ORDER FILTER WITH PROGRESSIVELY DECIMATED FILTERING WINDOW: APPLICATION TO COLOUR IMAGE ENHANCEMENT

Jocelyn Chanussot

Signal and Images Laboratory
LIS / ENSIEG – BP 46 – 38402 Saint Martin d'Hères Cedex – France
jocelyn.chanussot@lis.inpg.fr
(CNRS GdR ISIS – Information Signal Image viSion)

ABSTRACT

The classical definition of vector order filters consists in selecting the pixel that **minimizes** the cumulated distance to the other pixels of the filtering window. The "most representative" pixel of the filtering window is then selected and the noise is thus reduced. But, when the filtering window reaches a transition, the result is biased and the smoothing is not optimal anymore. The proposed technique consists in progressively decimating the filtering window by suppressing the pixels that **maximize** the cumulated distance. The process is then iterated (computation of the cumulated distances among the remaining pixels and suppression of the less typical pixels) until one single pixel remains : it is then selected for the filter output. In practical cases, the number of iterations is limited. Applied on colour images, the filter results in an enhancement of the edges and in an automatic registration of the different components. The counterpart is a slight degradation of the smoothing performances in homogeneous regions. The paper details the proposed method and compares it with other enhancement filters.

1. INTRODUCTION

Vector order filters are classically obtained by minimizing a cumulated local distance within the filtering window: the vector that is the "closest" to the set of the other vectors of the filtering window is considered as the most typical and representative vector. It is therefore selected and becomes the filter output. The construction of these filters requires the definition of an appropriate distance between vectors [1][2].

To compute a distance between two vectors X and Y, different norms of their difference *(Y-X)* can be used. In that way, we obtain the following generic expression for the cumulated distance between a vector Y and the set $\{Xj, j=1...N\}$ of the N vectors of the filtering window :

$$\varepsilon(Y)=\sum_{j=1}^{N}\left\|Y-X_j\right\| \qquad \text{(Eq. 1)}$$

where $\|\ \|$ represents a given norm. Classically, the filter output is the value Y that minimizes the cost function ε :

$$Y_{output}=\underset{Y}{\operatorname{argmin}}\{\varepsilon(Y)\} \qquad \text{(Eq. 2)}$$

Depending on the norm used to compute the difference between two pixels, different effects can be obtained (optimal filtering for different kind of noises, enhancement). Furthermore,

- For instance, taking the euclidean (or L_2) norm leads to the simple vector mean filter,
- L_1 norm leads to the classical vector median filter,
- L_∞ norm leads to the vector extension of the middle filter.

These filters are the **maximum likelihood estimators** in the case of images respectively corrupted by gaussian, exponential and uniform additive noise. They are therefore optimal in terms of mean square error minimization. A keypoint is that **this optimality is obtained with stationary signals**, corresponding to homogeneous regions. When the filtering window laps over two different regions, this assumption is not valid anymore. In an image, transitions correspond to non stationary regions and the considered filters are not optimal anymore : the result is biased. This is illustrated by the following figures.

Figure 1 presents the case of a filtering window of size 3x3 lapping over two different regions. For the sake of simplicity for the schematic representation, we assume that this picture has two components (the value of each pixel is a bi-dimensional vector). Six pixels of the filtering window belong to the left region. They correspond to the left cluster of values on figure 2. The three other pixels belong to the right region. Corresponding values constitute the upper right cluster on figure 2. Considering, for instance, the classical vector median filter, the output in that case is value 1 pointed on figure 2 (this value minimizes the cumulated distance to the other values of the filtering window). Nevertheless, value 2 is obviously the most typical value for the left region. But it has not been selected : the result has been biased by the right cluster.

The aim of the method proposed in this paper is to solve this problem.

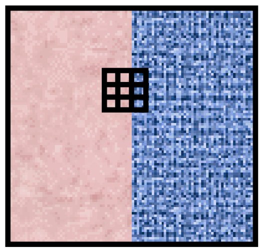


Figure 1 : 3x3 filtering window lapping over two regions

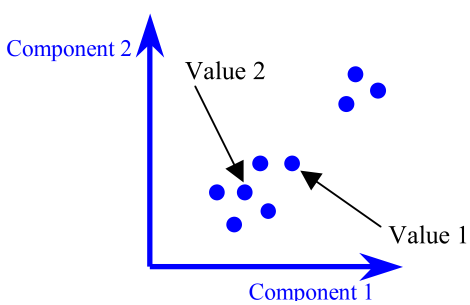


Figure 2 : Schematic representation of the pixels of the 3x3 filtering window presented in figure 1

2. PROPOSED ALGORITHM

To avoid the phenomenon previously described, the proposed method consists in progressively decimating the filtering window. This is performed in two steps that are iterated:

❶ the pixel Y **maximizing** the cumulated distance is removed from the filtering window :

$$Y = \underset{y}{\arg\max} \sum_{i=1}^{N} \| Y - X_i \| \qquad \text{(Eq.3)}$$

❷ the cumulated distances corresponding to the *N-1* remaining pixels are computed (Note that the suppression of one value can modify the relative cumulated distances between the remaining vectors).

❸ the process is iterated : one more pixel is removed and the remaining cumulated distances are computed.

The pixel defined by (Eq. 3) may not be uniquely defined. In that case, i.e. if for a given iteration several pixels maximize the cost function, they are all removed from the filtering window. In the next step, the cumulated distances are computed between the remaining pixels.

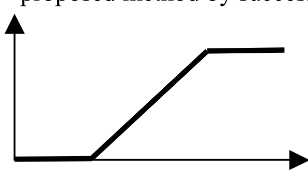
For a given filter (i.e. a given vector norm) and for a given size of filtering window, we propose to improve the filter with the algorithm presented above. The only parameter that is introduced is the number of iterations that is performed.

3. ADVANTAGES

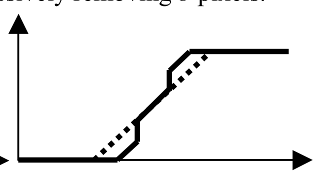
The advantages of the proposed improvement are located near the transitions :

The smoothness classically introduced on the transitions by the filters (even with the median filter if the image is noisy) is suppressed. Furthermore, a strong enhancement effect can be obtained.

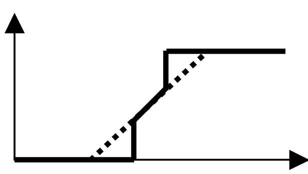
This is illustrated by the following figure, in a mono-dimensional scalar case for the sake of simplicity. A fuzzy transition between two homogeneous regions is considered (a). This transition is left unchanged by a median filtering. The width of the transition is q pixels. A filtering window of size N is used and the filter is modified with the proposed method by successively removing b pixels.



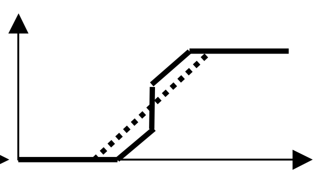
(a) original transition



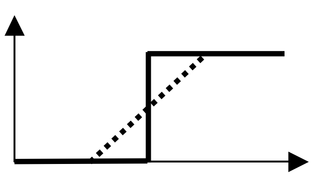
(b) general case



(c) $N<q$ & $b=N-1$



(d) $N>q$ & $b<q$



(e) $N>q$ et $b>q-1$

Figure 3 : effect on a fuzzy transition

In the most general case (b), the transition is enhanced, but three oblique sections remain. The two extreme ones can be suppressed when enough pixels are removed by the proposed algorithm (c). The central oblique section can be suppressed when the filtering window is big enough (d). When both conditions are fulfilled, the transition becomes perfectly sharp, with an optimum enhancement (e).

It is important to underline that the same result holds whatever shape has the transition (assuming it is monotonous): any transition can be perfectly enhanced, every edge becoming perfectly sharp.

Another interesting property lies in the **registration effect** that is also obtained : if a shift of one component (or more) appeared during the acquisition, it is automatically removed (still assuming the filtering window is wide enough, meaning wider than the maximum shift). This registration effect is illustrated in section 5.

4. DRAWBACKS

The proposed method can be applied to any L-filter following the general expression given by (Eq. 1). These filters are optimal (in the sense of maximum likelihood estimators minimizing the average square error for various additive noise distributions). Therefore, the proposed modification leads to sub-optimal performances in terms of noise reduction in stationary regions: there is a slight degradation of the smoothing performances in the homogeneous regions.

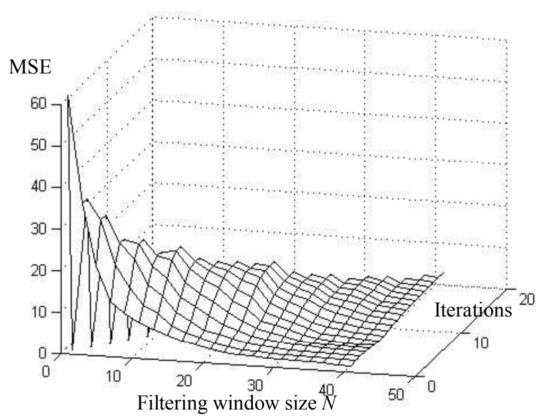


Figure 4 : evolution of the MSE

This degradation is illustrated by figure 4. In the case of a median filtering on a homogeneous signal corrupted by an additive noise with an exponential distribution, we see the evolution of the MSE with respect to the size of the filtering window and with respect to the number of iterations performed (i.e. the number of pixels successively removed from the filtering window).

The first curve, corresponding to "iterations = 0", describe the performances of the original filter (in this case the median filter, but let us recall that the same principle holds for any L-filter). The MSE decreases as the size of the filtering window increases.

For a fixed filtering window size, the MSE curve increases as the "iterations" parameter increases. This is the price we pay for the proposed modification: the more pixels are removed, the more we move away from the original optimal filter. The filter becomes more and more sub-optimal. This difference between the original filter and the modified filter decreases with the size of the filtering window. As a matter of fact, when many samples are considered at the same time, the assumption of a zero-mean additive noise becomes more and more valid and the suppression of pixels is less problematic: the less typical values are uniformly distributed around the data cluster.

A consequence of this study is the following tradeoff : the number of iterations performed is set to the half of the filtering window. The advantages are preserved and the drawbacks are limited. Looking at figure 1, this choice can be intuitively understood: in typical situations, less than one half of the pixels belong to another region than the current pixel. Non typical situations, such as narrow corners, are not correctly handled by classical algorithms

The second drawback of the proposed modification is the increase of the computation time. The cumulated distances between the remaining pixels have to be computed after each iteration (the removal of one pixel changes the relative order of the different values). An efficient implementation of the proposed modification consists in using a "look up array" where, for one initial filtering window, all the distances are stored. After the suppression of one pixel, the remaining distances are not re-computed: the contribution of the suppressed pixel is simply substracted (computational charge of the first iteration : N substractions and computation of the new maximum (among N-1 values) for the next pixel to be removed). This is illustrated by figure 5.

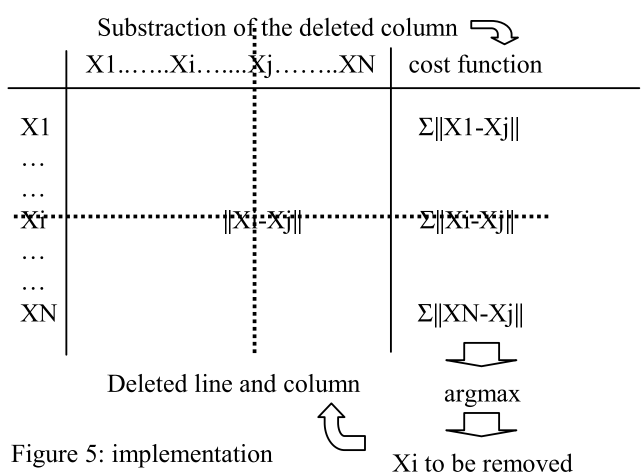


Figure 5: implementation

5. RESULTS & CONCLUSIONS

Results are presented on figure 6. Fig. 6 (a) presents an original 267x200 colour image and a zoom. A shift of the blue and the red components is artificially introduced (translation of magnitude 2 pixels). See (b) and note the appearance of false colours (light blue and green around the main dark blue feature on the zoom image). This image is filtered with 5x5 standard vector median (c) and with a modified vector median following our approach (d). The enhancement effect is clearly visible on (d), as well as the registration effect (note that the false colours have disappeared).

It is important to understand that any vector order filter could be modified with the same approach. Even filters dedicated to vector enhancement (such as dαβ filter with parameter α<1 [4]) can be modified leading to a stronger enhancement effect. The modified filter preserves the properties of the original filter (which ever it is, median or any other L-filter) except a slight degradation of the smoothing performances in stationary areas.

The proposed method is very simple, applicable to many filters, with known and bounded drawbacks. Aiming at suppressing the bias appearing in the estimation of the most typical value around transitions, it provides a nice enhancement of the contrast and a registration of the different components. No parameter or statistical model need to be estimated, it is thus very robust.

Note that the idea of reducing the filtering window by suppressing atypical pixels is not new. For instance, the α-trimmed filters already used it [3]. Nevertheless, our approach is more robust since its contrast enhancement property is independent from the local range of the transitions (unlike the α-trimmed filters where the setting of parameter α is linked to the local range and to the standard deviation of the noise). See also [5] for recent work on contrast enhancement and [6] for another strategy based on order filters.

6. REFERENCES

[1] J. Astola, P. Haavisto & Y. Neuvo – Vector median filters – Proc. of the IEEE – vol. 78 n. 4 – 1990 – pp 678-689

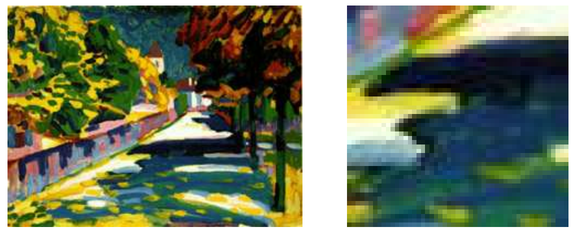
[2] I. Pitas & A.N. Venetsanopoulos – Nonlinear digital filters – Kluwer Academic Publishers, 1990

[3] S.R. Petterson, Y.H. Lee & S.A. Kassam – Some statistical properties of α-trimmed mean and standard type median filters – IEEE Trans. ASSP- vol. 36 n. 5 – 1988 – pp 707-713

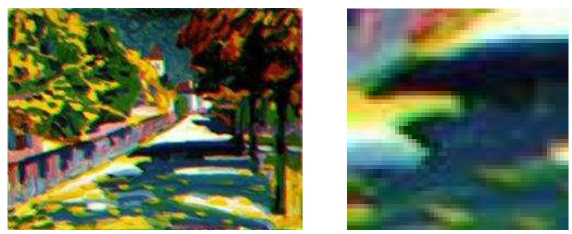
[4] W. Alshatti, Ph. Bolon & P. Lambert - A class of non linear vector filters: Application to multi-spectral image filtering - 7th European Signal & Image Processing COnference (EUSIPCO'94) - Edimburgh, U.K, 1994, pp. 860-863

[5] HD. Cheng, M. Xu & XJ. Shi – Contrast enhancement based on a novel homogeneity measurement – Pattern Recognition Letters – vol 36 – 2003 – pp 2687-2697

[6] P. Lambert & S. Gaspard - Enhancement and registration of multichannel images based on vector order filter - 10th European Signal & Image Processing COnference (EUSIPCO´2000) - Tampere, Finland, 2000, pp. 1357-1360.

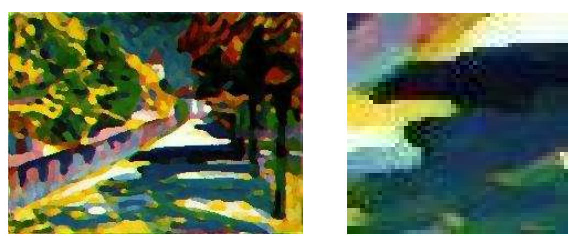


(a) original image



(b) original image with shifted components

(c) result of the standard median filter



(d) result of the modified median filter

Figure 6: results on a colour image