

Entrelacement de Bits et “Space Filling Curves”: une Extension des Filtres de Rang aux Images Couleur

Jocelyn Chanussot - Patrick Lambert

Laboratoire d'Automatique et de MicroInformatique Industrielle
LAMII / CESALP - Université de Savoie - BP 806 -74016 Annecy Cedex - e-mail : chanusso@esia.univ-savoie.fr
(CNRS-GdR G1047 - Information Signal Image)

- Résumé -

Nous présentons une technique de codage bijectif de vecteurs sous forme scalaire basée sur l'entrelacement des bits des différentes composantes. Ce codage induit une relation d'ordre total sur les vecteurs et permet donc une extension aisée des filtres de rang aux images multi-composantes. Cette extension peut alors se décomposer en trois étapes: codage, filtrage de l'image codée par un filtre scalaire, puis décodage pour reconstruction de l'image multi-composante filtrée. Nous présentons puis justifions cette technique, en particulier grâce à son interprétation en terme de “space filling curves”; enfin, nous la testons avec des filtres classiques de lissage et de rehaussement sur des images couleur.

1. Introduction

Les filtres de rang constituent un cas particulier de filtres d'ordre (L-filtres), filtres non-linéaires parmi les plus utilisés en traitement d'image [1]. Ils fournissent comme sortie l'une des statistiques d'ordre des données d'entrée. Pour étendre ces filtres aux images multi-composantes (cadre vectoriel), plusieurs approches sont possibles:

- On peut tout d'abord effectuer un traitement dit “marginal” de l'image: chaque composante est traitée de manière scalaire, indépendamment des autres ; ensuite, les résultats de chaque composante sont recombinaés pour obtenir l'image multi-composante filtrée [2]. Cette méthode n'est cependant pas pleinement satisfaisante, pour au moins trois raisons:
 - elle ne prend pas en compte la corrélation existant entre les différentes composantes de l'image ;
 - elle nécessite autant de filtrages que l'image compte de composantes, ce qui peut s'avérer rédhibitoire dans le cas de traitements longs et complexes avec un nombre élevé de composantes ;
 - enfin, attendu que le vecteur de sortie sera rarement l'un des vecteurs d'entrée, l'apparition de fausses couleurs (dans le cas d'images couleur) est inévitable.
- On peut alors tenter de réaliser une extension purement vectorielle du filtre scalaire considéré et pallier ainsi simultanément les trois inconvénients de l'approche marginale. On se trouve alors immédiatement confronté au problème du tri des données: il n'existe en effet pas d'ordre “universel” sur les vecteurs. Certaines méthodes de tri (tri réduit, somme de distances cumulées...) permettent de réaliser de bonnes extensions des

- Abstract -

We present a bijective coding scheme of vectors in a scalar form, based on a bit mixing paradigm. This coding induces a total ordering relation on the vectors and can thus be used to extend rank order filters to multicomponent images. This extension can then be split into three steps: coding, scalar filtering of the coded image, decoding to restore the multicomponent filtered image. We present and justify this approach, especially by interpreting the induced order in term of space filling curves. We then test the method with classical noise removing or edge enhancing filters applied on colour images.

principaux L-filtres [3] [4] [5]. Mais pour un filtre de rang quelconque, il est nécessaire de définir un ordre total sur les vecteurs: en effet, afin de pouvoir définir et utiliser une statistique d'ordre donnée quels que soient les vecteurs d'entrée, il faut pouvoir les trier sans ambiguïté.

La technique dite “par entrelacement de bits” fournit un ordre vectoriel total et permet une extension aisée des filtres de rang au cas multi-composante. Cette extension, illustrée ci-dessous dans le cas d'images couleur tridimensionnelles (figure 1), s'effectue alors en trois temps :

- codage réversible des pixels (vecteurs) sous forme de scalaires par entrelacement de bits,
- filtre de rang scalaire classique sur l'image codée,
- décodage de l'image obtenue pour reconstruction de l'image couleur filtrée.

Dans un premier temps, la technique proposée est détaillée puis l'ordre total induit par le codage est interprété en terme de “space filling curve”, enfin des résultats concernant le filtrage d'images couleur sont présentés.

2. Codage: l'entrelacement de bits

Suivant le formalisme introduit dans [6], la technique proposée ici est basée sur une transformation h qui code chaque pixel (vecteur) de l'image multi-composante sous forme d'un scalaire. On peut alors ordonner n'importe quel groupe de vecteurs en classant leurs images respectives par h .

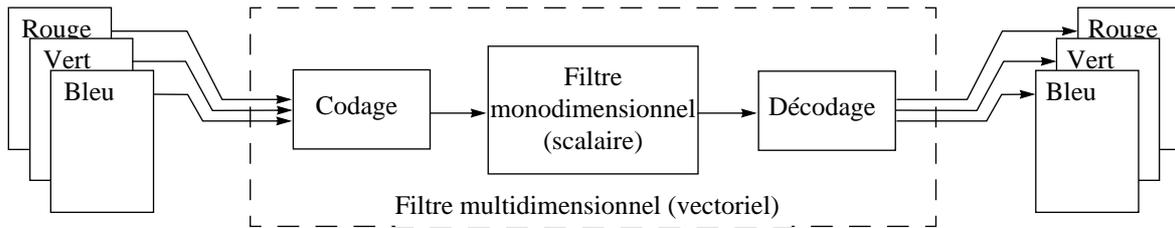


Figure 1: Synopsis de traitement (exemple d'une image couleur).

$$h: \mathfrak{R}^N \rightarrow \mathfrak{R}$$

$$X \rightarrow h(X)$$

La relation d'ordre vectoriel induite se définit alors par:

$$X < Y \Leftrightarrow h(X) < h(Y)$$

La transformation h proposée ici s'appuie sur la représentation binaire de chacune des composantes du vecteur X que l'on considère. Si les N composantes sont codées sur un même nombre p de bits, on dispose alors de $N \cdot p$ bits que l'on mélange afin de construire un scalaire codé sur $N \cdot p$ bits, image de X par h . Il existe bien sûr de nombreuses façons de "mélanger" ces $N \cdot p$ bits, mais en l'absence de connaissance *a priori* concernant la part relative d'information portée par chacune des composantes, on choisit une combinaison la plus symétrique possible: on place donc alternativement le premier bit de $X(1)$, puis le premier bit de $X(2)$... jusqu'à $X(N)$, puis on passe au deuxième bit et ainsi de suite jusqu'au p ^{ième} bit.

Analytiquement, cela peut s'écrire de la manière suivante: X est constitué de N composantes $X(i)$. Chacune de ces composantes est codée sur p bits $X(i)_j$:

$$X(i) = \sum_{j=1}^p 2^{(p-j)} \cdot X(i)_j, \text{ avec pour } j \text{ de } 1 \text{ à } p: X(i)_j \in \{0, 1\}$$

L'application h considérée peut s'écrire:

$$h(X) = \sum_{j=1}^p \left\{ 2^{N \cdot (p-j)} \cdot \sum_{i=1}^N 2^{N-i} \cdot X(i)_j \right\}$$

De même, si on considère un scalaire entier S codé sur $N \cdot p$ bits:

$$S = \sum_{j=1}^{N \cdot p} 2^{N \cdot p - j} \cdot s_j, \text{ avec pour tout } j \text{ de } 1 \text{ à } p: s_j \in \{0, 1\}$$

on peut calculer simplement les N composantes X_i de son antécédent X dans T par h :

$$X(i) = \sum_{j=1}^p 2^{p-j} \cdot s_{N \cdot (p-j) + N - i + 1} \text{ pour } i \in \{1 \dots N\}$$

Sans perte de généralité, pour illustrer ce mécanisme de codage, nous allons maintenant présenter la transformation h

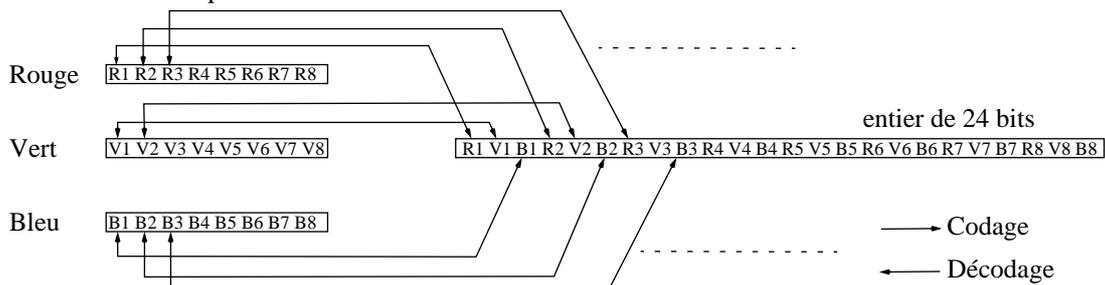


Figure 2: Codage et décodage d'un pixel couleur.

sous forme d'un schéma en nous limitant au cas d'une image couleur dans la base classique Rouge-Vert-Bleu (RVB), chacune des composantes R,V et B étant codée sur 8 bits R_j , V_j et B_j . La figure 2 présente la manière dont les $3 \cdot 8$ bits sont entrelacés pour former le scalaire de 24 bits.

Après codage, on obtient alors une image en "pseudo niveaux de gris" sur laquelle on peut appliquer un traitement scalaire, selon le synopsis de la figure 1. Après filtrage, le décodage par h^{-1} permet de reconstruire l'image couleur filtrée.

L'entrelacement décrit figure 2 est celui qui minimise la dissymétrie entre composantes inhérente à ce type de méthodes. Néanmoins, si l'on dispose d'information *a priori* concernant l'importance relative de l'information portée par chaque composante, on peut modifier très simplement l'entrelacement en conséquence: par exemple en plaçant plusieurs bits de la composante la plus significative en tête. On peut même, par cette méthode, retrouver le cas extrême de l'ordre classique lexicographique: il suffit simplement pour cela de concaténer les bits des différentes composantes, sans les entrelacer du tout.

Cette méthode par entrelacement de bits présente plusieurs avantages:

- Le codage est extrêmement simple à implanter et donne lieu à de faibles temps de calcul;
- pour l'étape de filtrage proprement dite, il suffit d'appliquer un simple filtre scalaire ;
- on peut donner une justification supplémentaire de l'approche proposée en remarquant que l'ordre induit par notre transformation h réalise une extension de l'ordre partiel canonique classique défini par:

$$X < Y \Leftrightarrow X(i) < Y(i) \quad \forall i \in \{1 \dots N\} \quad \text{Eq.1}$$

Deux vecteurs comparables au sens de cette relation vont garder le même ordre avec la relation proposée, mais maintenant tous les vecteurs sont comparables (on a "complété" l'ordre canonique partiel en un ordre total).

3. Ordre vectoriel et “space filling curve”

Pour tout ordre total sur un espace, il est possible de tracer une courbe joignant le plus petit élément de l’espace au plus grand en passant, par ordre croissant, par tous les autres. Cette courbe, qui passe une fois unique par chacun des points, remplit l’espace: c’est littéralement une “space filling curve” [7].

Le mathématicien Peano travailla le premier sur ces courbes (figure 3.d); plus récemment, Plataniotis en utilisa une autre (figure 3.c) pour faire du filtrage médian avec une approche comparable à celle exposée ici [8].

La courbe introduite par notre transformation est représentée figure 3.a. dans le cas de vecteurs à 2 dimensions (la courbe reste plane), chaque composante C_1 et C_2 étant codée sur 3 bits.

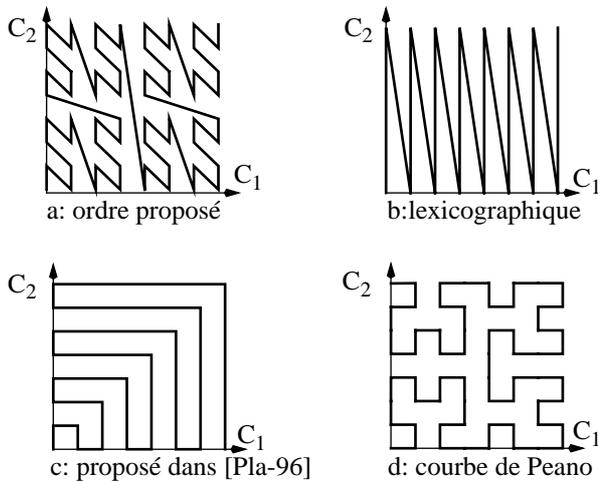


Figure 3: différentes “space filling curves” dans le cas bidimensionnel.

On peut faire quelques observations qui restent vraies dans le cas général (N composantes codées sur p bits):

- la courbe est auto-similaire (structure fractale). Cela revient à une approche multi-échelle de la comparaison des vecteurs: si les bits de poids forts sont différents, ils suffisent à classer les vecteurs, sinon, les bits de poids plus faibles sont comparés (on prend en compte des détails de plus en plus fins de la courbe jusqu’à pouvoir isoler les deux points que l’on souhaite comparer dans deux blocs différents).

- Ces transformations ne respectent pas, à différents degrés, la topologie de l’espace: des points proches dans l’espace peuvent se retrouver éloignés le long de la courbe, ce qui, du point de vue du traitement d’image, peut s’interpréter comme un accroissement du bruit; le phénomène inverse (points éloignés que la courbe rapproche) risque quant à lui de masquer certaines transitions.

Ce problème, lié à la non linéarité de la transformation h utilisée, est inévitable lorsque l’on passe d’un espace à plusieurs dimensions à un espace mono-dimensionnel. En effet, il n’existe pas de transformation qui soit à la fois linéaire et bijective et qui modifie la dimension de l’espace de départ. Puisque l’on a fait le choix de la bijectivité (indispensable pour induire une relation d’ordre total classant sans ambiguïté n’importe quelle famille de vecteurs), il s’agit donc de choisir la transformation qui perturbe le moins possible la topologie. Une topologie étant essentiellement définie par la notion de voisinage qu’elle induit sur les points de l’espace, nous avons donc utilisé un critère qui

mesure la dispersion moyenne le long de la courbe du voisinage d’un vecteur. En nous plaçant en 8-connexité, nous avons donc regardé quel pourcentage des 8 voisins de chaque vecteur se retrouvait parmi les 8 points les plus proches du point considéré, après transformation (c’est à dire, le long de la courbe). Le tableau 1 présente les résultats obtenus avec les différentes courbes de la figure 3. Ce critère place notre courbe juste derrière celle de Peano mais devant les autres. Cependant, il faut noter que l’ordre induit par le codage de Peano contredit souvent l’ordre canonique (Eq.1)(deux vecteurs comparables canoniquement le seront aussi avec Peano, mais parfois dans le sens contraire).

Courbe	Fig 3.a	Fig 3.b	Fig 3.c	Fig 3.d
Taux de Préservation Voisinage	48,6%	22%	39,8%	58%

Tableau 1: Préservation de la topologie avec différentes courbes.

D’une manière générale, à tout ordre total correspond une “space filling curve”, et réciproquement. L’étude de la courbe permet de caractériser l’ordre choisi: par exemple, plus la courbe est convolutive, enroulée sur elle-même, et plus la topologie est respectée. De même, la structure “par couche” de la courbe 3.c indique un ordre essentiellement basé sur la norme euclidienne des vecteurs; la forte anisotropie de 3.b montre le dissymétrie qu’introduit entre les composantes l’ordre lexicographique.

Remarque: la perturbation de la topologie, ou, de manière équivalente, la non linéarité de la transformation liée à l’ordre total considéré et la non conservation du barycentre qu’elle implique, a pour effet principal de réduire le domaine d’application de notre méthode. En effet, dès que l’on considère un filtre d’ordre quelconque, mettant en jeu une combinaison linéaire des statistiques d’ordres non réduite au cas particulier des filtres de rang, la méthode est inexploitable. Dans ce cas, les méthodes classiques [3] ou même les méthodes marginales sont plus indiquées. Par exemple, la figure 4 présente la luminance d’une image couleur (4.a: vue aérienne des îles Canaries) et la luminance de l’image filtrée par un simple moyennneur scalaire de taille 5 avec l’approche proposée: on constate une dégradation importante de l’image.

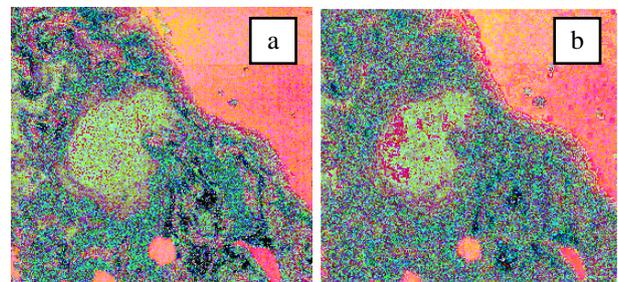


Figure 4: Effet du moyennneur.

4. Applications: lissage et rehaussement

- La méthode proposée a tout d’abord été testée avec le filtre médian sur une image couleur naturelle. La figure 5 présente les résultats obtenus:

- la figure 5.a présente le résultat (les contours) de la segmentation vectorielle par croissance de région de l'image couleur originale . On observe de nombreuses régions parasites.

- la figure 5.b présente le résultat de la même segmentation, mais appliquée sur l'image filtrée par un filtre médian vectoriel classique de taille 3. On constate que l'image a été convenablement lissée par l'étape de filtrage (il y a 25% de régions en moins).

- la figure 5.c présente le résultat de la même segmentation, mais appliquée sur l'image filtrée selon notre approche (codage par entrelacement, filtre médian scalaire de taille 3, décodage). On constate là aussi que le filtrage lisse convenablement l'image (24,5% de régions en moins).

On peut noter que les effets de lissage sont comparables dans les deux cas.

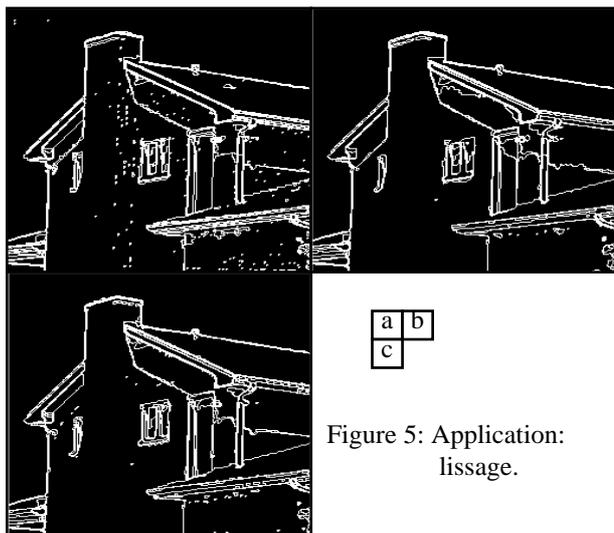


Figure 5: Application: lissage.

● Nous avons également testé la méthode avec un filtre de rang rehausseur. L'image couleur utilisée est celle dont la luminance est présentée sur la figure 4.a (vue aérienne des îles Canaries).

- la figure 6.a présente l'image filtrée par un filtre $d_{\alpha\beta}$ vectoriel de taille 7 avec $\alpha=0.3$ et $\beta=1.4$. On démontre que ce filtre dont la sortie, pour M vecteurs d'entrée X_i , est le vecteur Y minimisant le coût:

$$d_{\alpha\beta}(Y) = \left[\sum_{i=1}^M (\|Y - X_i\|_{\beta})^{\alpha} \right]^{1/\alpha}$$

est un filtre de rang pour $\alpha < 1$; on a alors un filtre dont le rehaussement est contrôlé par α [3].

- la figure 6.b présente l'image filtrée selon notre approche (codage, filtre scalaire, décodage). Le filtre scalaire utilisé est le filtre d_{α} de taille 7 avec $\alpha=0.3$. La sortie de ce filtre [1], pour M valeurs scalaires d'entrée X_i , est le scalaire Y minimisant le coût:

$$d_{\alpha}(Y) = \left[\sum_{i=1}^M |Y - X_i|^{\alpha} \right]^{1/\alpha}$$

On montre que pour $\alpha < 1$, on a un filtre de rang avec un effet de rehaussement contrôlé par α .

Les pouvoirs de rehaussements obtenus dans les deux cas sont également comparables, mais avec des temps de calculs bien moindre avec notre approche.

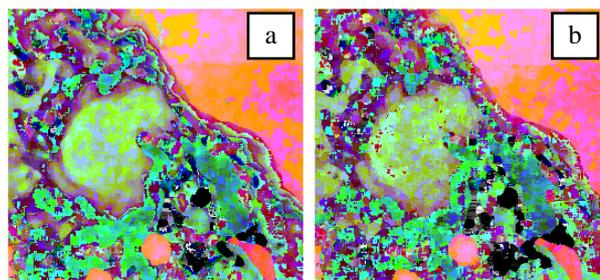


Figure 6: Application: rehaussement.

5. Conclusion

Les résultats présentés valident notre méthode: les extensions aux images multi-composantes des filtres classiques par codage + filtrage scalaire + décodage sont comparables avec ceux des extensions classiques. Par contre, la mise en œuvre (implantation et temps de calculs) est comparable à celle d'un filtre scalaire et donc largement moins lourde que celle des filtres vectoriels classiques.

Par contre, le domaine d'application de notre approche est strictement limité aux filtres de rangs. Cela ouvre néanmoins également des horizons pour les filtres morphologiques qui sont essentiellement basés sur les notions de borne supérieure et inférieure [9].

Références

- [1] Ph. Bolon, Filtrage d'Ordre: Vraisemblance et Optimalité des Prétraitements d'Image, Traitement du Signal, vol 9, n. 3, 1992, pp 225-250.
- [2] I. Pitas, Marginal Order Statistics in Color Image Processing, Signal processing V - Theories & Applications, Torres Editor, Elsevier, 1990, pp 809-812.
- [3] W. Alshatti, Approches vectorielles du Filtrage et de la Détection de Contours dans des Images Multi-Spectrales, thèse de l'Université de Savoie soutenue le 12 Octobre 1994.
- [4] J. Astola, P. Haavisto & Y. Neuvo, Vector Median Filtering, Proc. of the IEEE, vol 78, n. 4, Avril 1990, pp 678-689.
- [5] S. Sanwalka & A. Venetsanopoulos, Vector Order Statistics Filtering of Colour Images, 13eme colloque GRETSI, Juan-les-Pins Septembre 1991, pp 785-788.
- [6] J. Goutsias, H. Heijmans & K. Sivakumar, Morphological Operators for Image Sequences, Computer Vision and Image Understanding, vol 62, n. 3, 1995, pp 326-346
- [7] R. Stevens, A. Lehar & F. Preston, Manipulation and Presentation of Multidimensional Image Data Using the Peano Scan, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol PAMI-5, n.5, septembre 1983, pp 520-526.
- [8] K. Plataniotis, C. Regazzoni, A. Teschioni & A. Venetsanopoulos, A New Distance Measure for Vectorial Rank Order Filters based on Space Filling Curves, Proc. IEEE International Conference on Image Processing (ICIP'96), vol 1, pp 411-414, Lausanne septembre 1996.
- [9] J. Chanussot & P. Lambert, Bit Mixing Paradigm for Multivalued Morphological Filters, Proc. IEE International Conference on Image Processing and its Applications (IPA'97), Dublin juillet 1997.