begin{cases} \vec{x}_{VR_1} \in \{\vec{x}_1, \vec{x}_2 ... \vec{x}_N\} \\ \vec{x}_{VR_1} = arg \min_{i=1...N} \{\| \vec{x}_a - \vec{x}_i \|_2 + \\ \qquad \| \vec{x}_m - \vec{x}_i \|_2 + \| \vec{x}_{n^*} - \vec{x}_i \|_2\} \end{cases} \quad (5)$$

where $\vec{x}_{n^*}$ is the window central point. The application of the VR$_1$ filter requires the sample mean and the marginal median to be computed at each window location, which in turn requires $2np$ additions, p divisions and $O(n)$ comparisons; besides, for each window sample, the sum of 3 Euclidean distances must be evaluated, i.e. $3n^2$ square roots, $3pn^2$ multiplications and $6pn^2$ additions. At last, n^2-1 comparisons are needed to pick the sample with minimum sum of distances. The overall complexity, then, is still $O(n^2)$, but the presence of $3n^2$ square roots makes the algorithm less attractive from a computational point of view. A possible solution to get around the problem, has been advanced by Tang et al. [9] which suggest to use squared distances, that is:

$$\begin{cases} \vec{x}_{VR_2} \in \{\vec{x}_1, \vec{x}_2 ... \vec{x}_N\} \\ \vec{x}_{VR_2} = arg \min_{i=1...N} \{\| \vec{x}_a - \vec{x}_i \|_2^2 + \\ \qquad \| \vec{x}_m - \vec{x}_i \|_2^2 + \| \vec{x}_{n^*} - \vec{x}_i \|_2^2\} \end{cases} \quad (6)$$

which can be reduced to the more pleasant form

$$\begin{cases} \vec{x}_{VR_2} \in \{\vec{x}_1, \vec{x}_2 ... \vec{x}_N\} \\ \vec{x}_{VR_2} = arg \min_{i=1...N} \{\| \vec{x}_c - \vec{x}_i \|_2^2\} \end{cases} \quad (7)$$

with $\vec{x}_c = (\vec{x}_a + \vec{x}_a + \vec{x}_{n^*})/3$. In this way the filter complexity is noticeably reduced, since in addition to the computation of the sample mean, the marginal median and their linear combination, only n^2 squared Euclidean distance must be computed. The overall filter complexity, then, is pn^2 multiplications, $2p$ divisions, $2p(n^2+n+1)$ additions and $n^2-1+O(n)$ comparisons.

A slightly different approach to R-VM filtering has been proposed by Hardie and Arce [3] (VR$_E$ filters). According to their proposal, the samples are first sorted according to reduced ordering, but $\vec{x}_{n^*}$ is substituted only if its distance to the central point used for sorting is larger than that of the sample with a predefined rank. Though the filtering behavior of the VR$_E$ filter has some original characteristics, from the point of view of computational complexity it behaves as the R-VM filters analyzed above.

5. TESTS AND COMPARISONS

The above theoretical analysis has been validated through exhaustive testing. All the multivariate filters have been implemented and their computing requirements measured.

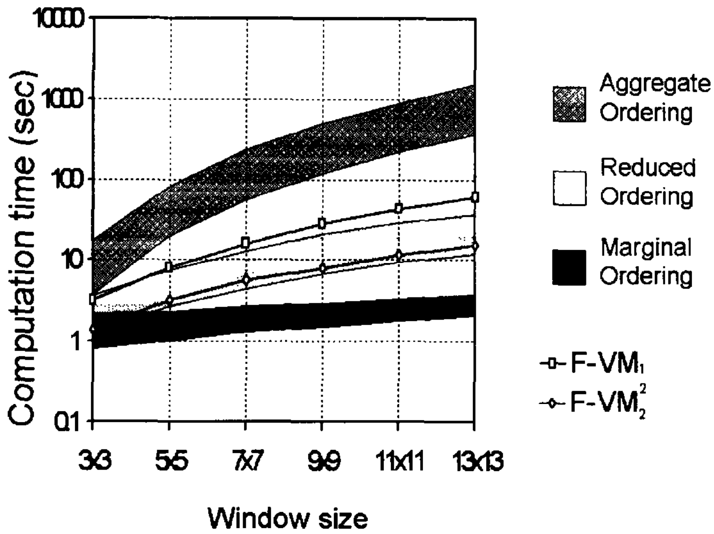

Figure 2: Fast algorithms permit to reduce significantly the computation time of VM filters. Data refer to the filtering of a 256×256 RGB images.

Filters have been applied to a 256×256 RGB image (the popular *pepper* image) corrupted with contaminated Gaussian noise ($\sigma = 20$, spike rate $= 5\%$). Noise has been added independently to each image component. Filters have been run on a DecStation 5000/240. The results we have obtained are reported in the diagrams of figures (1) and (2). In figure (1) the computational complexity of the various classes of filters is depicted. The range of complexities characterizing the filters of each class is highlighted by shadowing the area between the lowest and the largest computing time for that class. With only few exceptions (V-MTM, V-DW-MTM and VR_1 filters), filters belonging to the class of marginal medians have the lowest complexity, whereas VM filters are characterized by very large computing time. As expected, the complexity of multivariate medians based on reduced ordering is half-way between those of VM and M-VM filters.

In some cases the use of fast algorithms permits to lower the computational burden considerably. This is the case of the F-VM_1 and F-VM_2^2 filters. In figure (2) the computation saving achievable through these filters is pointed out. As it can be seen, the use of the fast algorithms permits to reduce the computational complexity to that typical of R-VM filters. This is quite obvious for the F-VM_2^2 filter, since in [5] it is demonstrated that such a filter is equivalent to the R-$VM_{2,mean}$ filter. On the contrary, the F-VM_1 filter represents a major improvement, since it permits to achieve the performance of classical vector medians at a computing cost which is typical of R-VM filters.

6. REFERENCES

[1] J. Astola, P. Haavisto and Y. Neuvo, "Vector Median Filters", *Proceedings of the IEEE*, vol. 78, no. 4, pp. 678–689, April 1990.

[2] V.Barnett, "The ordering of multivariate data", *Journal of the Royal Statistical Society A*, vol. 139, pp. 318–354, 1976.

[3] R. C. Hardie and G. R. Arce, "Ranking in R^p and its use in multivariate image estimation", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 1, no. 2, pp. 197–209, June 1991.

[4] I. Pitas and P. Tsakalides, "Multivariate ordering in color image filtering", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 1, no. 3, pp. 247–259, September 1991.

[5] M. Barni, V. Cappellini and A. Mecocci, "The use of different metrics in vector median filtering: application to fine arts and paintings", in *Signal Processing VI: Theories and Applications*, pp. 1485–1488, Brussels, Belgium, August 1992.

[6] M. Barni, V. Cappellini and A. Mecocci, "Fast vector median filter based on Euclidean norm approximation", *IEEE Signal Processing Letters*, Vol. 1, No. 6, pp. 92–94, June 1994.

[7] M. Barni and V. Cappellini, "A computationally efficient implementation of the L_1 vector median filter", *Proceedings of DSP'97 International Conference* Santorini, Greece, 2-4 July 1997.

[8] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*, Kluwer Academic Publishers, 1990.

[9] K.Tang, J.Astola and Y.Neuvo, "Nonlinear multivariate image filtering techniques", *IEEE Trans. on Image Proc.*, vol. 4, no. 6, June 1995.