

# Filtres connexes multivariés par fusion d’arbres de composantes

Edwin CARLINET, Thierry GÉRAUD

EPITA Research and Development Laboratory (LRDE)  
14-16 rue Voltaire, F-94270 Le Kremlin-Bicêtre, France  
edwin.carlinet@lrde.epita.fr

## Résumé –

Les arbres de composantes fournissent une représentation d’images de haut niveau, hiérarchisée et invariante par contraste, adaptée à de nombreuses tâches de traitement d’image. Pourtant, ils sont mal définis sur des données multivariées, telle que celles des images couleur, des images multimodalités, des images multibande, etc. Les solutions courantes, telles que le traitement marginal, ou l’imposition d’un ordre total sur les données, ne sont pas satisfaisantes et génèrent de nombreux problèmes, tels que des artefacts visuels, la perte d’invariances, etc. Dans cet article, inspiré par la manière dont l’arbre des formes multivariés (MToS) a été défini, nous proposons une définition pour un Min-Tree ou un Max-Tree multivarié. Nous n’imposons pas un ordre total arbitraire aux valeurs; nous utilisons uniquement la relation d’inclusion entre les composantes. En conséquence, nous introduisons une nouvelle classe d’ouvertures et de fermetures connectées multivariées.

## Abstract –

The component trees provide morphological representations of the inclusion of the connected components of images. Those trees are well-adapted for many image processing tasks. Yet they are ill-formed on multivariate images. Common workarounds such as marginal processing, or imposing a total order on data are not satisfactory and yield many problems, such as artifacts, loss of invariances, etc. In this paper, inspired by the way the Multivariate Tree of Shapes (MToS) has been defined, we propose a definition for Multivariate Min- or Max-trees which does not require an arbitrary total ordering on values; we only use the inclusion relationship between components. It follows the definition of a new class of multivariate connected openings and closings.

## 1 Introduction

La Morphologie Mathématique (MM) est une véritable boîte à outils pour concevoir des filtres de traitement d’image qui servent de blocs de bases permettant à l’utilisateur de créer rapidement ses applications. Les opérateurs basés sur la MM présentent de nombreux avantages du point de vue du praticien. Notamment, ils sont invariants au contraste, *i.e.*, ils conviennent au traitement d’images à faible contraste ainsi qu’aux images sous ou sur-exposées. Cette propriété est fondamentale pour traiter des images avec différents niveaux d’exposition.

Les filtres à base de MM reposent sur les treillis complets formés par les images numériques [12] qui sont bien définis dans le cas binaire et niveaux de gris car un ordre naturel existe. Au contraire, il n’existe toujours pas de consensus pour définir un ordre *naturel* sur les vecteurs. Le cas multivarié est encore plus subtil pour étendre les filtres morphologiques plus avancés tels que ceux nécessitant un *classement* total de valeurs. Par exemple, les opérateurs *connexes*, qui ne modifient que les valeurs de composantes connexes et qui préservent les contours de l’image, forment une classe de filtres largement utilisée. Ces filtres nécessitent que les valeurs soient totalement ordonnées.

Pour résoudre ce problème, de nombreuses tentatives (revues dans la Section 2) ont été faites pour définir un ordre total “sensé”. Dans cet article, nous adoptons une approche différente, suivant une méthodologie *par propriété*. Après un rappel concernant les arbres de composantes dans la Section 2 et leur relation avec les opérateurs connexes, nous expliquons dans la Section 3 nos exigences pour un Arbre de Composantes (AdC) multivarié, son processus de construction, et son utilisation pour étendre les filtres connexes aux données multivariées. Enfin nous concluons

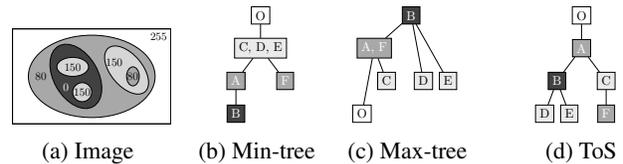


FIGURE 1 – Les différents arbres morphologiques.

dans la Section 4.

En raison du manque d’espace, ce papier présente uniquement le concept d’Arbre de Composantes Multivarié (ACM) ainsi que les ouvertures et les fermetures multivariées résultantes. Les expériences avec ces nouveaux outils seront exposés dans des travaux ultérieurs.

## 2 Problème et Etat de l’art

### 2.1 Filtre connexes & arbre morphologique

Les opérateurs de MM sont largement utilisés pour leur invariance au changement de contraste. Contrairement aux filtres par éléments structurants, les opérateurs *connexes* n’ont pas besoin d’a priori sur la géométrie de ce qui doit être filtré. Dans le cas binaire, les opérateurs connexes ne peuvent supprimer que certains ensembles de pixels connexes. Ceci s’applique aux images en niveaux de gris avec le principe de décomposition par ensemble de niveaux. Les opérateurs connexes sont ceux qui suppriment (ou fusionnent) certaines zones plates en modifiant leur niveau de gris. Ouvertures par reconstruction [17] et ouvertures par surface [18] et attributs [1] sont des exemples bien-connus de cette classe de filtres.

Un tournant est ensuite apparu dans [13], où une structure polyvalente, le Max-Tree, a été proposée. Avec le Max-Tree, filtrer une image se résume à élaguer certaines branches et à supprimer certains nœuds. En outre, il permet des filtrages beaucoup plus puissants grâce à des stratégies d'élagages avancées [15] et à des connectivités de seconde génération [10]. [6] proposent une structure similaire, l'Arbre des Formes (AdF), pour prendre en charge des opérateurs connexes auto-duaux. Le filtre de *grain* [5] est l'équivalent auto-dual de l'ouverture par *aire* et supprime les composantes connexes *extremes*. Plus important encore, les arbres morphologiques sont des représentations hiérarchiques de l'image. Ils permettent une analyse d'image multi-échelle et amènent un niveau supérieur de compréhension de l'image.

Plus formellement, soit une image  $u : \Omega \rightarrow E$  définie sur un domaine  $\Omega$  et prenant des valeurs sur un ensemble  $E$  muni d'une relation d'ordre  $\leq$ . Soit,  $[u < \lambda] = \{x \in \Omega, u(x) < \lambda\}$  (resp.  $[u > \lambda]$ ) avec  $\lambda \in E$  un ensemble de niveaux de  $u$  (aussi appelés coupe inférieure et supérieure). Nous notons  $\mathcal{CC}(X)$ ,  $X \in \mathcal{P}(\Omega)$  l'ensemble des composantes connexes de  $X$ . Si  $\leq$  est une relation *totale*, chaque paire de composantes  $X, Y \in \mathcal{CC}([u < \lambda])$  sont disjointes ou incluses. L'ensemble  $\mathcal{S} = \mathcal{CC}([u < \lambda])$  muni de la relation d'inclusion forme un arbre appelé *Min-Tree*. Son dual, défini sur les coupes supérieures est appelé *Max-Tree* (voir Figs. 1b and 1c). Le dernier arbre morphologique, l'*Arbre des Formes (AdF)* (voir Fig. 1d), est basé sur la fusion des Min et Max-Trees sont les trous des composantes ont été bouchées.

Par la suite, Min-Tree, Max-Tree et AdF seront utilisés de manière interchangeable, car nous allons implicitement considérer un ensemble de composantes  $\mathcal{S}$  doté de la relation d'inclusion (*i.e.*, la couverture de  $(\mathcal{S}, \subseteq)$ ) pour désigner la hiérarchie correspondante. Aussi, sans perte de généralité, nous allons considérer  $E = \mathbb{R}^n$ , et nous noterons  $u = \langle u_1, u_2, \dots, u_n \rangle$  où  $u$  est une image multibande et  $u_k$  sont des images scalaires.

## 2.2 Filtrages connexes pour les images multi-bandes

Une solution triviale consiste à appliquer les opérateurs de MM à l'image par canal et à recombinaison des résultats. Le traitement marginal est sujet au problème des fausses couleurs en créant des valeurs qui n'étaient pas dans l'image d'origine.

Puisque le problème de la définition d'un opérateur connexe multivarié réside dans l'absence d'un ordre total entre les valeurs, de nombreuses tentatives ont été faites pour définir un ordre total sur des données vectorielles. Deux stratégies sont principalement utilisées : la première donne la priorité à certaines (ou à toutes) composantes vectorielles, la seconde définit une fonction de projection et ordonne les vecteurs par leur rang (un pré-ordre total). Ordonner suivant une norme (luminance) ou suivant une distance à des couleurs de référence [9, 7] ont permis d'étendre les arbres morphologiques à la couleur mais laisse supposer que la géométrie est détenue par la luminance ou l'existence d'un fond uniforme et/ou réduit à un ensemble de valeurs. Pour obtenir un ordre plus "sensé", [16, 8] utilisent des techniques d'apprentissages pour déduire fonction de rang dépendante des données et possiblement du contexte local.

Dans [11], l'ordre partiel est considéré directement et on manipule la structure du graphe sous-jacente. Cependant, le *component-graph* est plus difficile à traiter et plus gourmand sur le plan computationnel. Enfin, dans [3], nous avons présenté

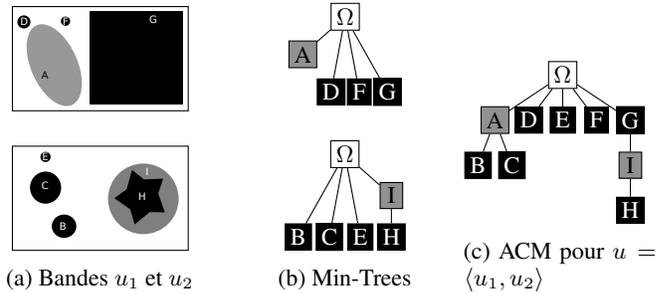


FIGURE 2 – Une image multi-bande, Min-Trees marginaux et le Min-Tree Multivarié qui remplit les exigences.

l'Arbre des Formes Multivarié (AdFM), extension de l'AdF aux images couleur. L'idée repose sur une stratégie de fusion de plusieurs AdFs et peut être transposée pour fusionner n'importe quel ensemble d'AdC.

## 3 Extension de l'AdC au multivarié

### 3.1 Exigences pour un AdC multivarié

La construction de l'ACM est conçue à partir des pré-requis suivants. Premièrement, nous avons un a priori, par canal, en ce qui concerne les objets que nous voulons détecter et leur arrière-plan (c'est-à-dire que nous connaissons la "direction" du contraste). Deuxièmement, nous n'avons pas d'a priori sur la manière dont les objets sont répartis entre les canaux, ils peuvent être visibles dans un, dans certains ou dans plusieurs canaux. L'ACM doit posséder les propriétés suivantes :

**Équivalence sur une bande.** Sur une image à bande unique, l'ACM doit être identique à l'AdC.

**Comportement conservateur.** Pour une composante connexe  $C$  de  $\mathcal{S}$ , si  $C$  est imbriqué ou disjoint de toutes les autres composantes connexes, elle doit apparaître dans l'ACM. Elle est illustrée dans la Fig. 2 avec deux Min-Trees ; puisqu'il n'y a pas de chevauchement entre les composantes des deux canaux de  $u$ , le Min-Tree Multivarié est intuitif. Notons que la "fusion" est purement basée sur l'inclusion, il n'y a pas de relation "inférieure à" (en termes de valeurs) entre les composantes de  $u_1$  et  $u_2$ . En effet, dans cet exemple,  $I$  devient un enfant de  $G$  alors que sa valeur dans  $u_2$  est supérieure à celle dans  $u_1$ .

**Invariance à tout changement marginal de contraste croissant.** Pour toute famille de fonction croissante  $\{\rho_1, \rho_2, \dots, \rho_n\}$ ,  $\mathcal{T}(\langle \rho_1(u_1), \rho_2(u_2), \dots, \rho_n(u_n) \rangle) = \mathcal{T}(u)$ . Cette propriété impose une propriété fondamentale de la MM : les objets bien contrastés et peu contrastés sont considérés de manière égale. L'intérêt est double : on peut gérer des canaux ayant des dynamiques différentes (là où les approches linéaires nécessitent généralement une normalisation correcte des données au préalable) et la représentation devient robuste à d'éventuels changements d'exposition sur certains canaux.

### 3.2 Construction de l'AdC Multivarié

#### 3.2.1 Obtention de l'arbre depuis la carte de profondeur

Le processus de fusion des AdC marginales consiste en 2 étapes principales : la construction d'une carte de profondeur  $\omega$  et la déduction d'un arbre depuis  $\omega$ . Commençons par motiver la construction de la carte de *profondeur*. La profondeur traduit

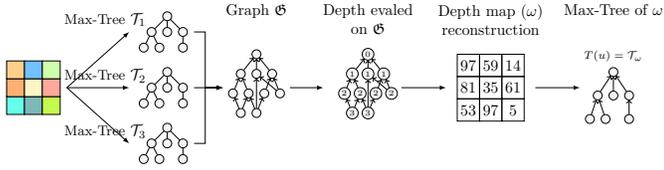


FIGURE 3 – Schema de construction de l’ACM.

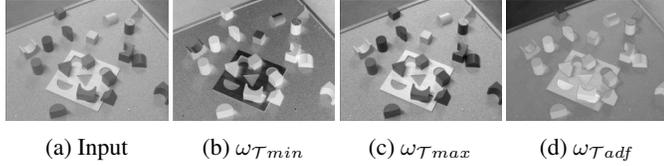


FIGURE 4 – Une image niveau de gris (a) et ses cartes de *profondeur* pour le Min-Tree (b), le Max-Tree (c) et l’AdF (d).

le niveau d’inclusion de l’objet. Plus un pixel apparaît clair dans la carte de profondeur, plus il se trouve profond dans la hiérarchie des objets. Dans [4], nous avons montré que, avec une hiérarchie et sa carte *depth* correspondante, on peut récupérer la hiérarchie initiale en calculant son max-tree. L’intérêt de la carte *depth* réside dans la manière dont elle abstrait la relation d’ordre de valeur sous-jacente entre les objets. En effet, quelle que soit la hiérarchie en entrée, qu’il s’agisse d’un Min-Tree, d’un Max-Tree ou d’un AdF, elle est équivalente au Max-Tree calculé sur la carte *depth*. Fig. 4 affiche les cartes de *profondeur* calculées à partir de différents arbres. Comme on peut le voir, les objets sombres apparaissent clairs pour le Min-Tree (b), les objets clairs apparaissent clairs pour le Max-Tree (c) et les formes les plus “incluses” apparaissent claires pour l’AdF (d). En fait, nous pouvons montrer que pour une image en niveaux de gris  $u$ ,  $\omega_{\mathcal{T}_{min}} = \rho(255 - u)$  et  $\omega_{\mathcal{T}_{max}} = \rho(u)$  avec  $\rho$  un changement de contraste croissant. Par conséquent, il est évident que le Max-Tree de  $\omega_{\mathcal{T}}$  est  $\mathcal{T}$  lui-même.

La carte *depth* conduit à une représentation alternative du contenu de l’image. Au lieu d’avoir une représentation basée sur la luminosité, nous avons maintenant une interprétation par pixel du *niveau d’inclusion* des objets. En d’autres termes, si un pixel apparaît plus lumineux que son voisin dans la carte *profondeur*, cela signifie qu’il appartient à un sous-objet.

### 3.2.2 Des arbres marginaux au graphe de composantes.

Les AdC  $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_n$  sont calculés sur chaque bande  $u_1, u_2, \dots, u_n$  de l’image auxquels on associe leur ensemble de composantes  $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_n$ . Soit  $\mathcal{S} = \bigcup \mathcal{S}_i$ , la Graphe de Composantes (GdC)  $\mathfrak{G}$  est la couverture de  $(\mathcal{S}, \subseteq)$ . Le GdC, décrit dans Fig. 3, est en fait le graphe d’inclusion de toutes les composantes connexes marginales. A noter que même si le GdC est un arbre, il n’est pas un arbre morphologique forcément valide (un point pouvant appartenir à plusieurs branches). Par conséquent, nous ne pouvons pas simplement extraire un arbre couvrant du GdC.

### 3.2.3 Du graphe à la carte de profondeur

Soit  $\rho$  la *profondeur* d’un noeud  $A$  dans  $\mathfrak{G}$ , i.e.,  $\rho(A)$  est la longueur du plus long chemin d’une composante  $A$  à partir de

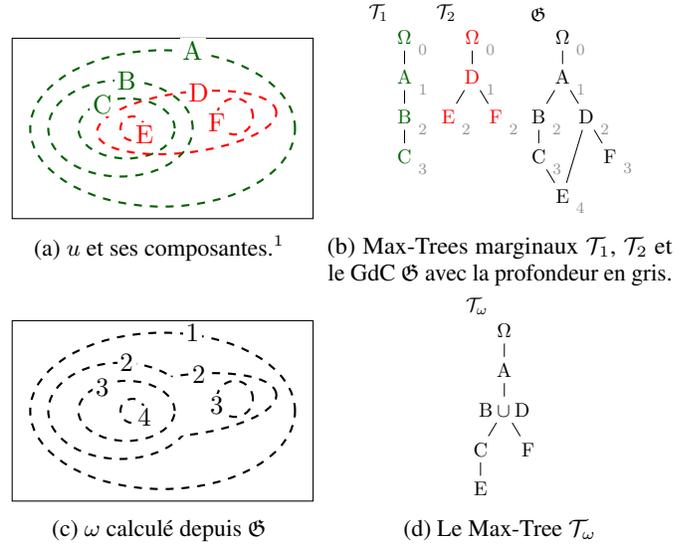


FIGURE 5 – Fusion des arbres à partir de la carte de profondeur. <sup>1</sup> $u_1(\Omega) < u_1(A) < u_1(B) < u_1(C)$  and  $u_2(\Omega) < u_2(D) < u_2(E) < u_2(F)$

la racine. Soit  $\omega : \Omega \rightarrow \mathbb{R}$  défini comme :

$$\omega(x) = \max_{X \in \mathcal{S}, x \in X} \rho(X) \quad (1)$$

La carte  $\omega$  associe à chaque point  $x$ , la profondeur de la composante la plus profonde contenant  $x$ . Soit  $\mathfrak{C} = \bigcup_{h \in \mathbb{R}} \mathcal{CC}([\omega \geq h])$ .  $(\mathfrak{C}, \subseteq)$  est en fait le Max-Tree de  $\omega$ . La méthode est illustrée dans Fig. 5 où une image à deux canaux possède des composantes marginales qui se chevauchent. Les arbres marginaux  $\mathcal{T}_1$  et  $\mathcal{T}_2$  sont fusionnés dans le GdC  $\mathfrak{G}$  pour lequel nous calculons la profondeur de chaque nœud (Fig. 5b). Ces valeurs sont reportées dans l’espace de l’image (Fig. 5c). Cette étape décide quelles composantes vont fusionner (ici  $B$  et  $D$  reçoivent la même valeur) en se basant sur le niveau d’inclusion et non plus sur les valeurs d’illumination. Une fois que  $\mathfrak{G}$  est transformé en  $\omega$ , il n’est plus possible de revenir en arrière et de récupérer les composantes connexes originales. C’est la seule partie “destructive” du processus.

### 3.2.4 Filtrage et reconstruction

Le filtrage s’effectue en deux étapes : d’abord un élagage qui supprime les nœuds et réaffecte les pixels ; puis une restitution lors de laquelle les pixels obtiennent leurs valeurs finales.

Soit  $\mathbb{P}$  un prédicat indiquant si un nœud doit être supprimé. Par exemple,  $\mathbb{P}(X) = \bar{X} > \alpha$  est un prédicat conservant les composantes de taille supérieure à  $\alpha$  (filtre de grain ou ouverture d’aire). La reconstruction après filtrage du Max-Tree  $\mathcal{T} = (\mathcal{S}, \subseteq)$  d’une image scalaire  $u$  est :

$$\tilde{u}(x) = \max_{X \in \mathcal{S}, x \in X | \mathbb{P}(X)} \lambda(X) \text{ avec } \lambda(X) = \min_{x \in X} u(x) \quad (2)$$

L’Eq. (2) décrit un filtrage **direct**, les pixels sont affectés au niveau du dernier nœud encore en vie dans la hiérarchie. Pour définir la restitution sur les données multivariées, on pourrait étendre l’Eq. (2). Chaque pixel serait alors affecté à l’infimum des pixels de son nœud. Cette stratégie de restitution (appelons-la *Rinf*) conduit à plusieurs artefacts visibles dans la Fig. 6b. D’abord cette restitution peut affecter des pixels appartenant à

des composantes devant être conservées. Deuxièmement, elle crée des fausses couleurs qui ne sont pas présentes dans l’image d’origine. La peinture *circles* de Kandinsky (Fig. 6) permet d’illustrer ces problèmes. En utilisant notre méthode, des régions voisines avec des couleurs non comparables (comme celles de *A*, *B* et *C*) fusionnent à la même profondeur et la stratégie *Rinf* leur affecte une couleur aberrante.

Ce problème est aussi présent avec la stratégie *marginale* (Fig. 6d). Ces fausses couleurs sont dues à la fusions des résultats de filtrages marginaux. Ensuite, les bordures des objets sont plus floues et on crée des zones plates plus petites que la taille de grain de filtrage.

La stratégie de restitution proposée dans [14] permet de conserver les valeurs de pixels d’origine lorsqu’ils sont dans un nœud “vivant” et utilise la couleur du contour de la composante sinon. Les résultats sont exposés Fig. 6c. Elle combine la force d’une approche vectorielle qui produit des zones plates “réelles” sans flou de contour et un filtrage de couleur similaire à celui obtenu par filtrage marginal. Cependant, il reste des problèmes de restitution pour les nœuds qui fusionnent des régions non comparables, car une seule couleur les représente. Cela produit des artefacts comme sur *B* et *C* où les pixels orange et magenta ont été colorisés en cyan à partir de *A*.

## 4 Conclusion

Nous avons présenté les idées préliminaires pour étendre les filtres connexes aux images multibandes. Notre approche repose sur l’extension des arbres de composantes aux données pour lesquelles il n’existe d’ordre naturel. Au lieu d’imposer un ordre total arbitraire, la méthode fusionne des composantes connexes à partir d’arbres de composantes marginaux. Elle crée un ordonnancement local des pixels, dépendant de leur niveau d’inclusion dans chaque hiérarchie. Au-delà du filtrage, notre méthode produit une représentation hiérarchique de l’image, l’ACM, qui permet d’étudier les formes de l’image [2]. Dans le futur, nous prévoyons de trouver un moyen de donner la priorité à certaines bandes tout en conservant l’idée d’ “ordonner par niveau d’inclusion”. Aussi, nous voulons étudier l’impact de la corrélation des données sur le processus de fusion pour éventuellement choisir un meilleur espace couleur d’entrée.

## Références

- [1] E. J. Breen and R. Jones. Attribute openings, thinnings, and granulometries. *CVIU*, 64(3) :377–389, 1996.
- [2] F. Cao, J.-L. Lisani, J.-M. Morel, P. Musé, and F. Sur. *A Theory of Shape Identification*, vol. 1948 of *LNCS*. Springer, 2008.
- [3] E. Carlinet and T. Géraud. MToS : A tree of shapes for multivariate images. *ITIP*, 24(12) :5330–5342, 2015.
- [4] E. Carlinet, T. Géraud, and S. Crozet. The tree of shapes turned into a max-tree : A simple and efficient linear algorithm. In *Proc. of ICIP*, pp. 1488–1492, 2018.
- [5] V. Caselles and P. Monasse. Grain filters. *Journal of Mathematical Imaging and Vision*, 17(3) :249–270, 2002.
- [6] V. Caselles, B. Coll, and J.-M. Morel. Topographic maps and local contrast changes in natural images. *International Journal of Computer Vision*, 33(1) :5–27, 1999.
- [7] B. Coll and J. Froment. Topographic maps of color images. In *Proc. of ICPR*, vol. 3, pp. 609–612, 2000.
- [8] O. Lézoray. Complete lattice learning for multivariate mathematical morphology. *JVCIR*, 35 :220–235, 2016.



(a) Image et sa carte de profondeur



(b) *Rinf*

(c) *RNC*

(d) Marginale

FIGURE 6 – Comparaison des ouvertures de tailles 500, 4000 et 16000 (de haut en bas) avec différentes stratégies de restitution.

- [9] B. Naegel and N. Passat. Component-trees and multi-value images : A comparative study. In *Proc. of ISMM*, vol. 5720 of *LNCS*, pp. 261–271. Springer, 2009.
- [10] G. K. Ouzounis and M. H. F. Wilkinson. Mask-based second-generation connectivity and attribute filters. *IEEE Trans. on PAMI*, 29(6) :990–1004, 2007.
- [11] N. Passat and B. Naegel. An extension of component-trees to partial orders. In *Proc. of ICIP*, pp. 3933–3936, 2009.
- [12] C. Ronse. Why mathematical morphology needs complete lattices. *Signal processing*, 21(2) :129–154, 1990.
- [13] P. Salembier, A. Oliveras, and L. Garrido. Antiextensive connected operators for image and sequence processing. *IEEE Trans. on Image Processing*, 7(4) :555–570, 1998.
- [14] F. Tushabe and M. Wilkinson. Color processing using max-trees : A comparison on image compression. In *Proc. of ICSAI*, pp. 1374–1380, 2012.
- [15] E. R. Urbach and M. H. F. Wilkinson. Shape-only granulometries and grey-scale shape filters. In *Proc. of Intl. Symp. on Mathematical Morphology (ISMM)*, vol. 2002, pp. 305–314, 2002.
- [16] S. Velasco-Forero and J. Angulo. Random projection depth for multivariate mathematical morphology. *IEEE Journal of Selected Topics in Signal Processing*, 6(7) :753–763, 2012.
- [17] L. Vincent. Morphological grayscale reconstruction in image analysis : Applications and efficient algorithms. *IEEE Trans. on Image Processing*, 2(2) :176–201, 1993.
- [18] L. Vincent. Morphological area openings and closings for grey-scale images. In *Shape in Picture : Mathematical Description of Shape in Grey-level Images*, pp. 197–208. Springer, 1994.