



HAL
open science

Ordre de couleurs : une approche par graphe

O Lezoray, C Meurie, A Elmoataz

► **To cite this version:**

O Lezoray, C Meurie, A Elmoataz. Ordre de couleurs : une approche par graphe. Compression et REprésentation des Signaux Audiovisuels - CORESA, Nov 2006, Caen, France. pp.64-69. <hal-04760862>

HAL Id: hal-04760862

<https://hal.science/hal-04760862v1>

Submitted on 30 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Ordre de couleurs : une approche par graphe

O. Lezoray

C. Meurie

A. Elmoataz

LUSAC EA 2607, IUT SRC, 120 Rue de l'exode, 50000 SAINT-LÔ, FRANCE

olivier.lezoray@unicaen.fr, cmeurie@info.unicaen.fr, abder.elmoataz@greyc.ensicaen.fr

Résumé

Cet article présente une nouvelle approche d'ordre de données vectorielles. Nous nous intéressons ici plus particulièrement au cas des images couleur. L'ordre que nous proposons permet de pallier les défauts des ordres vectoriels classiques. Celui-ci est construit sur un voisinage de pixels et non défini a priori. L'approche que nous proposons est basée d'une part sur l'extraction des infimum et supremum d'un ensemble de couleurs puis d'autre part sur la construction de l'ordre à partir de l'infimum. L'ordre ainsi construit définit un chemin hamiltonien sur le graphe non orienté totalement connecté représentant l'élément structurant.

Mots clefs

Ordre vectoriel, graphe, couleur, filtrage.

1 Introduction

En traitement d'images, dès lors que l'on s'intéresse à des images multivariées, se pose le problème de l'extension des algorithmes usuels pour le traitement des images en niveaux de gris. Ceci n'est pas un problème récent et de nombreux auteurs se sont penchés sur le cas des images couleur qui représente le cas le plus fréquent d'images multivariées [1]. La majorité des premiers travaux a concerné l'extension du filtre médian aux images couleur [2]. Dans le cas des images en niveaux de gris, à chaque pixel est associé une valeur dans \mathbb{R} et la relation $<$ étant un ordre total sur \mathbb{R} il est facile d'ordonner les valeurs d'un ensemble de pixels suivant cette relation d'ordre. Puisque chaque composante d'une image couleur peut être considérée comme une image en niveaux de gris, un filtre médian peut être appliqué sur chaque composante séparément. Ce type d'approche est appelé traitement marginal. Le traitement marginal n'exploite pas la corrélation existante entre les différentes composantes (particulièrement importante pour le cas des images couleur) et peut provoquer l'apparition de fausses couleurs. Un traitement vectoriel est donc plus approprié afin de prendre en compte la nature vectorielle des données. Cependant, cela nécessite la définition de nouveaux modèles pour l'extension des algorithmes scalaires au cas vectoriel. Le principal problème de ce genre d'extension repose sur la définition d'un ordre vectoriel approprié car il n'existe pas d'ordre naturel sur les vecteurs. Dans cet article nous proposons une approche de construc-

tion d'un ordre vectoriel. Nous utilisons ici le terme de construction car l'ordre que nous proposons n'est pas défini a priori mais dépend des données à ordonner. Dans la suite de cet article nous rappelons la notion d'ordre et les différents ordres vectoriels de la littérature. Nous présentons ensuite le lien qu'il existe entre un ordre vectoriel et la notion de filtrage. A partir des constatations effectuées, nous proposons une approche par graphe qui permet d'ordonner un ensemble de couleurs selon une relation d'ordre découverte à partir des données. Cette relation d'ordre est obtenue par décimation de l'arbre de recouvrement minimum selon le degré des noeuds du graphe. Nous présentons quelques résultats de comparaison de notre ordre vectoriel avec les ordres usuels.

2 Ordre vectoriel

Les trois ordres vectoriels importants sont le pré-ordre, l'ordre partiel et l'ordre total. Rappelons tout d'abord les définitions utiles à la caractérisation d'une relation d'ordre. Soit R une relation binaire sur un ensemble quelconque A . R est réflexive ssi $\forall x \in A, xRx$. R est transitive ssi $\forall x, y, z \in A, xRy \text{ et } yRz \Rightarrow xRz$. R est anti-symétrique ssi $\forall x, y \in A, xRy \text{ et } yRx \Rightarrow x = y$. Une relation binaire R sur un ensemble A est un pré-ordre ssi R est réflexive et transitive. Par exemple, pour les relations d'ordre sur \mathbb{R} , la relation binaire R est en général la relation $<$ ou \leq . Une relation binaire R sur un ensemble A est un ordre partiel ssi R est réflexive, transitive et anti-symétrique. Un ordre partiel est totalement ordonné ssi $\forall x, y \in A, xRy$ ou yRx . Cette dernière définition caractérise un ordre pour lequel il n'existe pas de paire de membres de l'ensemble A non ordonnés. On appelle un ordre partiel qui est totalement ordonné un ordre total. Selon Barnett [3], on peut distinguer plusieurs méthodes pour ordonner des données vectorielles : l'ordre marginal, l'ordre réduit, l'ordre partiel et l'ordre conditionnel ou lexicographique. Nous rappelons leurs principes. Soit x_1, x_2, \dots, x_n un ensemble de n vecteurs de dimension p , avec $x_i = \{x_i^1, x_i^2, \dots, x_i^p\}, x_i \in \mathbb{R}^p$.

- L'ordre marginal : les composantes vectorielles sont ordonnées selon chaque dimension p indépendamment. Pour deux vecteurs x_i et x_j , on a $x_i \leq x_j \Leftrightarrow x_i^k \leq x_j^k, \forall k \in \{1, 2, \dots, p\}$. Cet ordre est un ordre partiel.
- L'ordre réduit : chaque vecteur est réduit à un scalaire et les données vectorielles sont triées selon l'ordre des valeurs scalaires obtenues. Avec une fonction scalaire

$d : \mathbb{R}^p \rightarrow \mathbb{R}$ utilisée pour trier les vecteurs, on obtient pour deux vecteurs x_i et $x_j : x_i \leq x_j \Leftrightarrow d(x_i) \leq d(x_j)$. Cette approche est très utilisée notamment dans les filtres médians vectoriels couleur où la fonction scalaire est une fonction de distance entre les pixels couleurs.

- L'ordre partiel : les vecteurs sont regroupés en sous-ensembles qui sont ensuite ordonnés, cet ordre est fondé sur la structure géométrique des vecteurs et notamment sur leur enveloppe convexe.
- L'ordre conditionnel ou lexicographique est basé sur la relation d'ordre suivante. Pour deux vecteurs x_i et x_j :

$$x_i \leq x_j \begin{cases} x_i^1 < x_j^1, \text{ ou} \\ x_i^1 = x_j^1, \text{ et, } x_i^2 < x_j^2 \text{ ou } \dots \\ x_i^1 = x_j^1, \text{ et, } x_i^2 = x_j^2 \dots x_i^p < x_j^p \end{cases}$$

Cet ordre est un ordre vectoriel total mais il introduit une forte dissymétrie entre les composantes.

On trouve également d'autres ordres dans la littérature, nous ne les détaillons pas ici et nous ne retenons que l'ordre par entrelacement de bits pour ses propriétés intéressantes. Celui-ci, proposé par Chanussot [4], est un ordre total et constitue un apport important dans la littérature concernant les ordres vectoriels. Il s'appuie sur une représentation binaire des composantes codées sur 8 bits en RVB et construit un scalaire codé sur 24 bits en entrelaçant de façon symétrique chacun des bits des composantes.

3 Ordres et filtres

Un des champs d'application très important du traitement d'images multivariées et plus précisément d'images couleur est le filtrage. Il est bien établi à présent que le traitement d'images couleur doit se faire en tenant compte de la nature vectorielle des données étant donné la corrélation qui existe entre des composantes couleur. Nous insistons, dans cette section, sur le lien entre ordre vectoriel de couleur et deux familles de filtres que sont les filtres morphologiques et les filtres médians. Chacune de ces familles est basée sur un ordre de couleurs spécifique qui présente certains avantages ou inconvénients que nous présentons en détail.

3.1 Filtres morphologiques vectoriels

La morphologie mathématique est une approche du traitement d'images qui est basée sur une structure fondamentale, le treillis complet, \mathcal{L} [5] tel qu'une relation d'ordre \leq est définie sur \mathcal{L} et tel que pour chaque sous ensemble fini \mathcal{K} de \mathcal{L} , existent un supremum $\vee \mathcal{K}$ et un infimum $\wedge \mathcal{K}$. Pour pouvoir appliquer la morphologie mathématique aux images couleur et construire un treillis complet, il est nécessaire de pouvoir ordonner les couleurs et de vérifier l'existence des supremum et infimum. Pour certains ordres vectoriels, le supremum et l'infimum d'un ensemble de vecteurs ne font pas toujours partie de cet ensemble. Dans le cas spécifique des images couleur, ce problème se manifeste par l'introduction de fausses couleurs par les opé-

rateurs morphologiques. Une contrainte supplémentaire à la définition de filtres morphologiques est donc que le supremum (\vee) et l'infimum (\wedge) d'un ensemble fassent partie de celui-ci. Ceci n'est pas vrai pour tous les ordres et l'ordre marginal introduit par exemple de fausses couleurs. C'est pourquoi les ordres habituellement utilisés en morphologie mathématique sont des ordres totaux. Dans ce cadre, les principaux ordres de couleur sont l'ordre lexicographique [6, 7] et l'ordre par entrelacement de bits [4] qui permettent d'imposer que le supremum et l'infimum appartiennent au treillis \mathcal{L} . L'ordre lexicographique est fortement dissymétrique et quand un ordre lexicographique est utilisé avec des opérateurs morphologiques dans un espace couleur quelconque, on trouve que la plupart des décisions sur l'ordre des vecteurs dans un élément structurant sont prises au premier niveau de la relation d'ordre. Ceci donne des opérateurs qui ne sont pas homogènes dans leur traitement de l'espace. Ceci est un inconvénient majeur pallié par l'utilisation d'espaces de type luminance/teinte/saturation [7, 8]. L'ordre par entrelacement de bits permet quand à lui de limiter la dissymétrie entre les composantes, mais étant donné qu'il est basé sur un entrelacements de bits, il n'est utilisable que pour des espaces où les composantes sont codées en entiers. Ceci n'est pas le cas de beaucoup d'espaces couleur. Cet ordre est donc principalement conçu pour l'espace RVB. Une fois l'ordre de couleurs choisi, on peut appliquer les deux principales opérations de morphologie mathématique à savoir l'érosion ϵ et la dilatation δ . On peut ensuite obtenir un certain nombre d'opérations morphologiques par composition de ces deux opérations élémentaires.

3.2 Filtres médians vectoriels

Le filtre vectoriel le plus populaire est le filtre médian vectoriel (VMF) [2]. Le VMF est un opérateur vectoriel qui a été introduit comme une extension du filtre médian scalaire. Ce filtre est basé sur un ordre de vecteurs sur un élément structurant donné. La sortie de ce type de filtre est définie comme le vecteur de plus faible rang selon un ordre réduit [9] basé sur les distances entre couleurs. Nous en détaillons le principe. Sur un élément structurant donné, on obtient un ensemble W de n vecteurs x_1, x_2, \dots, x_n de dimension p , avec $x_i = \{x_i^1, x_i^2, \dots, x_i^p\}, x_i \in \mathbb{R}^p$. En général, la différence entre deux vecteurs x_i et x_j peut être mesurée par une distance de Minkowski $\|x_i - x_j\|_\gamma = \left(\sum_{k=1}^p \|x_{k(i)} - x_{k(j)}\|^\gamma \right)^{1/\gamma}$ où γ désigne la norme employée qui est habituellement une norme euclidienne ($\gamma = 2$). A partir d'un ensemble de vecteurs, le filtre VMF définit un ordre réduit où chaque vecteur est réduit à une valeur scalaire définie par $d_i = \sum_{k=1}^n \|x_i - x_k\|$. Ceci revient à calculer pour chaque vecteur de l'ensemble la somme des distances à tous les autres vecteurs de l'ensemble. Les valeurs d_i des vecteurs x_i sont ensuite triées en ordre ascendant et cet ordre des distances ($d_{(1)} \leq d_{(2)} \leq \dots \leq$

$d_{(n)}$) est utilisé pour trier les vecteurs de l'ensemble initial ($x^{(1)} \leq x^{(2)} \leq \dots \leq x^{(n)}$). La sortie d'un filtre VMF est donc le vecteur qui minimise la distance aux vecteurs de l'ensemble constituant l'élément structurant. La différence d'orientation entre deux vecteurs peut également être utilisée comme mesure de distance. Ce type de distance est à la base des filtres médians vectoriels directionnels (VDF) [10]. Le VDF est un filtre dont la sortie est le vecteur qui minimise la somme des orientations avec les autres vecteurs de l'élément structurant. Ce type de filtre ne prenant en compte que l'information chromatique, un autre filtre d'ordre, appelé filtre directionnel de distance (DDF) combinant les deux critères [11], a été proposé. On trouve énormément d'autres améliorations de ce type de filtre médian vectoriel, ces améliorations portant essentiellement sur une pondération des distances (voir dans [1, 2] pour une revue complète). Bien que très utilisé, ce type d'ordre est un ordre réduit et ne fournit donc pas d'ordre total entre les vecteurs, ce qui ne permet pas son utilisation pour des filtres morphologiques, même si ils sont très utiles pour supprimer du bruit [1].

3.3 Constatations

La première constatation est que les ordres réduits utilisés pour les filtres médians ne sont pas adaptés au filtrage morphologique puisqu'ils ne définissent pas un treillis complet. Ils sont cependant adaptés à n'importe quel type d'image multivariée et à n'importe quel espace couleur. La seconde constatation est que les ordres utilisés pour le filtrage morphologique sont majoritairement des ordres totaux mais qui présentent le principal défaut de n'être adaptés qu'au traitement dans un espace couleur donné : *RVB* pour l'ordre par entrelacement de bits et *HSI* pour l'ordre lexicographique. De plus, ces ordres ne peuvent pas être utilisés sur des images multivariées : pour l'ordre par entrelacement, nous sommes limités au nombre de bits maximum pouvant coder un entier et pour l'ordre lexicographique, contrairement à ce qui est dit dans [12], il n'est absolument pas envisageable de l'utiliser pour des données de grande dimension car déterminer l'ordre optimal parmi toutes les permutations possibles d'un ordre lexicographique est un problème NP-complet [13]. Tous les ordres usuels ayant des désavantages, nous cherchons à en concevoir un qui fonctionne sur des vecteurs de dimensions quelconques.

4 Ordre par graphe

Dans cette section, nous proposons donc une alternative aux approches classiques pour ordonner des couleurs. Trouver un ordre de vecteurs, sur un voisinage de pixels constitué par un élément structurant, revient à trouver un chemin hamiltonien dans un graphe non-orienté. Ce graphe a pour noeuds chacun des pixels et pour arêtes toutes les associations deux à deux des noeuds car on cherche à ordonner des pixels non forcément connexes (le graphe est complet). Trouver un chemin hamiltonien (un chemin passant par tous les sommets du graphe en ne passant qu'une

fois par chaque noeud) est un problème NP-complet, nous proposons donc ici une méthode heuristique pour trouver un tel chemin. Nous rappelons tout d'abord ce que l'on veut obtenir. A partir d'un ensemble de pixels décrits par un vecteur de dimension p , on définit un ensemble non ordonné de vecteurs x_1, x_2, \dots, x_n . On désire définir un ordre sur ces vecteurs, c'est-à-dire obtenir un ensemble ordonné $x^{(1)} \leq x^{(2)} \leq \dots \leq x^{(n)}$. $x^{(1)}$ et $x^{(n)}$ doivent représenter respectivement l'infimum et le supremum de cet ensemble et $x^{(n+1)/2}$ le médian de cet ensemble. Ceci correspond exactement à la définition d'un ordre scalaire. Nous nous restreignons au cas de la couleur ($p = 3$) mais le principe reste identique pour des dimensions plus élevées.

4.1 Extraction de l' \wedge et du \vee

L'idée principale de cette nouvelle approche est la suivante : un ordre total définit un chemin hamiltonien le long de tous les pixels de l'élément structurant. Considérons un élément structurant B et un graphe complet G_0 où chaque pixel de B est un noeud du graphe et où les noeuds sont tous liés deux à deux (connexité totale). Pour obtenir un ordre total de tous les vecteurs couleurs, nous avons besoin de simplifier le graphe G_0 afin d'obtenir un chemin partant de la borne inférieure \wedge et arrivant à la borne supérieure \vee de l'ensemble couleur. Ce chemin doit posséder deux propriétés. Il doit passer par tous les noeuds mais en ne les traversant qu'une seule et unique fois. Il doit aussi définir un ordre des vecteurs couleurs comme étant un treillis complet (un ordre total et la définition du \vee et de l' \wedge). Il est difficile (problème NP-complet) de trouver un tel chemin parmi toutes les possibilités présentes dans le graphe G_0 . C'est pourquoi, nous proposons de faire une approximation de ce chemin [14] en calculant l'arbre de recouvrement minimal [15] (« Minimum Spanning Tree : MST ») du graphe complet G_0 et où toutes les arêtes reliant deux noeuds sont valuées par une norme L_2 entre les vecteurs décrivant chaque noeud. L'arbre de recouvrement minimal n'est pas très loin de la solution que nous attendons [16]. La figure 1 illustre ce point, le MST du graphe est calculé et les noeuds peuvent être classés comme noeuds internes et externes (feuilles) selon leur degré. On rappelle que le degré d'un noeud est son nombre de voisins. Un chemin est hamiltonien si deux noeuds sont de degré 1 et que tous les autres sont de degré 2. Donc, si le MST est un ordre des vecteurs couleurs, il définit un tel chemin parmi les noeuds : ce chemin possède seulement deux noeuds externes et chaque noeud interne est connecté à exactement deux noeuds. C'est rarement le cas, pourtant nous pouvons faire une supposition sur les propriétés des noeuds du MST puisque c'est une généralisation à des dimensions plus élevées d'une liste à une dimension triée [16]. Si un noeud du MST est un noeud interne, il ne peut pas être le \vee ou l' \wedge de l'ensemble des couleurs, car il aurait été considéré dans ce cas comme un noeud externe du chemin d'ordre des vecteurs couleurs. Pour trouver le \vee et l' \wedge de l'ensemble des vecteurs couleurs, nous pouvons utiliser

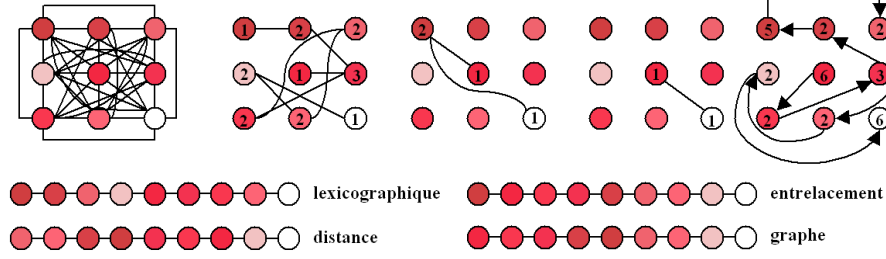


Figure 1 – Ligne du haut : processus de décimation du MST du graphe (de gauche à droite, le graphe G_0 , les MST successifs $MST(G_\lambda)$ avec $\lambda \in [0 - 2]$) et l'ordre de vecteurs construit. Lignes du bas : comparaison entre différents ordres.

cette dernière propriété en construisant un nouveau graphe complet G_1 constitué uniquement des couleurs possibles : les noeuds externes du MST de G_0 . Chaque noeud de G_1 est lié à tous ses autres noeuds candidats (les externes de $MST(G_0)$) et le MST de G_1 est calculé. Ce procédé peut être réitéré jusqu'à ce que le MST obtenu à l'itération i soit réduit à deux noeuds définissant respectivement le \vee et l' \wedge . Cependant une dernière étape est à effectuer à savoir, définir quel noeud est le \vee et par élimination l' \wedge . L' \wedge est identifié comme étant le noeud se rapprochant le plus d'une couleur de référence C_{ref} [6] (généralement le noir). L'algorithme complet [14] est résumé dans l'algorithme 1 présenté ci-après.

```

end ← faux
λ ← 0
Construire  $G_\lambda$  sur l'élément structurant  $B$ 
Répéter
  Calculer  $MST(G_\lambda)$ 
   $n = \text{NombreNoeudsExternes}(MST(G_\lambda))$ 
  Si ( $n=2$ ) Alors
    | end ← vrai ;
  Sinon
    |  $\lambda \leftarrow \lambda + 1$ 
    |  $G_\lambda = \text{NoeudsExternes}(MST(G_{\lambda-1}))$ 
    | Lier tous les noeuds de  $G_\lambda$ 
  Fin Si
jusqu'à ce que (end= vrai)
 $G_\lambda = (\{V_1, V_2\}, \{E_1\})$ 
 $\vee = \text{argmax } d(V_i, C_{ref})$  et  $\wedge = \text{argmin } d(V_i, C_{ref})$ 

```

Algorithme 1 – Détermination du \vee et de l' \wedge d'un ensemble de vecteurs couleurs.

La figure 1 illustre la détermination du \vee et de l' \wedge : la première étape est la construction du graphe G_0 , les autres étapes quant à elles présentent les $MST(G_\lambda)$ pour $\lambda \in [0 - 2]$ jusqu'à ce que le critère d'arrêt soit atteint. Sur chaque noeud candidat, à chaque niveau λ , est précisé son degré. Dans cet exemple, la couleur de référence est le noir,

le \vee est donc représenté par le pixel inférieur droit et l' \wedge par le pixel central. Une fois ces deux limites déterminées, nous pouvons effectuer les opérations élémentaires de morphologie mathématique : l'érosion et la dilatation sur un voisinage de pixels considéré.

4.2 Construction de l'ordre

La méthode que nous venons de présenter ne construit pas un ordre des vecteurs couleurs d'un élément structurant mais permet d'en extraire ses deux vecteurs extrêmes. A partir de ces deux vecteurs extrêmes, nous disposons des deux bornes de l'ordre $\wedge = x^{(1)}$ et $\vee = x^{(n)}$. Il nous reste à présent à construire un ordre. Pour cela nous utilisons la cascade de MST générée lors de l'algorithme 1. Pour chaque niveau de décimation λ , nous pouvons associer à chaque noeud x_i son degré $deg_\lambda(x_i)$. Nous pouvons quantifier l'importance globale de chaque noeud en effectuant une somme de ces degrés : le degré global de chaque noeud est défini par $deg(x_i) = \sum_\lambda \lambda deg_\lambda(x_i)$. Pour effectuer un

parcours du graphe initial G_0 en partant de l' \wedge jusqu'au \vee , il suffit de sélectionner comme noeud suivant du chemin celui qui a le degré global minimum en prenant soin de ne passer qu'une fois par chaque noeud. Comme certains noeuds peuvent avoir le même degré global, nous pondérons celui-ci par la valuation de l'arête reliant les noeuds afin de sélectionner le noeud le plus proche en tenant également compte de la distance. En procédant ainsi on attendra naturellement le \vee en dernier car son degré global est égal à celui de l' \wedge et est supérieur à tous les autres. L'algorithme 2 présente la construction du chemin hamiltonien constituant l'ordre. $u \sim v$ signifie que le noeud u est un voisin de v et que u n'appartient pas déjà au chemin construit (on ne passe qu'une fois par chaque noeud). $w_{uv} = \|u - v\|$ est la valuation de l'arête entre deux noeuds u et v . Le chemin hamiltonien définissant l'ordre des pixels des couleurs de la figure 1 est précisé à droite de celle-ci, les degrés globaux des noeuds étant superposés sur ceux-ci. Afin d'illustrer les différences entre les principaux ordres de la littérature, la figure 1 présente une comparaison des ordres obtenus respectivement par un ordre lexicographique, un ordre par entrelacement de bits, un ordre réduit basé sur les distances et notre approche par graphe. La comparaison est effectuée

sur les couleurs de l'élément structurant de la figure 1. Une analyse visuelle nous montre que deux ordres sont moins efficaces (lexicographique et par mesure de distance) et que les ordres par entrelacement de bits et par notre approche par graphe sont meilleurs et très proches (une seule différence ici).

```

i ← 1
x(i) ← ∧
Répéter
| x(i+1) ← arg minxk~x(i) (wxkx(i) deg(xk))
| i ← i + 1
jusqu'à ce que (i = n)

```

Algorithme 2 – Construction du chemin reliant l'∧ et le ∨ dans un ensemble de vecteurs couleurs.

5 Résultats et conclusion

La figure 2 présente une comparaison entre les ordres lexicographiques, par entrelacement de bits et avec notre approche par graphe (figures 2(b)-(g)) sur des opérations d'érosion et de dilatation en *RVB*. Les figures 2(h)-(k) présentent les différences entre les résultats obtenus. On peut constater tout d'abord qu'il est très difficile visuellement de quantifier la différence entre ces ordres, même si les images de différence nous montrent que notre ordre et l'ordre lexicographique sont proches. La figure 3 présente le même genre de résultat mais en comparant, sur une image bruitée, des filtres VMF et VDF utilisant un ordre réduit basé sur les distances avec un ordre basé sur notre approche par graphe. Dans le cas de notre approche, il suffit de changer la pondération des arêtes (distance ou angle) pour obtenir un filtre VMF ou VDF. De même que précédemment, les filtres VMF ou VDF par ordre de distance et notre ordre produisent des résultats très similaires. Cependant, ce qu'il faut remarquer, c'est que notre approche fait aussi bien que les ordres reconnus de la littérature tout en dépassant leurs principales limitations. En effet, notre ordre peut s'appliquer à des images codées dans n'importe quel espace couleur, à des images multispectrales contenant un nombre arbitraire même élevé de composantes et il permet d'effectuer, sous un même formalisme de base, des filtres morphologiques ou médians. En conséquence cet ordre peut s'utiliser facilement pour des filtres morphologiques plus complexes tels que les nivellements, des filtres médians pondérés, mais d'autres applications telles que le tri de couleurs dans un histogramme avant leur mise en correspondance. Ceci sera l'objet de nos futurs travaux ainsi que des comparaisons plus complètes entre les différents ordres.

Références

- [1] K. Plataniotis et A.N. Venetsanopoulos. *Color Image Processing and Applications*. Springer Verlag, 2000.
- [2] R. Lukac, B. Smolka, K. Martin, K.N. Plataniotis, et A.N. Venetsanopoulos. Vector filtering for color imaging. *IEEE Signal Processing Magazine, Special Issue on Color Image Processing*, 22(1) :74–86, 2005.
- [3] V. Barnett. The ordering of multivariate data. *Journal of the royal society of statistics*, A 139(3) :318–355, 1976.
- [4] J. Chanussot et P. Lambert. Total ordering based on space filling curves for multivalued morphology. Dans *Proc. of ISMM*, pages 51–58, 1998.
- [5] C. Ronse. Why mathematical morphology needs complete lattices. *Signal Processing*, 21(2) :129–154, 1990.
- [6] A. Hanbury et J. Serra. Mathematical morphology in the HLS colour space. Dans *Proc. of BMVC'2001*, volume 2, pages 451–460, 2001.
- [7] J. Angulo. Unified morphological color processing framework in a lum/sat/hue representation. Dans *Proc. of ISMM*, pages 387–396, 2005.
- [8] G. Louverdis, M.I. Vardavoulia, I. Andreadis, et Ph. Tsalides. A new approach to morphological color image processing. *Pattern recognition*, 35(8) :1733–1741, 2002.
- [9] J. Astola, P. Haavisto, et Y. Neuvo. Vector median filters. *Proceedings of the IEEE*, 74(4) :678–689, April 1990.
- [10] P.E. Trahanias, D. Karakos, et A.N. Venetsanopoulos. Directional processing of color images : Theory and experimental results. *IEEE Transactions on Image Processing*, 5 :868–880, 1996.
- [11] D.G Karakos et P.E. Trahanias. Combining vector median and vector directional filters : the directional-distance filters. Dans *Proceedings of the International Conference on Image Processing*, pages 171–174, 1995.
- [12] J. Angulo. Morphological color processing based on distances. Application to color denoising and enhancement by centre and contrast operators. Dans *Proc. of VIIP*, pages 314–319, 2005.
- [13] M. Schmitt et L. Martignon. On the complexity of learning lexicographic strategies. *JMLR*, 7 :55–83, 2006.
- [14] O. Lezoray, C. Meurie, et A. Elmoataz. A graph approach to color mathematical morphology. Dans *Proc. of ISSPIT*, pages 856–861, 2005.
- [15] R. Diestel. *Graph Theory*, volume 173. Springer-Verlag, 2005.
- [16] Ch. Theoharatos, G. Economou, et S. Fotopoulos. Color edge detection using the minimal spanning tree. *Pattern recognition*, 38 :603–606, 2005.

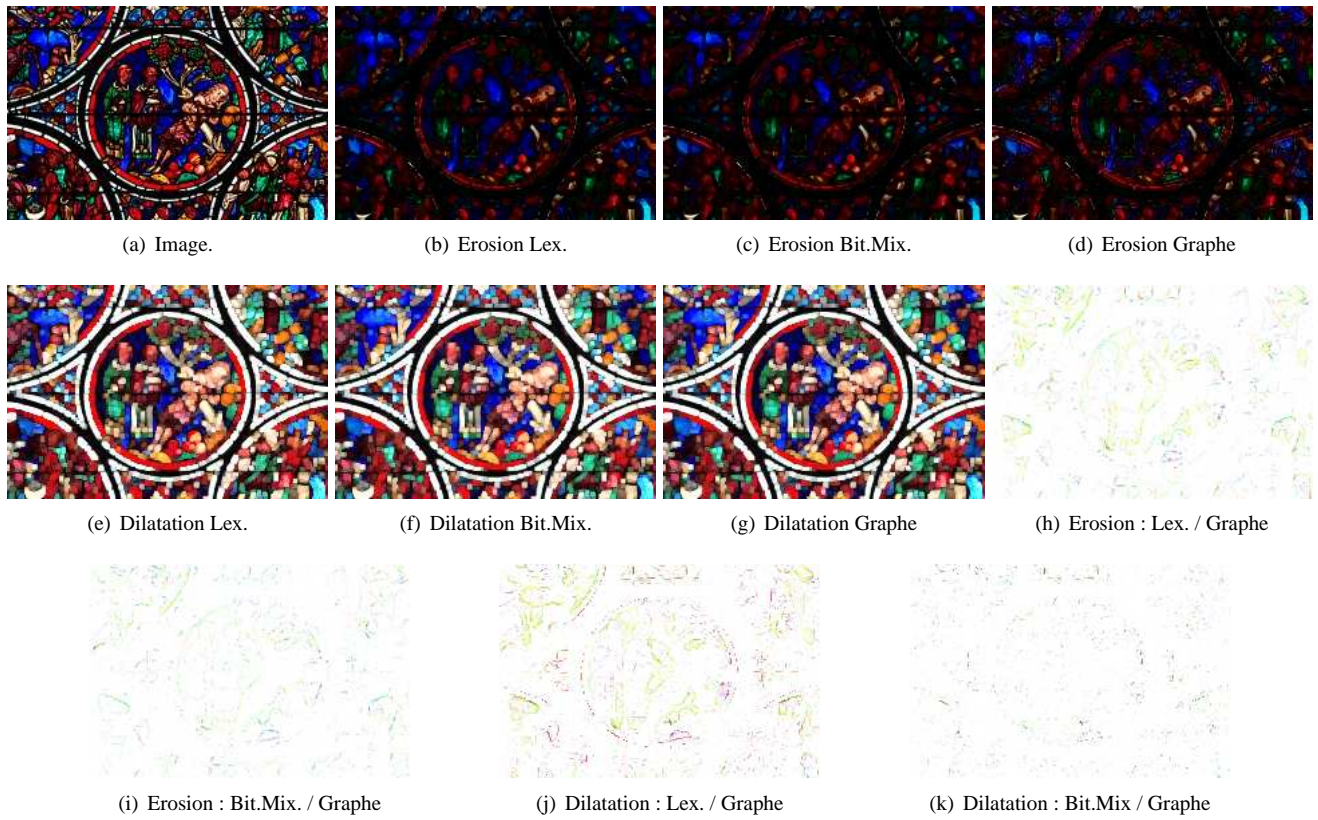


Figure 2 – Comparaison entre ordres pour la morphologie mathématique

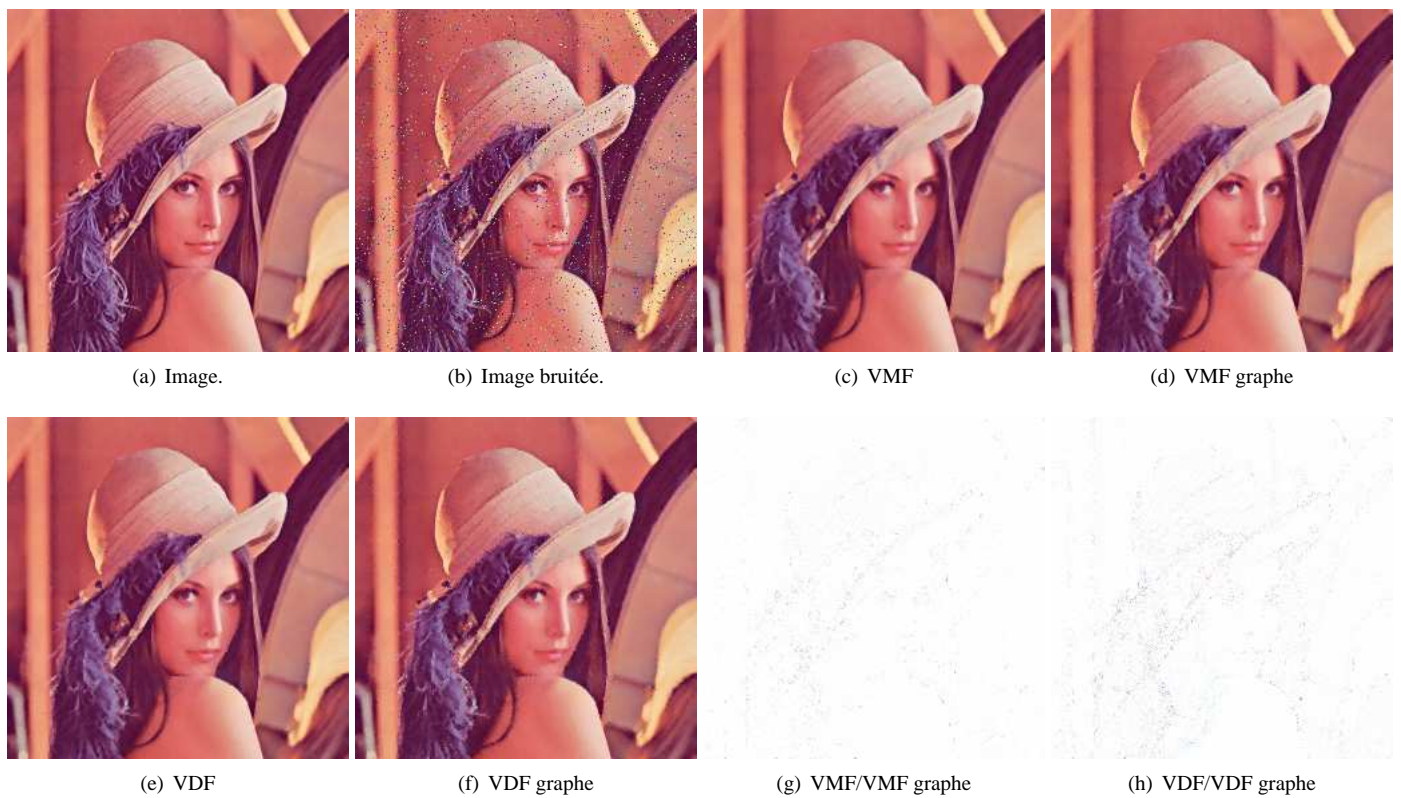


Figure 3 – Comparaison entre ordres pour des filtres médians.