

Algorithmes Rapides pour une Classe de Filtres Non-linéaires Vectoriels

Laurent Lucat et Pierre Siohan

CNET/DSM, Laboratoire du CCETT, Groupe France Télécom
4, rue du Clos Courtel, BP 59, 35512 Cesson Sévigné Cédex, France
E-mail : llucat@ccett.fr

RÉSUMÉ

Les filtres non-linéaires vectoriels, du type médian vectoriel, impliquent un classement des N vecteurs d'entrée selon une mesure de "centralité/atypisme" associée à chaque vecteur. Le calcul des N mesures étant de complexité $O(N^2)$, l'utilisation de ces filtres avec des masques de taille conséquente pose un sérieux problème de charge calculatoire.

Grâce à une nouvelle formulation mathématique de cette mesure (définie avec la norme L_1), nous proposons une technique de calcul rapide des N mesures, qui peut être utilisée pour l'ensemble de ces filtres.

ABSTRACT

Vector non-linear filters, such as the vector median filter, involve a sorting of N input vectors according to a centrality criterion associated to each vector. The computation of the N ordering criteria having a $O(N^2)$ complexity, the use of these filters with large masks leads to a critical computational burden.

Owing to a new mathematical formulation of the criterion (with the L_1 metric), we derive a fast algorithm for computing the N criteria, which can be used for every filter of this class.

1 Introduction

L'utilisation en traitement d'images du filtre médian et des filtres dérivés connaît un large succès, dû notamment à leur excellente performance en réduction du bruit impulsif et à leur capacité à préserver les contours non bruités. Afin de prendre en considération les spécificités des signaux multi-composantes (tels que les images en couleurs), notamment la corrélation qui existe entre les composantes, le filtrage non-linéaire de type médian a été étendu au cas des signaux multi-composantes, donnant naissance à une famille de filtres non-linéaires vectoriels, dont le plus connu est certainement le filtre médian vectoriel [1]. Contrairement au filtrage "marginal" qui consiste à appliquer un filtre indépendamment sur chaque voie, le filtrage vectoriel opère globalement sur toutes les composantes du signal. Dans le cas d'un opérateur de type médian, l'approche vectorielle permet par exemple d'obtenir de meilleurs résultats en réduction de bruit de type impulsif en présence de contours.

L'extension du filtrage médian au cas vectoriel implique que l'on se donne une règle pour trier les échantillons vectoriels d'entrée du filtre. On distingue classiquement 4 techniques de tri [4], basées sur l'ordre marginal (tri scalaire indépendant sur chaque composante), l'ordre conditionnel (tri scalaire des vecteurs selon une composante privilégiée), l'ordre partiel et l'ordre réduit. De nombreux filtres proposés dans la littérature utilisent l'ordre réduit ; il s'agit de trier les vecteurs selon un critère de "centralité / atypisme" de chaque vecteur vis-à-vis de l'ensemble des vecteurs considérés. Il existe plusieurs définitions précises de ce critère ; par la suite, nous désignerons par "mesure" la définition précise de ce critère. Dans

[6], nous proposons une classification des filtres non-linéaires vectoriels utilisant l'ordre réduit, basée sur les diverses mesures rencontrées dans la littérature.

Nous considérons ici la mesure (pondérée) d_w définie par :

$$d_w(x_j) = \sum_{i=1}^N w_i \|x_j - x_i\|, \quad (1)$$

où $\{x_i\}_{i=1..N}$ sont les vecteurs d'entrée, $\{w_i \in \mathbb{R}^+\}_{i=1..N}$ sont les poids associés aux vecteurs et $\|\cdot\|$ désigne une norme, habituellement L_1 ou L_2 .

Lorsque tous les poids sont égaux (à 1), ce qui équivaut à une absence de pondération, cette expression se simplifie, et la mesure non pondérée d s'écrit alors :

$$d(x_j) = \sum_{i=1}^N \|x_j - x_i\|. \quad (2)$$

Notons que d étant un cas particulier de d_w , toutes les formulations énoncées pour d_w sont également valables pour d .

Un frein important à l'utilisation des filtres non-linéaires vectoriels, impliquant la mesure d_w (ou singulièrement la mesure d), dans des systèmes temps réel est leur complexité algorithmique élevée, due à de nombreux calculs de distances. En effet, le calcul de $\{d_w(x_i)\}_{i=1..N}$ selon (1) implique le calcul des N^2 distances $\{\|x_j - x_i\|\}_{1 \leq i, j \leq N}$. En tenant compte de la symétrie $\|x_j - x_i\| = \|x_i - x_j\|$ et du fait que $\|x_i - x_i\| = 0$, le nombre de distances à calculer est réduit à $\frac{N(N-1)}{2}$. Malgré cette simplification, le calcul de $\{d_w(x_i)\}_{i=1..N}$ reste néanmoins de complexité $O(N^2)$. Ainsi, tous les filtres faisant appel à la mesure d_w , telle qu'explicitée en (1), sont de complexité minimale $O(N^2)$. Aussi, la recherche d'algorithmes

rapides pour ces filtres, visant à exploiter la redondance persistante dans le calcul de $\{d_w(x_i)\}_{i=1..N}$ selon (1), s'avère-t-elle du plus grand intérêt.

Astola [1] a proposé un algorithme rapide de complexité $O(N)$ utilisable lorsque le masque est monodimensionnel et lorsqu'un seul nouveau vecteur apparaît à chaque décalage du masque. Cette approche perd de son intérêt si les vecteurs d'entrée sont pondérés ou si on essaie de l'étendre au cas des masques bi- voire tridimensionnels.

Barni a développé des algorithmes rapides pour le filtre médian vectoriel standard : un algorithme approché dans le cas de la norme L_2 [2] et un algorithme exact dans le cas de la norme L_1 [3]; cependant, pour ce dernier, sa méthode n'est pas utilisable lorsque les vecteurs sont pondérés.

Nous proposons ici une technique originale de calcul du jeu de mesures $\{d_w(x_i)\}_{i=1..N}$ (définies avec la norme L_1), basée sur une réécriture de l'expression (1), et qui donne lieu à des algorithmes rapides pour les filtres utilisant cette mesure. L'expression de $d_w(x_i)$ obtenue étant équivalente à l'originale, les algorithmes proposés sont donc exacts.

Au d'2, nous présentons les filtres connus utilisant la mesure d_w (ou singulièrement d), c'est-à-dire les filtres pour lesquels le procédé de calcul rapide proposé est utilisable. Une nouvelle écriture mathématique de d_w (et d) est proposée au d'3, et cette nouvelle expression sera utilisée au d'4 pour l'obtention d'algorithmes rapides pour les filtres considérés. Nous illustrerons ensuite l'efficacité de la technique de calcul proposée par quelques résultats de simulation logicielle.

2 Filtres utilisant la mesure d_w

Les filtres non-linéaires vectoriels que nous considérons ici sont ceux qui peuvent être représentés sous la forme :

$$y = F(\{x_i\}_{i=1..N}, \{d_w(x_i)\}_{i=1..N}, \dots) \quad (3)$$

où y représente la sortie du filtre et F est une fonction spécifique à chaque filtre. Par la suite, nous citons quelques exemples de filtres issus de la littérature, et pouvant s'écrire sous cette forme.

- Le filtre médian vectoriel standard (VMF) est défini par :

$$y_{VMF} = \arg \min_{x_j} d(x_j). \quad (4)$$

- Le VMF est un cas particulier de filtre médian vectoriel pondéré (WVMF) :

$$y_{WVMF} = \arg \min_{x_j} d_w(x_j). \quad (5)$$

- Le filtre médian vectoriel étendu (EVMF) [1] choisit en sortie un des vecteurs d'entrée ou bien leur moyenne :

$$y_{EVMF} = \arg \min_{x \in X_1} d(x),$$

avec : $X_1 = \{x_1; \dots; x_N; x_{moy}\}$ et $x_{moy} = \frac{1}{N} \sum_{i=1}^N x_i$.

- Le filtre médian vectoriel pondéré étendu (EWVMF) [7] correspond à la version pondérée du EVMF :

$$y_{EWVMF} = \arg \min_{x \in X_2} d_w(x),$$

où : $X_2 = \{x_{1..N}; x_{moyp}\}$ et $x_{moyp} = \frac{1}{\sum_{i=1}^N w_i} \sum_{i=1}^N w_i x_i$.

- Une variante du EWVMF est le filtre médian vectoriel pondéré à moyenne α -tronquée (α -TWVMF), où les valeurs les plus atypiques sont écartées lors du calcul de la moyenne. Sont également concernés les filtres d'ordre vectoriels, tels que définis en [5]. Les filtres médians vectoriels hybrides et médians vectoriels à étages utilisent également la mesure d_w , mais ils sont souvent utilisés avec des fenêtres de très petite taille, réduisant ainsi la charge de calcul.

3 Nouvelles écritures des mesures d et d_w (norme L_1)

La définition de d_w (équation (1)), déclinée avec la norme L_1 , s'écrit :

$$d_w(x_j) = \sum_{i=1}^N w_i \sum_{k=1}^P |x_j^k - x_i^k|,$$

où P désigne le nombre de composantes, et x_i^k est la k ième composante du vecteur x_i . Adoptons les notations suivantes :

$$\begin{aligned} X^k &= \{x_i^k\}_{i=1..N}, & W &= \sum_{i=1}^N w_i, \\ A_j^k &= \{x_i^k | x_i^k \leq x_j^k\}_{i=1..N}, & B_j^k &= X^k - A_j^k, \\ D_w^k &= \sum_{i=1}^N w_i x_i^k, & D_w &= \sum_{k=1}^P D_w^k. \end{aligned}$$

L'expression de d_w peut ainsi s'écrire :

$$\begin{aligned} d_w(x_j) &= \sum_{k=1}^P \sum_{x_i^k \in X^k} w_i |x_j^k - x_i^k| \\ &= \sum_{k=1}^P \left\{ \sum_{x_i^k \in A_j^k} w_i (x_j^k - x_i^k) + \sum_{x_i^k \in B_j^k} w_i (x_i^k - x_j^k) \right\} \\ &= \sum_{k=1}^P \left\{ x_j^k \left(\sum_{A_j^k} w_i - \sum_{B_j^k} w_i \right) - \sum_{A_j^k} w_i x_i^k + \sum_{B_j^k} w_i x_i^k \right\} \\ &= \sum_{k=1}^P \left\{ x_j^k \left(2 \sum_{x_i^k \in A_j^k} w_i - W \right) - 2 \sum_{A_j^k} w_i x_i^k + D_w^k \right\} \end{aligned}$$

Si $rg(x_j^k)$ désigne le rang de x_j^k selon le tri scalaire de X^k , $x_{(r)}^k$ l'élément de X^k de rang r , et $w_{[r]}^k$ le poids associé à $x_{(r)}^k$, i.e. $w_{[r]}^k = \{w_i | rg(x_i^k) = r\}$, d_w peut alors s'écrire :

$$d_w(x_j) = D_w + 2 \sum_{k=1}^P \left\{ x_j^k \left(\sum_{r=1}^{rg(x_j^k)} w_{[r]}^k - \frac{W}{2} \right) - \sum_{r=1}^{rg(x_j^k)} w_{[r]}^k x_{(r)}^k \right\}$$

Nous définissons ensuite les suites $\{W^k(R)\}_{R=1..N}$ et $\{S_w^k(R)\}_{R=1..N}$ par :

$$W^k(R) = \sum_{r=1}^R w_{[r]}^k - \frac{W}{2} \quad \text{et} \quad S_w^k(R) = \sum_{r=1}^R w_{[r]}^k x_{(r)}^k$$

L'expression de d_w peut alors se mettre sous la forme :

$$d_w(x_j) = D_w + 2 \sum_{k=1}^P \{x_j^k \cdot W^k(rg(x_j^k)) - S_w^k(rg(x_j^k))\} \quad (6)$$

Cette expression est valable dans le cas général, y compris lorsque tous les poids sont égaux à 1 (cas non pondéré). Néanmoins, dans ce dernier cas, on peut obtenir une formulation simplifiée. En posant $D^k = \sum_{X^k} x_i^k$, $D = \sum_{k=1}^P D^k$, et $S^k(R) = \sum_{r=1}^R x_{(r)}^k$, l'expression simplifiée s'écrit :

$$d(x_j) = D + 2 \sum_{k=1}^P \{x_j^k (rg(x_j^k) - \frac{N}{2}) - S^k(rg(x_j^k))\}. \quad (7)$$

4 Algorithmes rapides

Les filtres de la forme explicitée en (3) nécessitent le calcul préliminaire de $\{d_w(x_i)\}_{i=1..N}$ (calcul de complexité $O(N^2)$ selon (1)). Tous ces filtres peuvent bénéficier d'algorithmes rapides en intégrant un calcul rapide de $\{d_w(x_i)\}_{i=1..N}$.

Le calcul rapide de $\{d_w(x_i)\}_{i=1..N}$ que nous proposons est basé sur l'équation (6) (équation (7) dans le cas non pondéré) ; il peut être décomposé en 3 étapes :

1. un tri marginal de X (tri scalaire de chaque composante X^k), de complexité $O(PN \log N)$.
Les suites $\{x_{(R)}^k\}_{R=1..N}$ et $\{rg(x_i^k)\}_{i=1..N}$ sont alors connues pour chaque $k \in \{1; \dots; P\}$, et les P suites $\{w_{[R]}^k\}_{R=1..N}$ sont déduites de $\{w_i\}_{i=1..N}$ en effectuant les mêmes permutations que lors du tri de X^k .
2. un calcul récursif des suites W^k et S_w^k , de complexité globale $O(PN)$:

Pour $k = 1$ à P :

– initialisation :

$$\begin{aligned} W^k(1) &= w_{[1]}^k - \frac{W}{2} \\ S_w^k(1) &= w_{[1]}^k x_{(1)}^k \end{aligned}$$

– récurrence : pour $R = 2$ à N :

$$\begin{aligned} W^k(R) &= W^k(R-1) + w_{[R]}^k \\ S_w^k(R) &= S_w^k(R-1) + w_{[R]}^k x_{(R)}^k \end{aligned}$$

3. le calcul final de $\{d_w(x_i)\}_{i=1..N}$ selon (6), de complexité $O(PN)$. Le terme D_w peut être obtenu facilement par : $D_w = \sum_{k=1}^P D^k = \sum_{k=1}^P S_w^k(N)$, les termes $S_w^k(N)$, $k \in \{1; \dots; P\}$, étant obtenus à l'étape 2.

Ainsi, en utilisant l'expression de $d_w(x_i)$ que nous proposons, le calcul de $\{d_w(x_i)\}_{i=1..N}$ est de complexité $O(PN \log N)$ contre $O(PN^2)$ pour un calcul basé sur la définition standard de d_w . Le gain dû à la formulation que nous proposons réside dans la récursivité des suites W^k et S_w^k (S^k dans le cas non pondéré). En effet, si ces suites étaient quelconques (non récursives), leur calcul serait de complexité $O(N^2)$; grâce à la récursivité, leur calcul est de complexité $O(N)$, pour chaque $k \in \{1, \dots, P\}$.

En intégrant notre calcul rapide de $\{d_w(x_i)\}_{i=1..N}$ dans le calcul de la sortie des filtres non-linéaires vectoriels de la

forme (3), on peut ainsi espérer un gain en temps de calcul pour les filtres concernés, au moins pour des valeurs assez grandes de N .

Dans le cas non pondéré (cas particulier), le calcul rapide de $\{d(x_i)\}_{i=1..N}$ est similaire au calcul rapide de $\{d_w(x_i)\}_{i=1..N}$; la récurrence (unique) de l'étape 2 s'écrit :

Pour $k = 1$ à P :

$$- S^k(1) = x_{(1)}^k$$

– Pour $R = 2$ à N :

$$S^k(R) = S^k(R-1) + x_{(R)}^k$$

et l'étape 3 utilise l'équation (7).

Selon la fonction F particulière à chaque filtre, certaines simplifications sont possibles dans l'étape 3 du calcul de $\{d_w(x_i)\}_{i=1..N}$. Une représentation mathématique des versions rapides des filtres considérés peut être obtenue en explicitant (6) (ou (7)) dans la définition de chaque filtre, avec les simplifications éventuelles. Nous donnons par la suite des exemples de définition de filtres rapides ainsi obtenus.

- Le VMF rapide est obtenu en combinant (7) et (4), en omettant le terme D , indépendant de j :

$$y_{VMF}^{rap} = \arg \min_{x_j} \sum_{k=1}^P \{x_j^k (rg(x_j^k) - \frac{N}{2}) - S^k(rg(x_j^k))\}. \quad (8)$$

- De même, le WVMF rapide s'obtient en combinant (6) et (5), en délaissant le terme constant D_w :

$$y_{WVMF}^{rap} = \arg \min_{x_j} \sum_{k=1}^P \{x_j^k \cdot W^k(rg(x_j^k)) - S_w^k(rg(x_j^k))\}. \quad (9)$$

- Pour le EVMF rapide, on calcule la mesure $d^{min} = D + d^*$, où d^* est la mesure, tronquée du terme D , associée à y_{VMF}^{rap} (calculé selon (8)). Ensuite, la mesure associée à x_{moy} est calculée par :

$$d(x_{moy}) = \sum_{i=1}^N \sum_{k=1}^P \left| \frac{D^k}{N} - x_i^k \right|,$$

D^k étant égal à $S^k(N)$, déjà obtenu par récurrence. La sortie du EVMF rapide est alors donnée par :

$$y_{EVMF}^{rap} = \begin{cases} x_{moy} & \text{si } d(x_{moy}) \leq d^{min} \\ y_{VMF}^{rap} & \text{sinon.} \end{cases}$$

- De façon similaire, le EWVMF est obtenu par :

$$y_{EWVMF}^{rap} = \begin{cases} x_{moyp} & \text{si } d_w(x_{moyp}) \leq d_w^{min} \\ y_{WVMF}^{rap} & \text{sinon.} \end{cases}$$

$$\text{avec : } \begin{cases} d_w(x_{moyp}) = \sum_{i=1}^N w_i \sum_{k=1}^P \left| \frac{D_w^k}{W} - x_i^k \right|, \\ d_w^{min} = D_w + d_w^*(x_j), \end{cases}$$

où $d_w^*(x_j)$ est la mesure "tronquée" associée à y_{WVMF}^{rap} (calculé selon (9)).

Notons que, pour des filtres impliquant une fenêtre glissante, une amélioration supplémentaire des algorithmes est possible dans l'étape du tri marginal (étape 1 du calcul rapide de $\{d_w(x_i)\}_{i=1..N}$), où il n'est pas toujours nécessaire de retrier toutes les données à chaque décalage du masque. Si le gain à espérer dans le cas d'un masque monodimensionnel est non négligeable, l'intérêt décroît cependant avec la dimension du masque, et il dépend fortement de sa forme.

5 Simulations et conclusions

Afin d'évaluer les performances en temps de calcul des algorithmes proposés, des simulations de filtrage ont été réalisées, avec diverses tailles de fenêtre, sur des images couleurs (720 * 576 pixels). La figure 1 reporte les temps de calcul comparatifs pour plusieurs algorithmes de VMF : l'un basé sur la définition classique de d , engendrant le calcul de $\frac{N(N-1)}{2}$ distances, l'algorithme proposé par Barni et le nôtre. La figure 2 donne les temps de calculs pour les filtrages WVMF et EWVMF, comparant notre algorithme à l'algorithme "standard".

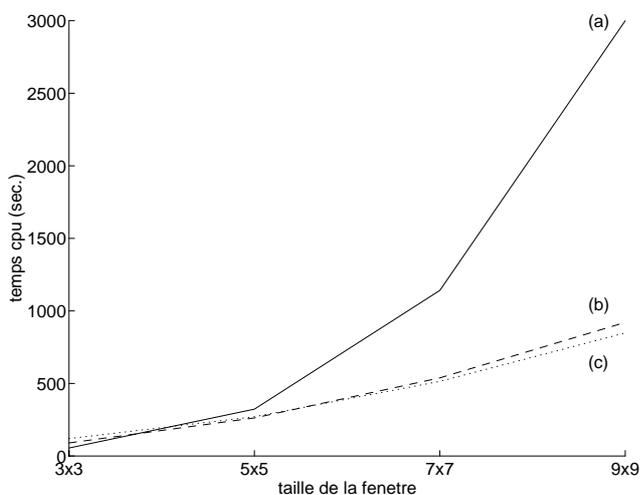


FIG. 1 — Temps de calcul pour le VMF : définition originale (a), notre algorithme (b), celui de Barni (c).

Ces simulations illustrent l'intérêt d'utiliser les algorithmes "rapides", dès lors que la fenêtre d'analyse dépasse la taille 3x3. Les performances de notre VMF rapide sont très proches de celles de l'algorithme proposé par Barni.

Un intérêt majeur de notre approche est sa "généricité", puisque tous les filtres non-linéaires vectoriels de la forme explicitée en (3) sont concernés par le procédé de calcul rapide, y compris les filtres utilisant une pondération des vecteurs d'entrée. Signalons que, par exemple, aucun algorithme rapide n'avait été proposé jusqu'ici pour le WVMF.

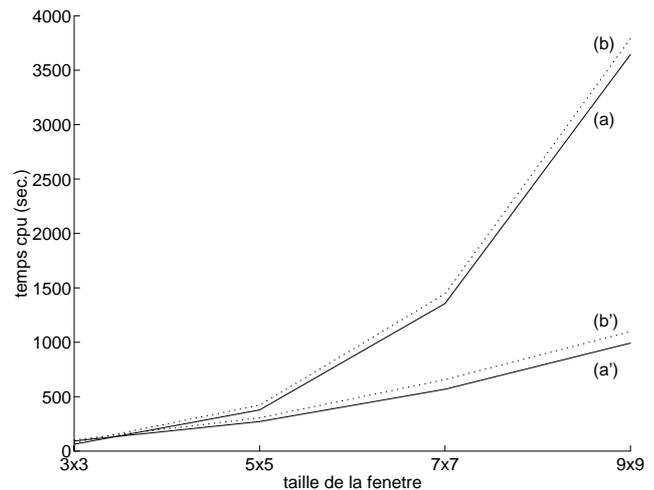


FIG. 2 — Temps de calcul pour le WVMF (a), EWVMF (b), et leur version rapide correspondante (a') et (b').

Références

- [1] J. Astola, P. Haavisto, and Y. Neuvo. "Vector Median Filters". In *Proceedings of the IEEE*, volume 78-4, pages 678–689, April 1990.
- [2] M. Barni, F. Bartolini, F. Buti, and V. Cappellini. "Optimum Linear Approximation of the Euclidean Norm to Speed Up Vector Median Filtering". In *Proc. of ICIP*, volume 1, pages 362–365, 1995.
- [3] M. Barni and V. Cappellini. "A Fast L1-Metric Vector Median Filter". *Quatorzième Colloque GRETSI, Juan-les-Pins*, pages 583–586, September 1993.
- [4] R.C. Hardie and G. Arce. "Ranking in R^p and its Use in Multivariate Image Processing". *IEEE Tr. on Circuits and Systems for Video Technology*, 1(2) :197–209, June 1991.
- [5] S.A. Kassam and M. Aburdene. "Multivariate Median Filters and Their Extensions". In *Proc. of IEEE ISCAS Symposium*, pages 85–88, June 1991.
- [6] L. Lucat and P. Siohan "Vector Median-Type Filters and Fast-Computation Algorithms". To appear in *Proc. of IEEE ISCAS Symposium*, June 1997.
- [7] T. Viero, K.O. Östämö, and Y. Neuvo. "Three-Dimensional Median-Related Filters for Color Image Sequence Filtering". *IEEE Tr. on Circuits and Systems for Video Technology*, 4(2) :129–142, April 1994.