

Une approche morphologique de segmentation interactive avec l’arbre des formes couleur

Edwin CARLINET^{1,2}, Thierry GÉRAUD¹

¹EPITA Research and Development Laboratory (LRDE)
14-16 rue Voltaire, 94270 Le Kremlin-Bicêtre, France

²Université Paris-Est, LIGM, Équipe A3SI, ESIEE
Cité Descartes, BP 99, 93162 Noisy-le-Grand Cedex, France

edwin.carlinet@lrde.epita.fr, thierry.geraud@lrde.epita.fr

Résumé – L’Arbre des Formes (AdF) est un arbre morphologique à la fois auto-dual et invariant par changement de contraste. Il fournit une représentation haut-niveau de l’image, intéressante pour de nombreuses tâches de traitement d’images. Malgré son potentiel et sa simplicité, il reste largement sous-utilisé en reconnaissance des formes et vision par ordinateur. Dans cet article, nous présentons une méthode de segmentation interactive qui s’effectue simplement en manipulant cet arbre. Pour cela, nous nous appuyons sur une représentation récemment définie : “l’Arbre des Formes Couleur”. La méthode de segmentation interactive que nous proposons ne requiert aucun apprentissage statistique ; néanmoins elle obtient des résultats qui rivalisent avec ceux de l’état de l’art. Bien que préliminaires, les résultats obtenus mettent en avant le potentiel et l’intérêt des méthodes travaillant dans l’espace des formes.

Abstract – The Tree of Shapes is a self-dual and contrast invariant morphological tree that provides a high-level hierarchical representation of images, suitable for many image processing tasks. Despite its powerfulness and its simplicity, it is still under-exploited in pattern recognition and computer vision. In this paper, we show that an interactive segmentation can be achieved with some simple tree processings. To that aim, we rely on the “Color Tree of Shapes”, recently defined. We propose a method for interactive segmentation that does not involve any statistical learning, yet yielding results that compete with state-of-the-art approaches. Although they are preliminary, they highlight the potential of such an approach that works in the shape space.

1 Introduction

L’Arbre des Formes (AdF) [6, 12] (aussi connu sous l’arbre des lignes de niveau) a connu dernièrement un regain d’intérêt dans la communauté de traitement d’images car il permet de représenter une image sous une forme hiérarchique à la fois efficace et facilement manipulable. Grâce à l’abstraction offerte par l’AdF, des opérations avancées de traitement d’images s’effectuent ainsi de manière simple [4]. Par exemple, la simplification d’une image se traduit par une sélection ou une suppression de nœuds dans l’arbre. De même, débruiter une image correspond en fait à élaguer des branches de ce même arbre. La simplicité et la versatilité du cadre de travail n’enlève rien à son potentiel, et des traitements basés sur l’AdF se sont montrés aussi efficaces que les travaux les plus récents dédiés à certaines applications. Ainsi, les “Régions de Morses basées sur des Arbres” (TBMR) [14] parviennent au niveau des Régions Stables Maximales (MSER) pour la mise en correspondance de scènes. Les profils d’attributs basés sur l’AdF obtiennent les résultats à l’état de l’art en matière de classification d’images hyper-spectrales, etc. La versatilité de l’AdF n’est plus à démontrer tant ses champs d’application sont divers : visualisation, filtrage, recalage, simplification (voir [14] pour

une liste non exhaustive d’applications basées sur l’AdF). Ce n’est pas surprenant si l’AdF obtient de si bons résultats, mais cohérent avec ses propriétés mathématiques [2]. L’AdF est un support pour des opérateurs connexes auto-duaux et invariants par changement de contraste. En d’autres termes : **1.** il permet de traiter les objets foncés sur fond clair et inversement de manière *symétrique*, **2.** il est insensible au contraste et donc capable de détecter les objets faiblement contrastés, **3.** il ne déplace pas les contours de l’image. Dit encore plus simplement, *il possède les propriétés d’une représentation d’image que l’on désire pour le traitement d’images.*

Cependant, l’AdF est uniquement défini pour les images en niveaux de gris puisqu’il requiert un ordre total sur les valeurs. Nous avons montré qu’imposer un ordre total entre les couleurs mène à des comportements “étranges” [3]. En poursuivant les travaux menés dans [3], nous avons formalisé un moyen de calculer l’AdF sur des données multivariées qui ne nécessite pas d’imposer un ordre total : l’Arbre des Formes Couleur (AdFC) [5]. A l’instar de l’AdF sur les images scalaires, l’AdFC vérifie des invariances similaires à savoir l’invariance à tout changement et/ou inversion de contraste *marginal*. Formellement, soit une image \mathbf{u} dont les valeurs sont dans \mathbb{R}^3 , $\mathbf{u} = (u_1, u_2, u_3)$, et soit la famille $\mathbf{F} = (F_1, F_2, F_3)$ où chaque

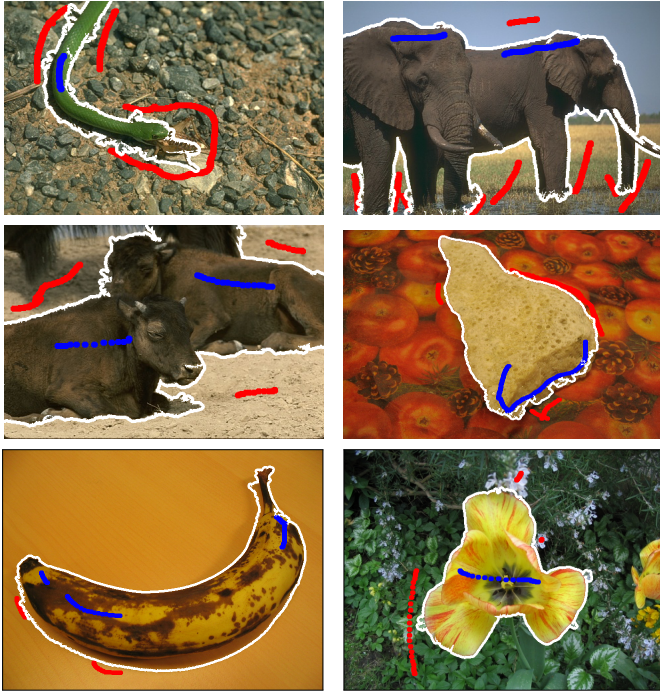


Figure 1: Segmentation interactive avec notre approche. L'utilisateur fournit les marques rouges et bleues qui définissent le fond B et l'objet F . La méthode calcule la frontière F/B représentée en blanc.

$F_i : \mathbb{R} \rightarrow \mathbb{R}$ sont des fonctions strictement monotones ($\forall x, y \in \mathbb{R}, x < y \Rightarrow F(x) < F(y)$ or $F(x) > F(y)$). Alors, l'AdFC $T(\mathbf{u})$ est invariant par changement et/ou inversion de contraste marginal, i.e., $T(\mathbf{F}(\mathbf{u})) = T(\mathbf{u})$.

Dans cet article, nous allons montrer la pertinence de l'AdFC pour la segmentation d'image dans un contexte de segmentation interactive où l'utilisateur positionne des marqueurs dans des régions d'intérêt. Dans [5], nous avons évoqué rapidement le sujet à titre d'illustration, cet article vise donc à détailler plus précisément le fonctionnement de la méthode et ses avantages par rapport aux autres approches à l'état de l'art.

L'article est organisé comme suit. Dans la Section 2, nous rappelons les bases de l'Arbre des Formes (AdF) et de son extension couleur et nous présentons le schéma de la méthode de segmentation basée sur l'AdFC. La Section 3 présente les différences entre notre méthode et les autres travaux de segmentation interactive. Dans la Section 4, nous rentrons plus en détails sur son fonctionnement. Finalement, la Section 5 montre quelques résultats de notre approche et nous concluons dans la Section 6.

2 Aperçu général

2.1 L'Arbre des Formes Couleur

Soit une image $u : \Omega \rightarrow E$ définie sur un domaine Ω et à valeurs dans E muni d'une relation d'ordre \leq . Soit $[u < \lambda]$

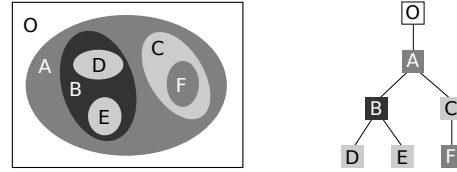


Figure 2: Une image (gauche) et son AdF (droite).

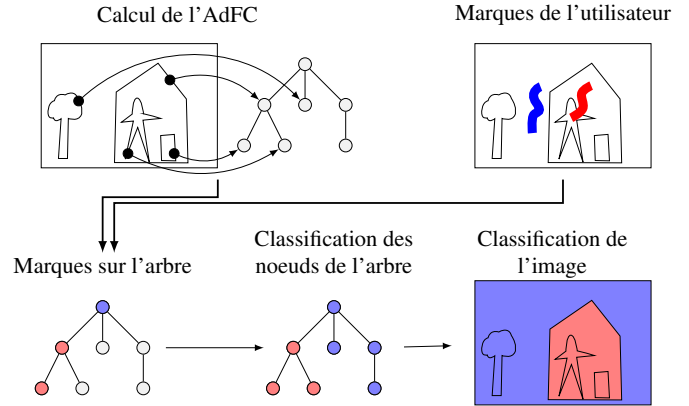


Figure 3: Schéma de la méthode de segmentation interactive.

(resp. $[u > \lambda]$) avec $\lambda \in \mathbb{R}$ l'ensemble des niveaux inférieurs (resp. supérieurs) de u défini par $[u < \lambda] = \{x \in \Omega, u(x) < \lambda\}$. On note $\mathcal{CC}(X), X \in \mathcal{P}(E)$, l'ensemble des composantes connexes de X . Si \leq est une relation d'ordre totale, deux composantes connexes quelconques $X, Y \in \mathcal{CC}([u < \lambda])$ sont soit disjointes soit incluses. L'ensemble $\{\Gamma, \Gamma \in \mathcal{CC}([u < \lambda])\}_\lambda$ muni de la relation d'inclusion forme un arbre : le *min-tree*. Son dual, basé sur les lignes de niveaux supérieures, est le *max-tree*. Étant donné l'opérateur de bouchage de trou \mathcal{H} , on appelle *forme* n'importe quel élément de l'ensemble:

$$\mathcal{S} = \{ \mathcal{H}(\Gamma), \Gamma \in \mathcal{CC}([u < \lambda]) \cup \mathcal{CC}([u > \lambda]) \}_\lambda.$$

Si \leq est totale, deux formes quelconques sont soit disjointes soit incluses, ainsi la couverture de (\mathcal{S}, \subseteq) forme un arbre appelé l'*Arbre des Formes (AdF)* (voir Figure 2).

Si \leq n'est pas totale (e.g., sur les couleurs), deux formes peuvent se chevaucher sans s'inclure et (\mathcal{S}, \subseteq) ne forme plus un arbre. Au lieu de définir un nouvel ordre total sur les valeurs, l'Arbre des Formes Couleur (AdFC) [5] tente de fusionner des Arbres des Formes, calculés marginalement sur chaque canal de l'image, en une seule structure (un graphe). Cette fusion est basée uniquement sur la relation d'inclusion. Ensuite, un arbre est extrait depuis ce graphe en utilisant la topologie du graphe uniquement et non plus sur les valeurs d'origines. Cet arbre vérifie de nombreuses propriétés dont l'invariance par changement et/ou inversion de contraste marginal.

2.2 Description de la méthode

Le problème de segmentation interactive est le suivant. Soit deux ensembles de points F and B in $\mathcal{P}(\Omega)$ représentant les



Figure 4: Segmentation interactive avec ou sans trou. En travaillant dans l’espace des formes, marquer la région externe de l’objet permet de récupérer la région entière. Notons que cela n’empêche pas d’obtenir des objets avec des trous si l’utilisateur le souhaite.

marques de l’utilisateur pour la classe *fond* (\mathcal{F}) et la classe *objet* (\mathcal{B}), on cherche à étiqueter tous les points de Ω par l’une de ces deux classes.

En considérant la distance entre deux points (p, p') de Ω :

$$d_{TV}(p, p') = \min_{C_{pp'}} \int_0^1 |\nabla u(C_{pp'}(t)) \cdot \dot{C}_{pp'}(t)| dt, \quad (1)$$

où $C_{pp'}(t)$ est un chemin dans Ω de p vers p' , on peut calculer la distance entre un point x et les marqueurs F et B puis affecter à x l’étiquette de la classe la plus proche. Cette approche a été employée par [13, 1]. Néanmoins, dans Dubrovina et al. [8], les auteurs ont montré que le calcul exact de cette distance nécessite une approche par ensemble de niveaux et requiert l’AdF.

Ainsi, l’idée fondamentale de la méthode est d’utiliser la représentation de l’image par l’AdFC de l’image au lieu de travailler directement sur son domaine d’origine. On applique ensuite le même principe que précédemment (une classification au plus proche voisin) mais en utilisant la topologie de l’arbre en lieu et place de la topologie de l’espace 2D. La segmentation finale est obtenue en reconstruisant l’image depuis l’arbre où tous ses nœuds auront été annotés. La méthode peut donc se résumer aux étapes suivantes (voir aussi la Figure 3):

1. Calculer l’AdFC $T(\mathbf{u})$ de l’image \mathbf{u} ,
2. Valuer les arrêtes de $T(\mathbf{u})$ par la distance inter-nœuds (voir la Section 4 pour plus de détails)
3. Étiqueter les nœuds de $T(\mathbf{u})$ depuis les marques entrées par l’utilisateur. On obtient ainsi deux ensembles de nœuds marqués pour \mathcal{F} et \mathcal{B} ,
4. Affecter tous les autres nœuds à l’une des classes \mathcal{F} ou \mathcal{B} en calculant leur distance aux nœuds marqués (en utilisant la topologie de l’arbre), puis en assignant la classe du nœud le plus proche.
5. Reconstruire l’image depuis l’arbre annoté,
6. Nettoyer : conserver seulement les composantes connexes *objet* significatives (suppression des régions isolées de petite taille) .

Un avantage certain de travailler dans l’espace de formes est la capacité de récupérer de larges régions d’intérêts qui ne sont pas marqués par l’utilisateur. Cette propriété est intéressante pour les objets composés d’autres objets. Une forme étant une composante sans trous, il est suffisant de sélectionner la partie frontalière de l’objet englobant pour récupérer tous les objets (voir la Figure 4).

Un second avantage de l’approche est qu’elle requiert aucune modélisation statistique de la région. Elle utilise unique-

ment les ensembles de niveau ce qui permet de récupérer de larges composantes avec peu de marques. La quantité de marques requises dépend en fait du nombre de lignes de niveau qui séparent le fond de l’objet.

3 Travaux similaires

Notre approche est similaire à celle proposée par Dubrovina et al. [8] puisque nous utilisons aussi l’AdF (dans notre cas l’AdFC) pour effectuer la classification des nœuds puis reconstruire l’image depuis l’arbre annoté. La différence principale réside dans le fait que Dubrovina et al. [8] utilisent l’arbre sur une carte de probabilité où chaque pixel représente la vraisemblance d’être un élément *objet*. Cela implique une modélisation statistique des marques de l’utilisateur et donc dépend de l’efficacité de la modélisation de la fonction de probabilité. Leur travail est en fait une extension de [1] donnant une meilleure évaluation de distance géodésique entre les pixels non-annotés et les marques. Dans [1], les auteurs ont remarqué l’importance de la méthode de modélisation statistique qui a permis de meilleures segmentations tout en nécessitant moins de marques par rapport à leurs travaux précédents [13]. Néanmoins, il s’avère que la qualité de l’estimation (donc de la segmentation) dépend fortement du rapport entre la complexité de la région et du nombre de marques rentrées par l’utilisateur. En fait, là où la plupart des algorithmes d’édition d’images et le *matting* d’images repose sur de l’apprentissage statistique pour la modélisation *objet / fond*, notre méthode s’en affranchit.

4 Détails algorithmiques

La construction de l’AdFC est dehors du champ de cet article. Nous détaillons ici la manière par laquelle l’AdF est utilisé pour segmenter (les étapes 2–6 de la Section 2.2).

La seconde étape de l’algorithme consiste en la valuation d’une distance sur les arrêtes de l’arbre. Contrairement aux arbres morphologiques standards, un nœud de l’AdFC peut être associé à plusieurs couleurs. Par conséquence, on considère pour chaque nœud sa couleur moyenne et les arrêtes sont valuées par la distance (dans l’espace L^*b^*) entre les nœuds. Ensuite, nous avons besoin d’étiqueter les nœuds par l’une des classes \mathcal{F} ou \mathcal{B} à partir des marques saisies par l’utilisateur. Dans quelques rares cas, un nœud peut recevoir des informations contradictoires (si l’utilisateur tague des points du même nœud en tant que *fond* et *objet*). Dans cette situation, le nœud est annoté par la classe majoritaire.

La 4^e étape consiste à calculer pour chaque nœud (forme) non annoté S , sa distance $d_{\mathcal{F}}$ and $d_{\mathcal{B}}$ aux nœuds marqués *objet* et *fond* les plus proches. Ce calcul peut être effectué en deux étapes. On note $par(S)$, le parent de S dans T , et $d(S, par(S))$ la distance entre la couleur moyenne de S et celle de $par(S)$. À l’initialisation, $d_{\mathcal{B}}(S) = 0$ si le nœud possède des points *fond*, $+\infty$ sinon. Une première passe parcourt l’arbre depuis les feuilles vers la racine et calcule :

$d_{\mathcal{B}}(\text{par}(S)) = \min(d_{\mathcal{B}}(\text{par}(S)), d_{\mathcal{B}}(S) + d(S, \text{par}(S)))$.

Puis la seconde passe va de la racine vers les feuilles et calcule :

$d_{\mathcal{B}}(S) = \min(d_{\mathcal{B}}(S), d_{\mathcal{B}}(\text{par}(S)) + d(S, \text{par}(S)))$.

Le même processus s'effectue pour le calcul de $d_{\mathcal{F}}$ et finalement, un nœud est annoté par $\arg \min_{C \in \{\mathcal{F}, \mathcal{B}\}} d_C(S)$.

Dans l'arbre $T(\mathbf{u})$, tous les points appartiennent à un nœud unique. La reconstruction de l'image consiste donc à affecter à chaque point la classe du nœud auquel il appartient. Pour finir, l'étape de post-traitement consiste à nettoyer et supprimer les composantes connexes non significatives, à savoir les régions de tailles inférieures à la moitié de la plus grande.

La méthode a deux étapes principales: la construction de l'AdFC et la classification des nœuds. La première est quasi-linéaire en nombre de pixels [9, 7, 5], la seconde est linéaire en nombre de nœuds (inférieur aux nombres de pixels). La complexité globale de la méthode est donc quasi-linéaire, dominée par la construction de la structure. En pratique, la totalité du processus est de l'ordre de 2s pour une image 512×512 .

5 Expérimentations et discussions

Figure 1 montre quelques résultats obtenus pour la segmentation interactive en utilisant notre approche (implémentée avec notre plate-forme Olena [10, 11]). À ce stade, et malgré sa simplicité, notre méthode rivalise avec l'état de l'art tel que GrabCut d'un point de vue qualitatif. Il semble cependant prématuré d'effectuer une comparaison approfondie (quantitative) avec ces méthodes alors qu'il subsiste quelques problèmes que nous devons d'abord résoudre. Un cas typique d'échec de notre méthode est par exemple lorsque l'objet est traversé par plusieurs lignes de niveau du fond. Dans ce cas, on ne parvient pas à séparer *fond* et *objet* correctement. Nous avons observé récemment que choisir un point à l'infini dans le fond ou dans l'objet (autrement dit, en changeant l'enracinement de l'arbre) pouvait en partie le résoudre. Malgré ces problèmes et au vu de la qualité des résultats actuels, une classification directe dans l'espace des formes semble une approche prometteuse et l'AdFC apparaît ainsi comme une structure adaptée à la segmentation.

6 Conclusion

Dans cet article, nous avons montré la versatilité et le potentiel de l'AdFC à des fins de segmentation. Nous avons pour cela proposé une méthode de classification à base de marqueurs, utilisant l'AdFC, qui ne requiert aucun apprentissage statistique et qui malgré tout, obtient des résultats rivalisant avec l'état de l'art. En perspective de ce travail, même si notre méthode est d'ores et déjà robuste sans modélisation statistique, nous prévoyons d'intégrer des stratégies de classification plus avancées. Nous pensons conserver l'idée fondamentale de travailler (en partie) dans l'espace des formes, et nous souhaitons comparer quantitativement les résultats de notre approche avec ceux des méthodes de l'état de l'art.

References

- [1] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *Proc. of Intl. Conf. on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007.
- [2] F. Cao, J.-L. Lisani, J.-M. Morel, P. Musé, and F. Sur. *A Theory of Shape Identification*, volume 1948 of *Lecture Notes in Mathematics*. Springer, 2008.
- [3] E. Carlinet and T. Géraud. Getting a morphological tree of shapes for multivariate images: Leads, traps, and pitfalls. In *Proc. of IEEE Intl. Conf. on Image Processing (ICIP)*, pages 615–619, Paris, France, 2014.
- [4] E. Carlinet and T. Géraud. A comparative review of component tree computation algorithms. *IEEE Transactions on Image Processing*, 23(9):3885–3895, September 2014.
- [5] E. Carlinet and T. Géraud. A color tree of shapes with illustrations on filtering, simplification and segmentation. In *Proc. of Intl. Symp. on Mathematical Morphology (ISMM)*, pages 363–374, Reykjavik, Iceland, 2015.
- [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] S. Crozet and T. Géraud. A first parallel algorithm to compute the morphological tree of shapes of n D images. In *Proc. of IEEE Intl. Conf. on Image Processing (ICIP)*, pages 2933–2937, Paris, France, 2014.
- [8] A. Dubrovina, R. Hershkovitz, and R. Kimmel. Image editing using level set trees. In *Proc. of IEEE ICIP*, pages 4442–4446, Paris, France, 2014.
- [9] T. Géraud, E. Carlinet, S. Crozet, and L. Najman. A quasi-linear algorithm to compute the tree of shapes of n -D images. In *Proc. of Intl. Symp. on Mathematical Morphology (ISMM)*, pages 98–110, Uppsala, Sweden, 2013.
- [10] R. Levillain, T. Géraud, and L. Najman. Milena: Write generic morphological algorithms once, run on many kinds of images. In *Proc. of Intl. Symp. on Mathematical Morphology (ISMM)*, pages 295–306. Springer, 2009.
- [11] R. Levillain, T. Géraud, and L. Najman. Why and how to design a generic and efficient image processing framework: The case of the Milena library. In *Proc. of IEEE Intl. Conf. on Image Processing (ICIP)*, pages 1941–1944, Hong Kong, September 2010.
- [12] P. Monasse and F. Guichard. Fast computation of a contrast-invariant image representation. *IEEE Trans. on Image Processing*, 9(5):860–872, 2000.
- [13] A. Protiere and G. Sapiro. Interactive image segmentation via adaptive weighted distances. *IEEE Trans. on Image Processing*, 16(4):1046–1057, 2007.
- [14] Y. Xu, P. Monasse, T. Géraud, and L. Najman. Tree-based morse regions: A topological approach to local feature detection. *IEEE Transactions on Image Processing*, 23(12):5612–5625, 2014.