



Pôle de recherche et d'enseignement supérieur

École doctorale MSTIC

PH.D. THESIS

Thèse soumise pour l'obtention du grade de
Docteur de l'Université Paris Est en Informatique

27 novembre 2015

A TREE OF SHAPES FOR MULTIVARIATE IMAGES

EDWIN CARLINET

Directeur de thèse: Jean Serra
Co-directeur de thèse: Thierry Géraud

Jury :

Président :

Rapporteurs : Coloma BALLESTER, Pr., Universitat Pompeu Fabra
Philippe SALEMBIER, Pr., Universitat Politècnica de Catalunya
Ludovic MACAIRE, Pr., Université Lille 1

Examineurs : Jesús ANGULO, Pr., MINES ParisTech
Pascal MONASSE, Pr., École des Ponts ParisTech

RÉSUMÉ

De nombreuses applications issues de la *vision par ordinateur* et de la *reconnaissance des formes* requièrent une analyse de l'image multi-échelle basée sur ses régions. De nos jours, personne ne considérerait une approche orientée « pixel » comme une solution viable pour traiter ce genre de problèmes. Pour répondre à cette demande, la Morphologie Mathématique a fourni des représentations hiérarchiques des régions de l'image telles que l'Arbre des Formes (AdF). L'AdF représente l'image par un arbre d'inclusion de ses lignes de niveaux. L'AdF est ainsi auto-dual et invariant au changement de contraste, ce qui fait de lui une structure bien adaptée au traitements d'images de haut niveau.

Néanmoins, il est seulement défini aux images en niveaux de gris et la plupart des tentatives d'extension aux images multivariées (*e.g.* en imposant un ordre total « arbitraire ») ne sont pas satisfaisantes.

Dans ce manuscrit, nous présentons une nouvelle approche pour étendre l'AdF *scalaire* aux images multivariées : l'Arbre des Formes Multivarié (AdFM). Cette représentation est une « fusion » des AdFs calculés marginalement sur chaque composante de l'images. On vise à fusionner les *formes* marginales de manière « sensée » en préservant un nombre maximal d'inclusion. La méthode proposée a des fondements théoriques qui consistent en l'expression de l'AdF par une carte topographique de la variation totale curvilinéaire depuis la bordure de l'image. C'est cette reformulation qui a permis l'extension de l'AdF aux données multivariées. De plus, l'AdFM partage des propriétés similaires avec l'AdF scalaire ; la plus importante étant son invariance à tout changement ou inversion de contraste marginal (une sorte d'*auto-dualité* dans le cas multidimensionnel).

Puisqu'il est évident que, vis-à-vis du nombre sans cesse croissant de données à traiter, nous avons besoin de techniques *rapides* de traitement d'images, nous proposons un algorithme efficace qui permet de construire l'AdF en temps quasi-linéaire vis-à-vis du nombre de pixels et quadratique vis-à-vis du nombre de composantes. Nous proposons également des algorithmes permettant de manipuler l'arbre, montrant ainsi que, en pratique, l'AdFM est une structure facile à manipuler, polyvalente, et efficace.

Finalement, pour valider la pertinence de notre approche, nous proposons quelques expériences testant la robustesse de notre structure aux composantes non-pertinentes (*e.g.* avec du bruit ou à faible dynamique) et nous montrons que ces défauts n'affectent pas la structure globale de l'AdFM. De plus, nous proposons des applications concrètes utilisant l'AdFM. Certaines sont juste des modifications mineures aux méthodes employant d'ores et déjà l'AdF scalaire mais adaptées à notre nouvelle structure. Par exemple, nous utilisons l'AdFM à des fins de filtrage, segmentation, classification et de détection d'objet. De ces applications, nous montrons ainsi que les méthodes basées sur l'AdFM surpassent généralement leur analogue basé sur l'AdF, démontrant ainsi le potentiel de notre approche.

Keywords : Arbre des Formes, opérateurs connexes, morphologie mathématique, ensemble de niveaux, hiérarchies, images multivariées, images couleur.

RÉSUMÉ LONG

RÉSUMÉ

L'Arbre des Formes (AdF) est un arbre morphologique qui fournit une représentation hiérarchique de l'image auto-duale et invariante par changement de contraste. De ce fait, il est adapté à de nombreuses applications de traitement d'images. Néanmoins, on se heurte à des problèmes avec l'AdF lorsqu'on doit traiter des images couleurs car sa définition tient uniquement en niveaux de gris. Les solutions les plus courantes sont alors d'effectuer un traitement composante par composante (marginal) ou d'imposer un ordre total. Ces solutions ne sont généralement pas satisfaisantes et font survenir des problèmes (des artefacts de couleur, des pertes de propriétés. . .) Nous insistons ici sur la nécessité d'une représentation à la fois auto-duale et invariante par changement de contraste et nous proposons une méthode qui construit un arbre unique, l'Arbre des Formes Multivarié (AdFM), en fusionnant des formes issues des composantes marginales tout en préservant les propriétés intrinsèques de l'arbre. Cette méthode s'affranchit de toute relation d'ordre totale en utilisant uniquement la relation d'inclusion entre les formes et en effectuant une fusion dans l'espace des formes. Finalement, nous montrerons la pertinence de notre méthode et de la structure en les illustrant à travers diverses applications de vision par ordinateur.

1 INTRODUCTION

L'AdF [32, 28] est une représentation hiérarchique de l'image témoignant de l'inclusion de ses lignes de niveaux. L'efficacité de cette structure pour le traitement d'images réside dans ses propriétés. D'abord, c'est une représentation morphologique (donc invariante par changement de contraste) basée sur l'inclusion des composantes connexes de l'image obtenues à différents niveaux de seuillage. En conséquent, un filtrage basique de cet arbre est un filtre connexe, c'est-à-dire un filtre qui ne déplace pas les contours des objets mais se contente de supprimer ou préserver certains d'entre eux [112]. Ensuite, non seulement cette structure est invariante par changement de contraste global mais elle l'est aussi *localement* [21]. Cette propriété est très importante dans de nombreuses applications de vision par ordinateur où la robustesse au changement d'illumination est un véritable challenge, e.g. pour la mise en correspondance de scènes ou la reconnaissance d'objets. La fig. 1d montre cette invariance en simulant un changement d'illumination directement avec l'AdF, on a donc la même représentation arborescente sur cette image que sur l'image d'origine montrée sur la fig. 1b. Enfin, au delà de son invariance au changement de contraste, l'AdF est aussi une représentation auto-duale de l'image. Cette propriété est fondamentale dans un contexte où les structures peuvent apparaître à la fois sur un fond

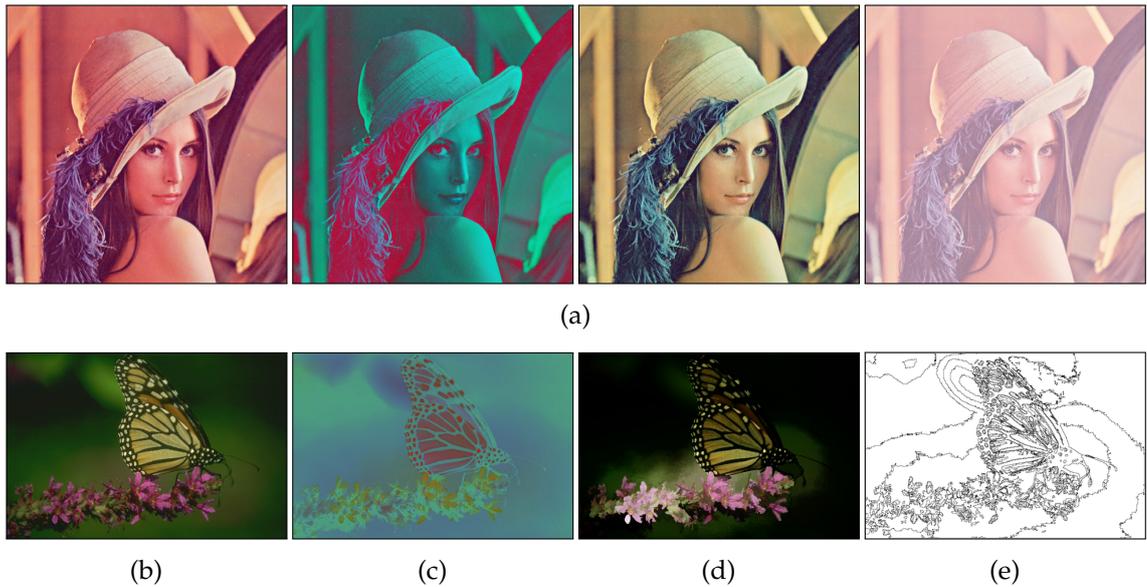


FIGURE 1: (a) A propos du besoin d'invariance au changement et inversion de contraste (quelques exemples de changement et/ou inversion de contraste sur Lena). (b) Image originale. (c) Changement et/ou inversion de contraste indépendant sur chaque canal de (b). (d) Changement de contraste local sur (b) (simulation d'un changement d'illumination). (e) Quelques « lignes de niveaux » de l'Arbre des Formes Multivarié (AdFM). Les trois images (b), (c), (d) ont le même arbre des formes dont les lignes de niveaux sont montrées sur (e).

plus clair ou plus foncé. C'est pourquoi les opérateurs auto-duaux sont particulièrement bien adaptés pour le traitement des images où l'organisation du contenu n'est pas connu a priori. Alors que des opérateurs morphologiques tentent d'être auto-duaux (e.g. les filtres séquentiels alternés) en combinant des filtres extensifs et non-extensifs, certains dépendent en fait de l'ordre d'application (i.e. de quel filtre est appliqué en premier). Les filtres auto-duaux ont la capacité de traiter réellement de manière symétrique les objets foncés et clairs au même titre [53, 117] (voir la fig. 1c).

Malgré ces puissantes propriétés, l'AdF est encore largement sous-exploité, même si quelques auteurs l'ont d'ores et déjà utilisé efficacement dans des applications de traitement d'images et de vision par ordinateur. Dans [43, 13, 139, 142], les auteurs utilisent une approche par optimisation d'énergie sur la hiérarchie de l'AdF pour la simplification et la segmentation d'images en sélectionnant les lignes de niveaux significatives. D'autres applications incluent la segmentation de vaisseaux sanguins [144], le réglage d'images [28] et la mise en correspondance de scènes en étendant les MSER aux « Maximally Stable Shapes » [28] et aux « Tree-Based Morse Regions » [143].

Alors que l'AdF est bien défini sur les images à niveaux de gris, cela se complique sur les données multivariées. En effet, comme la plupart des arbres morphologiques (e.g. le min- et max-tree), l'AdF repose sur une relation d'ordre sur les valeurs qui doit être totale.

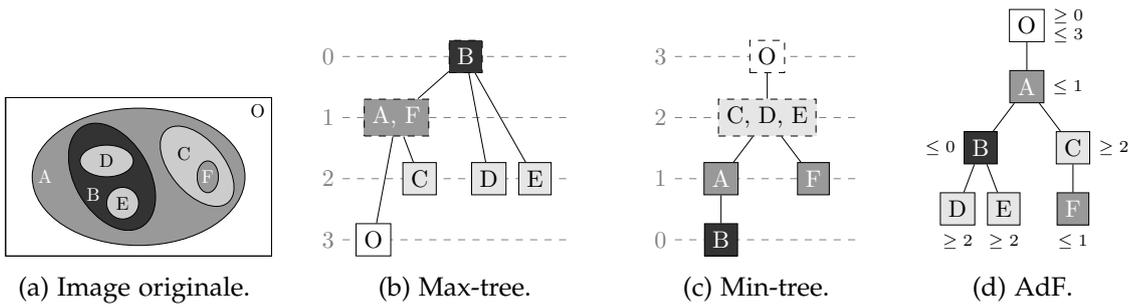


FIGURE 2: Représentations arborescentes basées sur le principe de décomposition.

Dans le cas contraire, les composantes connexes de coupes inférieures et supérieures se chevauchent et l'arbre d'inclusion est mal formé. Pour résoudre ce problème, on s'est le plus souvent concentré à définir un ordre total sur les données multivariées. Cependant, de notre point de vue, le concept le plus important dans les arbres morphologiques reste l'inclusion des formes. Par conséquent, on introduit une nouvelle approche qui s'affranchit de la nécessité d'un ordre total mais essaye de construire un ensemble de formes qui ne se chevauchent pas à partir d'un ensemble de formes quelconque en se basant uniquement sur la relation d'inclusion.

2 L'ARBRE DES FORMES ET LES PROBLÈMES LIÉS À LA COULEUR

2.1 L'arbre des formes : définition et propriétés

Soit $u : \Omega \rightarrow E$, une image définie sur un domaine Ω et prenant ses valeurs dans E muni d'une relation d'ordre \leq . Les lignes de niveaux de u sont la collection des ensembles de points $\{x \mid u(x) = \lambda, \lambda \in E\}$. En utilisant la représentation de $[\mathbf{1}]$, on peut s'assurer que ces ensembles forment des courbes fermées lorsque \leq est un ordre total. En fait, l'ordre doit être total dû à la définition des lignes de niveaux en termes de contours de coupes inférieures ou supérieures. Soit $[u \leq \lambda]$ (resp. $[u \geq \lambda]$) une coupe inférieure (resp. supérieure) de u définie par $[u \leq \lambda] = \{x, u(x) \leq \lambda\}$. On note $\mathcal{CC}(X), X \in \mathcal{P}(E)$ l'ensemble des composantes connexes de X . Si \leq est total, deux composantes connexes $X, Y \in \mathcal{CC}([u < \lambda])$ sont disjointes ou incluses. L'ensemble $\mathcal{CC}([u < \lambda])$ muni de la relation d'inclusion est un arbre appelé *min-tree* et son arbre dual, défini sur les coupes supérieures, est le *max-tree* (voir les figs. 2b and 2c). Soit l'opérateur de bouchage de trou \mathcal{H} , on appelle une *forme* un élément de $\mathcal{S} = \mathcal{H}([u \leq \lambda]) \cup \mathcal{H}([u \geq \lambda])$. Si \leq est total, là encore deux formes seront disjointes ou incluses et donc la couverture de (\mathcal{S}, \subseteq) forme un arbre : l'*Arbre des Formes (AdF)* (voir la fig. 2d). Dans le reste de ce papier, par abus de notation, on considère implicitement la couverture de (\mathcal{S}, \subseteq) lorsqu'on écrit (\mathcal{S}, \subseteq) . Notons aussi que l'AdF traduit à la fois l'inclusion des formes et l'inclusion des lignes de niveaux puisque celles-ci sont les frontières des formes. Enfin, sans perte de généralité,



(a) Image d'origine u



(b) Simplification sur une version en niveaux de gris de u (198 régions)



(c) Simplification avec un traitement marginal (123 + 141 + 136 régions)



(d) Simplification avec notre approche (158 régions)

FIGURE 3: Les problèmes liés à la simplification avec les approches « standard » pour traiter la couleur. (b) montre le problème de fuite lors que la luminance n'est suffisante pour obtenir complètement l'information géométrique. (c) montre le problème de fausses couleurs dues au traitement marginal. (d) Notre méthode récupère correctement le contenu principal de l'image sans introduire d'artefact visuel.

on considère $E = \mathbb{R}^n$ tout au long de ce papier et on notera u pour les images scalaires ($n = 1$) et \mathbf{u} pour les images multivariées.

2.2 Le problème des données multivariées : solutions standard et travaux connexes

Les définitions précédentes des lignes de niveaux (en termes d'ensemble d'iso-niveau ou en tant que contour de formes) sont toutes deux mal formées en présence d'un ordre partiel. En effet, les ensembles d'iso-niveau ne forment pas des courbes fermées et les formes issues des coupes inférieures et supérieures peuvent se chevaucher, *i.e.*, (\mathcal{S}, \subseteq) forme un graphe. Un moyen largement utilisé mais peu acceptable de résoudre le problème est de traiter l'image en niveaux de gris uniquement (e.g. la luminance). Cette approche reste plausible si on prétend que l'information géométrique est principalement contenue par la luminance [29]. Cependant, il n'est pas rare de rencontrer des images où les contours des objets n'existent qu'à travers la composante chromatique (notamment sur les images de document ou de synthèse). Ces exemples contredisent cette supposition

et montrent que la chrominance contient également l'information géométrique (voir la fig. 3b).

Une autre solution communément employée est le traitement de l'image canal par canal, puis la recombinaison des résultats individuels. Le traitement marginal est sujet au problème connu des fausses couleurs puisqu'il permet de créer des couleurs qui n'étaient pas présentes dans l'image d'origine. Les fausses couleurs peuvent être un problème ou pas (par exemple si les fausses couleurs sont suffisamment proches des couleurs d'origines pour ne pas être distinguées), mais dans le cas de la simplification, il produit des artéfacts de couleurs indésirables comme montrés sur la fig. 3c. Aussi, le traitement marginal produit au final plusieurs arbres (un pour chaque canal de l'image) alors que nous souhaitons obtenir une structure *unique* de l'image. Dans [2], nous avons proposé des idées préliminaires sur la manière d'obtenir un arbre unique à partir de plusieurs arbres.

Puisque le problème des formes qui se chevauchent est dû à l'ordre partiel des couleurs, des auteurs ont tenté d'imposer arbitrairement un ordre ou un pré-ordre *total*. Ils diffèrent en le fait qu'un nœud de l'arbre puisse être associé à une ou plusieurs valeurs. La manière d'ordonner un espace multivarié a été largement étudiée pour étendre les opérateurs morphologiques. [14] classe les méthodes en quatre groupes : les ordres marginaux (M-ordering), les ordres conditionnels (C-ordering), les ordres partiels (P-ordering) et les ordres réduits (R-ordering). Alors que la première classe mène à un ordre partiel, les trois autres donnent un ordre ou un préordre total. Les ordres conditionnels visent à organiser les vecteurs en donnant des priorités à certaines (ou toutes) composantes du vecteur. L'ordre lexicographique, bien connu, appartient à cette classe et est la méthode la plus commune pour étendre l'AdF aux couleurs [36]. Les ordres réduits visent à projeter des données vectorielles sur un espace réel à travers une fonction de rang. Les fonctions de rang bien connues sont par exemple la norme l_1 , la luminance dans un espace couleur donné, ou la distance à un ensemble de couleurs de référence. Elles ont été appliquées dans [123, 104, 89] pour la compression d'image et pour la détection d'objets astronomiques en utilisant les min ou max-trees mais l'idée est transposable pour l'AdF [3]. Des stratégies plus avancées ont été conçues pour donner naissance à un ordre total plus « sensé » où celui-ci dépend du contenu de l'image. VELASCO-FORERO et ANGULO [127, 126] utilisent l'apprentissage automatique pour obtenir un P-ordering basé sur le partitionnement de l'espace, puis établissent une distance entre ces regroupements. Dans [74], l'apprentissage de variétés est utilisé pour inférer une fonction de rang sur les couleurs et dans [72], un ordre total est calculé localement sur une fenêtre spatiale glissante. Une liste non-exhaustive pour inférer un ordre total sur les données multivariées peut être trouvée dans [9].

Une autre approche introduite par PASSAT et NAEGEL [102] utilise directement l'ordre partiel des valeurs et manipule la structure sous-jacente qui est un graphe. Le graphe des composantes est encore à l'état de développement mais a montré des résultats prometteurs dans le cadre de filtrage [90]. Néanmoins, le graphe des composantes fait face à un problème de complexité algorithmique qui oblige les auteurs à effectuer le

filtrage localement. De ce fait, le graphe des composantes est à l'heure actuelle non adapté si l'on veut une représentation de l'image entière.

Dans [2], nous avons introduit une nouvelle approche où, au lieu d'essayer d'imposer un ordre total sur les valeurs, nous calculons marginalement les AdFs et les fusionnons en un arbre unique. Le procédure de fusion ne dépend pas d'un ordre total sur les valeurs mais sur des propriétés calculées dans l'espace des formes. Néanmoins, la stratégie de fusion proposée dans ce papier souffre d'un manque de « cohérence » car elle fusionne ensemble des formes sans liaisons apparentes. Dans [3], inspiré par le travail de PASSAT et NAEGEL [102], nous avons proposé le graphe des formes qui fusionne les AdFs marginaux en une seule structure et de façon efficace. Nous avons montré que cette structure avait un fort potentiel comparée aux approches standard qui imposent un ordre total. Cependant, la méthode construit un graphe, ce qui est une limitation puisque nous ne pouvons plus utiliser les outils fournis par l'état de l'art sur les arbres de composantes (filtrage, détection d'objets, méthodes de segmentation...) Le travail présenté ici peut être vu comme la poursuite des idées introduites dans [2] et [3] puisque le Graphe des Formes (GdF) est utilisé comme une représentation intermédiaire pour extraire un arbre unique depuis les formes issues des AdFs marginaux.

3 L'ARBRE DES FORMES COULEUR

3.1 Contraintes et propriétés de l'arbre des formes couleur

Commençons d'abord par relâcher la définition de *forme*. Une *forme* X est une composante connexe de Ω sans trous (*i.e.* telle que $\mathcal{H}(X) = X$). Soit une famille d'ensemble de formes $\mathcal{M} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_n\}$ où chaque élément $(\mathcal{S}_i, \subseteq)$ forme un arbre. On note $\mathcal{S} = \bigcup \mathcal{S}_i$ l'ensemble initial de formes. Notons que (\mathcal{S}, \subseteq) ne forme généralement pas un arbre mais un graphe puisque des formes peuvent se chevaucher. On cherche à définir un nouvel ensemble de formes \mathcal{S}' tel que deux formes soient disjointes ou incluses. On ne se limite pas aux ensembles $\mathcal{S}' \subseteq \mathcal{S}$. En d'autres termes, on autorise la méthode à créer de nouvelles formes qui n'étaient pas présentes dans l'ensemble de formes d'origine. On note $T(\mathbf{u}) : \Omega^{\mathbb{R}^n} \rightarrow (\mathcal{P}(\mathcal{P}(\Omega)), \subseteq)$, le processus qui construit un arbre des formes $(\mathcal{S}', \subseteq)$ à partir d'une image $\mathbf{u} \in \Omega^{\mathbb{R}^n}$.

Plus formellement, on souhaite que la méthode T produise $(\mathcal{S}', \subseteq)$ avec les propriétés suivantes :

- (R1) La couverture du domaine $(\bigcup_{X \in \mathcal{S}'} X) = \Omega$ (tout point appartient à au moins une forme)
- (R2) Une structure arborescente $\forall X, Y \in \mathcal{S}'$, soit $X \cap Y = \emptyset$ ou $X \subseteq Y$ ou $Y \subseteq X$ (deux formes sont disjointes ou incluses)

Et on impose également les contraintes suivantes :

- (R₃) *L'équivalence avec l'AdF scalaire.* Si $\mathcal{M} = \{\mathcal{S}_1\}$ alors $\mathcal{S}' = \mathcal{S}_1$ (si l'image est scalaire, alors la méthode doit produire le même arbre que l'AdF calculé sur l'image d'origine).
- (R₄) Pour chaque forme $X \in \mathcal{S}$ telle que pour chaque autre forme $Y \in \mathcal{S}$, $X \cap Y = \emptyset$ ou $X \subseteq Y$ ou $Y \subseteq X$ alors, $X \in \mathcal{S}'$ (une forme qui ne chevauche aucune autre forme doit exister dans l'arbre final).
- (R₅) *L'invariance par changement et inversion de contraste* Une transformation ψ est dite morphologique (i.e. invariante par changement de contraste) si pour une fonction croissante $g : \mathbb{R} \rightarrow \mathbb{R}$, $g(\psi(u)) = \psi(g(u))$. De plus, la transformation est dite auto-duale si elle est invariante par complémentation $\mathbb{C}(u) = -u$ i.e. $\mathbb{C}(\psi(u)) = \psi(\mathbb{C}(u))$. Soit une fonction monotone $F : \mathbb{R} \rightarrow \mathbb{R}$ telle que $\forall x, y \in \mathbb{R}$, $x < y \Rightarrow F(x) < F(y)$ ou $F(x) > F(y)$, ψ est à la fois auto-duale et invariante par changement de contraste si $F(\psi(u)) = \psi(F(u))$. L'AdF est un support pour certains opérateurs morphologiques auto-duaux et une représentation T est dite auto-duale et morphologique si $T(F(u)) = T(u)$. Pour l'extension au multivarié de cette propriété, notons \mathcal{F} l'ensemble de toutes les fonctions monotones, T est dit invariant par changement et inversion de contraste si pour $\mathbf{F} \in \mathcal{F}^n$, $\mathbf{F}(\mathbf{x}) = (F_1(x_1), F_2(x_2), \dots, F_n(x_n))$, alors $T(\mathbf{F}(\mathbf{u})) = T(\mathbf{u})$.

3.2 Description de la méthode

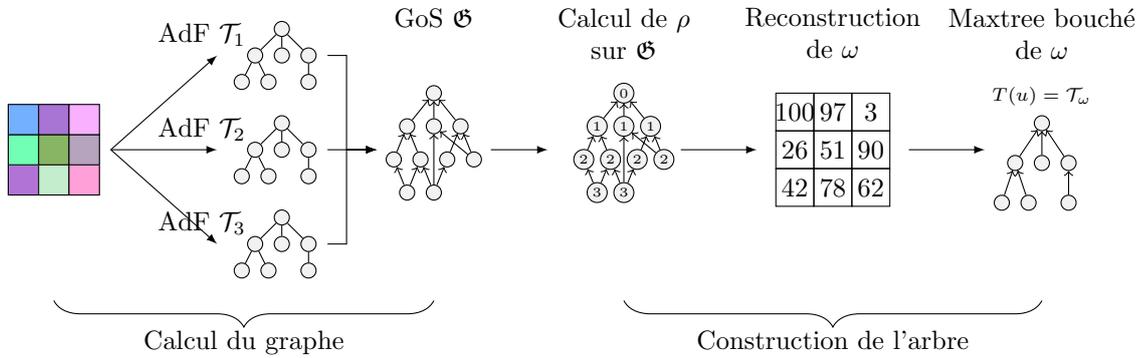


FIGURE 4: Les 5 étapes de la méthode proposée. (1) L'image d'entrée \mathbf{u} est décomposée en ses différentes composantes u_1, u_2, \dots, u_n , (2) l'AdF est calculé sur chaque composante, (3) les AdFs sont fusionnés en une unique structure : le GdF, (4) une image scalaire ω est calculée en utilisant la profondeur des nœuds auxquels les points appartiennent. (5) un arbre est calculé à partir de ω .

La méthode que nous proposons est un processus en 5 étapes (voir la fig. 4). Tout d'abord, \mathbf{u} est décomposé en ses composantes individuelles u_1, u_2, \dots, u_n sur lesquelles on calcule les AdFs $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_n$ associés aux ensembles de formes $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_n$. On note

$\mathcal{S} = \bigcup \mathcal{S}_i$, on appelle le GdF \mathfrak{G} la couverture de (\mathcal{S}, \subseteq) , i.e. c'est le graphe d'inclusion des toutes les formes issues de coupes marginales. Soit $\rho : \mathcal{P}(\Omega) \rightarrow \mathbb{N}$ un attribut de forme algébrique décroissant i.e. $\forall A, B \in \mathcal{S}, A \subset B \Rightarrow \rho(A) > \rho(B)$

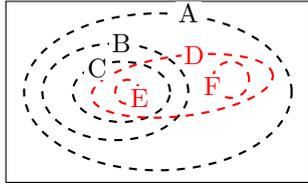
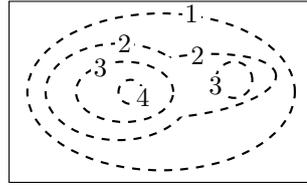
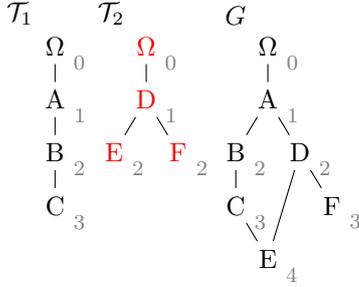
Un exemple d'un tel opérateur est l'attribut de *profondeur*. La profondeur d'une forme A de \mathfrak{G} est la longueur du chemin le plus long de la racine vers A . ω Soit $\omega : \Omega \rightarrow \mathbb{R}$ défini par :

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

ω est une image scalaire qui associe en chaque point de x , la profondeur de la forme la plus profonde qui le contient (voir les figs. 5b and 5c) Soit $\mathbf{C} = \{CC([\omega \geq h]), h \in \mathbb{R}\}$. (\mathbf{C}, \subseteq) est en fait le max-tree de ω . Ce dernier pouvant donner lieu à des composantes avec des trous, on considère donc $\mathcal{S}' = \mathcal{H}(\mathbf{C})$ et $(\mathcal{S}', \subseteq)$ pour l'AdF final \mathcal{T}_ω (voir la fig. 5d).

Intuition. On explique maintenant l'intuition de cette démarche. D'abord, nous commençons par calculer les AdFs marginaux de u qui nous donnent un ensemble de formes initial. Les différents arbres (ensemble) fournissent une représentation de l'image d'origine et u peut être reconstruite marginalement depuis ces arbres. Cependant, la manipulation simultanée des différents arbres est délicate et il manque une information capitale : comment les formes d'un arbre donné sont liées (au sens de l'inclusion) aux formes des autres arbres. Le graphe \mathfrak{G} n'est rien de plus que ces arbres fusionnés en une unique structure et ajoute la relation d'inclusion qui manquait précédemment. Par conséquence, \mathfrak{G} est plus « riche » que $\{\mathcal{T}_1, \dots, \mathcal{T}_n\}$ parce que la transformation de $\{\mathcal{T}_1, \dots, \mathcal{T}_n\}$ à \mathfrak{G} est réversible ; \mathfrak{G} est une représentation complète de u (u peut être reconstruite depuis \mathfrak{G}). De plus, \mathfrak{G} est invariant par changement et inversion de contraste marginal de u car $\{\mathcal{T}_1, \dots, \mathcal{T}_n\}$ le sont.

La seconde partie de la méthode tente d'extraire un arbre depuis \mathfrak{G} vérifiant les contraintes données dans la section 3.1. Le problème majeur est d'obtenir un nouvel ensemble de formes depuis \mathfrak{G} qui ne se chevauchent pas. Notons d'abord que pour n'importe quel attribut décroissant ρ , on a (\mathcal{S}, \subset) qui est isomorphe à $(\mathcal{S}, \mathcal{R})$ où $A \mathcal{R} B \Leftrightarrow \rho(A) > \rho(B)$ and $A \cap B \neq \emptyset$. En termes plus simples, la relation d'inclusion entre deux formes que nous voulons conserver peut s'exprimer en terme de valeurs d'attributs dans \mathbb{R} . Supposons maintenant que (\mathcal{S}, \subset) est un arbre et considérons l'image $\omega(x) = \max_{X \in \mathcal{S}, x \in X} \rho(x)$, on a ainsi $\mathbf{C} = \{CC([\omega \geq h]), h \in \mathbb{R}\} = \mathcal{S}$. Dit autrement, le max-tree de l'image ω reconstruite à partir de l'attribut ρ évalué sur un arbre \mathcal{T} donne exactement le même arbre \mathcal{T} (pré-requis R3). Plus généralement, si une forme A ne chevauche aucune autre forme de \mathcal{S} alors elle appartiendra à $CC([\omega \geq h])$ (pré-requis R4). Dans la section qui suit, nous allons maintenant justifier le choix de la profondeur comme attribut pour ρ .


 (a) Image originale u (2-canaux) et ses formes.

 (c) L'image ω construite depuis \mathcal{O}

 (b) Les ADFs marginaux \mathcal{T}_1 , \mathcal{T}_2 et le GdF. La profondeur apparaît en gris clair près des nœuds.

 (d) Le max-tree \mathcal{T}_f of ω

FIGURE 5: Schéma illustrant le fonctionnement de la méthode

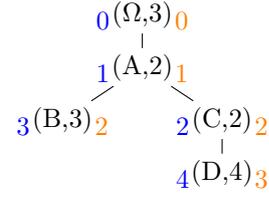
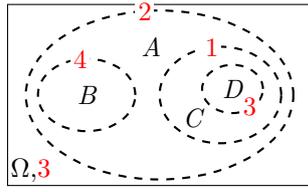
3.3 Le calcul de la carte d'attribut ω

La 4^e étape de la méthode consiste à choisir un attribut à évaluer sur le GdF \mathcal{O} . C'est une étape critique qui décide de quelle forme va être fusionnée ou supprimée. Nous expliquons maintenant pourquoi nous utilisons la profondeur comme attribut pour fusionner les formes.

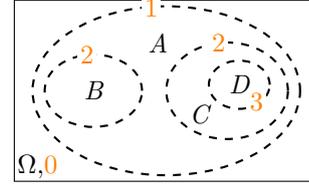
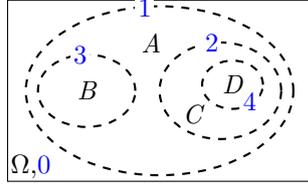
Considérons la distance entre deux points (p, p') dans Ω définie par

$$d_{\text{TV}}(p, p') = \min_{C_{(p,p')}} \int_0^1 |\nabla u(C(t)) \cdot \dot{C}(t)| dt \quad (2)$$

où $C(t)$ est un chemin dans Ω de p à p' . L'eq. (2) est en fait la variation totale (TV) du chemin minimisant la TV parmi tous les chemins de p vers p' . Cette mesure a par exemple été utilisée par DUBROVINA et al. [44] à des fins de segmentation où l'AdF est utilisé comme support pour le calcul efficace de la distance. Soit $\omega_{\text{TV}}(x) = d_{\text{TV}}(\partial\Omega, x)$, la carte de distance de variation totale depuis la bordure de l'image. Celle-ci peut être calculée par simple calcul d'attribut sur l'arbre : il suffit de sommer les variations absolues le long du chemin de la racine vers un nœud. Ainsi, au lieu de considérer l'arbre \mathcal{T} des lignes de niveaux de u , on peut considérer à la place le max-tree \mathcal{T}_ω des lignes équidistantes de TV. Les deux arbres sont équivalents en niveaux de gris. Le problème avec la variation totale est qu'elle n'est pas invariante au changement de contraste de u . Une distance invariante



(a) Image originale u , ses formes et ses lignes (b) L'AdF de u et l'évaluation de ρ_{TV} (bleu) et de ρ_{CV} (orange).



(c) Les lignes de niveaux de ω_{TV}

(d) Les lignes de niveaux de ω_{CV}

FIGURE 6: Équivalence entre les lignes de niveaux de l'image en niveaux de gris u et des lignes de niveaux des cartes de distances ω_{TV} et ω_{CV} .

par changement de contraste de u serait simplement de compter le nombre de variation, *i.e.*, le nombre minimum de ligne de niveaux à traverser pour atteindre p :

$$d_{CV}(p, p') = \min_{C(p, p')} \int_0^1 \mathbb{1}\{\nabla u(C(t)) \cdot \dot{C}(t)\} dt \quad (3)$$

Algorithmiquement parlant, construire ω_{CV} consiste à calculer l'attribut de profondeur $\rho_{CV}(A) = |\{X \in \mathcal{S} \mid A \subset X\}|$ et reconstruire $\omega_{CV}(x) = \max_{X \in \mathcal{S}, x \in X} \rho_{CV}(X)$ (voir la fig. 6).

Basé sur l'équivalence entre les lignes de niveaux et les lignes « d'équidistances » en niveaux de gris, on peut étendre cette idée en couleur. Comme dans l'eq. (3), on souhaite compter le nombre minimum de lignes de niveaux marginales à traverser. Plus formellement :

$$\rho(A) = \max_{\phi \in [\Omega \rightsquigarrow A]} |\phi| \text{ and } \omega_{CV}(x) = \max_{X \in \mathcal{S}, x \in X} \rho(X)$$

où $[\Omega \rightsquigarrow A]$ désigne l'ensemble de tous les chemins de la racine vers A dans \mathfrak{G} . On compte ainsi le nombre de formes *qui s'incluent* à traverser pour atteindre la forme la plus profonde contenant x . ρ peut être calculé efficacement depuis \mathfrak{G} en utilisant un algorithme classique de plus court chemin.

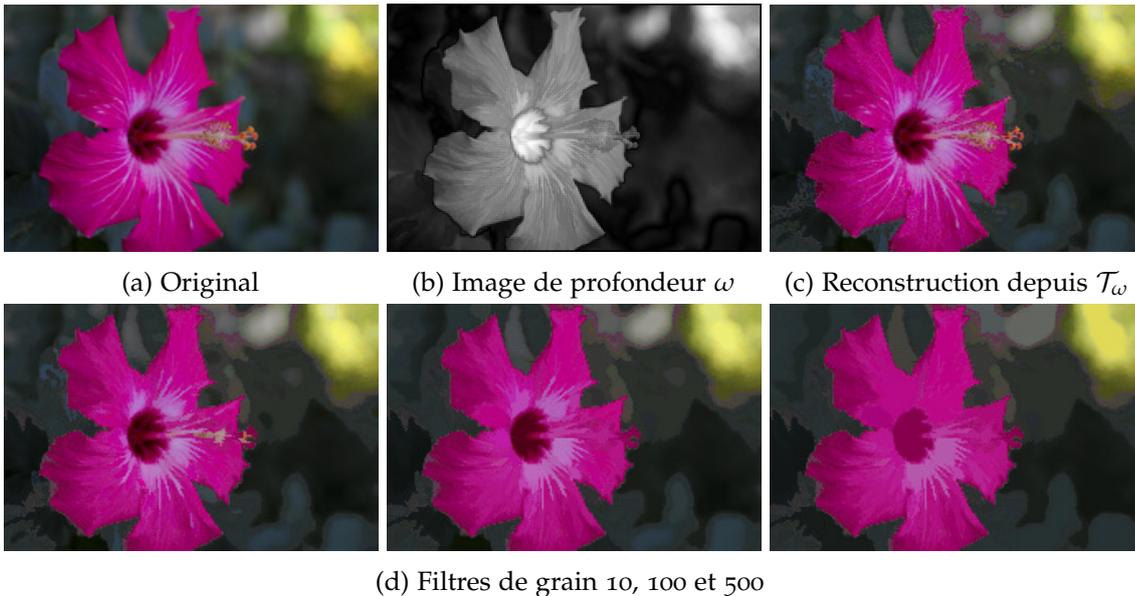
4 APPLICATIONS

4.1 Filtres de grains

Le filtre de grain [29, 108] est un opérateur qui supprime les régions de l'image qui sont des extrema locaux dont l'aire est inférieure à un certain seuil. En ce sens, il est lié



FIGURE 7: Schéma du processus de filtrage de grain.



(d) Filtrage de grains 10, 100 et 500

FIGURE 8: Filtrage de grains

aux *extreme filters* mais assurent un traitement complètement symétrique des minima et maxima, *i.e.*, il est *auto-dual*. Avec l'AdF, un filtre de grain est un simple élagage de l'arbre, supprimant tous les nœuds qui ne passent un certain critère de taille. Le processus est illustré schématiquement dans la fig. 7. On assigne à chaque nœud la taille de la composante qu'il représente et on supprime les nœuds de taille inférieure à 4, *i.e.*, tous les nœuds en dessous de la courbe rouge. Les pixels qu'ils contiennent sont ensuite rattachés au plus proche ancêtre encore « vivants ». Malgré sa simplicité, nous montrons la puissance de ce filtre à travers de la simplification et l'extraction de canevas de document.

Simplification d'images

Les filtres de grain permettent de révéler la « validité » de l'arbre dans le sens où une petite taille de grain doit supprimer ce que nous percevons comme bruit ou détail, alors qu'une grande taille de filtrage doit montrer les objets principaux et la structure de l'image. Dans la fig. 8, nous montrons la carte d'inclusion ω calculée par notre méthode

et l'image reconstruite depuis le max-tree \mathcal{T}_ω . La reconstruction consiste en le calcul, pour chaque nœud, de la couleur moyenne de ses pixels, puis en l'affectation de cette valeur aux pixels. Puisque \mathcal{T}_ω n'est pas une représentation inversible de \mathbf{u} , cette dernière ne peut pas être reconstruite depuis \mathcal{T}_ω . Néanmoins, la reconstruction est proche de l'image d'origine. Dans la fig. 8d, nous avons appliqué des filtres de grain de tailles croissantes qui suppriment les détails de manière « sensée » et fournit une reconstruction avec peu d'artefacts de couleur, ce qui valide l'organisation structurelle de notre arbre.

Extraction du canevas de document



FIGURE 9: Filtrages pour la extraction de canevas de document. Première ligne : images d'origine ; seconde ligne : résultats des filtres de grain.

Nous utilisons un filtre de grain pour extraire les boîtes de texte et les parties graphiques du document. En effet, les parties textuelles sont composées de lettres qui sont

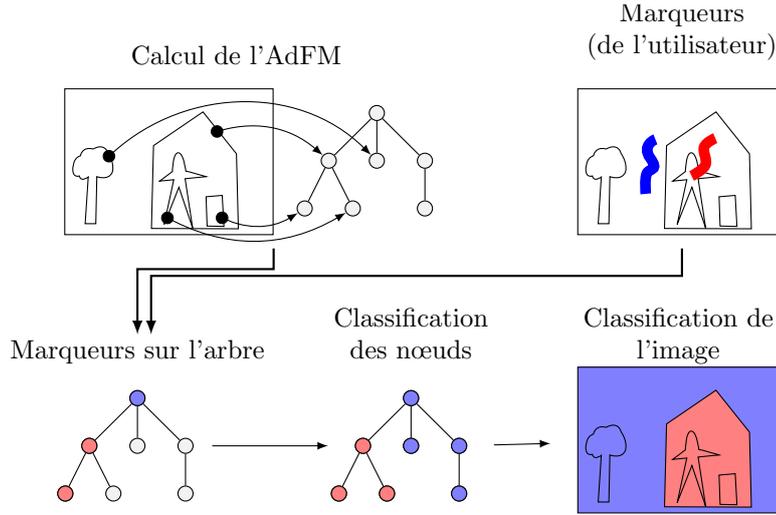


FIGURE 10: Schéma de la méthode de segmentation interactive.

de petites composantes si l'AdFM est bien-formé. Au contraire, les boîtes de texte et le contenu graphique sont de larges composantes qui persistent après le filtrage. Figure 9 montre l'extraction de contenu non-textuel où l'auto-dualité est fondamentale puisque le texte peut être au dessus d'un fond plus clair ou plus foncé. Comme chacun peut le voir, les images filtrées ne contiennent que les graphiques et les boîtes de texte alors que les lettres sont dans le résidu.

4.2 Segmentation interactive

Le problème de segmentation interactive peut être résumée comme suit. Étant donné deux ensembles de points F et B dans $\mathcal{P}(\Omega)$ représentant les marques rentrées par l'utilisateur, on souhaite annoter chaque point de Ω avec une des deux classes.

On considère la mesure entre deux points (p, p') de Ω :

$$d_{TV}(p, p') = \min_{C_{pp'}} \int_0^1 |\nabla u(C_{pp'}(t)) \cdot \dot{C}_{pp'}(t)| \cdot dt, \quad (4)$$

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 [107, 12]. Néanmoins, dans [44], 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'AdFM 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 fig. 10) :

1. Calculer l'AdFM $T(\mathbf{u})$ de l'image \mathbf{u} ,
2. Valuer les arrêtes de $T(\mathbf{u})$ par la distance inter-nœuds (distance euclidienne entre les couleurs moyennes).
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} or \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).

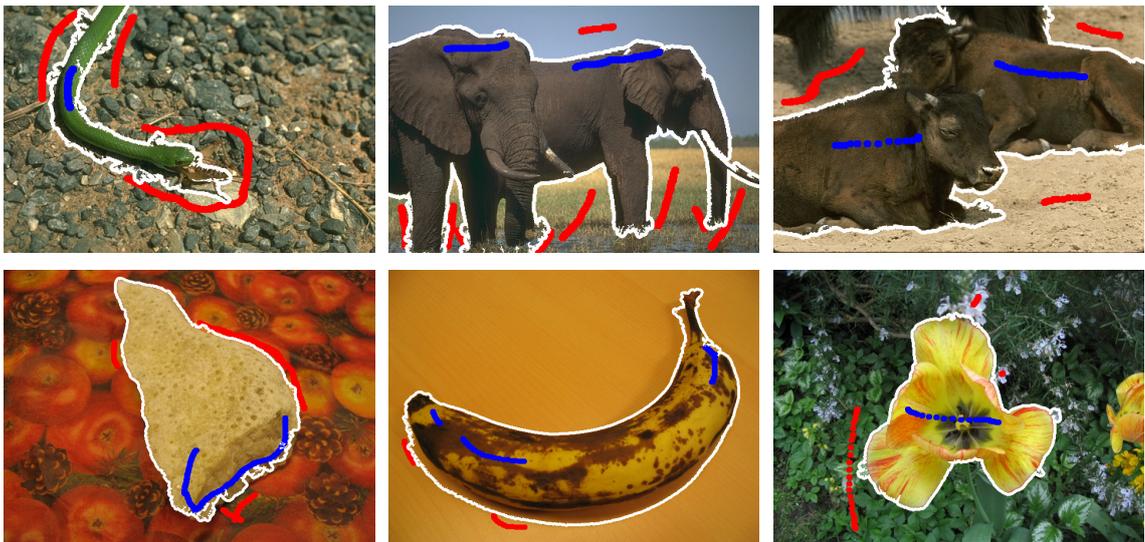


FIGURE 11: Segmentation interactive avec notre méthode. Les marques rouges et bleues définissent respectivement le fond \mathcal{B} et l'objet \mathcal{F} . La ligne blanche est la frontière \mathcal{F}/\mathcal{B} calculée par notre méthode

Des exemples de résultat de notre méthode sont présentés dans la fig. 11.

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.

Un second avantage de l'approche est qu'elle requiert aucune modélisation statistique de la région. Elle utilise uniquement 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.

4.3 Détection de documents dans les vidéos

Dans le cadre de la compétition ICDAR *Smartphone Document Capture and OCR (SmartDoc-2015)* [19], on souhaite détecter automatiquement les documents dans des vidéos acquises par téléphones intelligents. La base de données couvre différents types de document (textuel et/ou avec un contenu graphique) et différents problèmes d'analyse de scène (changement d'illumination, bruit de déplacement, changement de perspective, etc.). Une nouvelle fois, la méthode que nous proposons s'appuie l'AdFM de l'image. Dit simplement, on cherche à identifier des nœuds dans l'arbre qui remplissent des critères de forme de document. Ceux-ci sont exprimés en tant qu'énergie à deux termes :

1. À quel point la bordure de la forme correspond à un quadrilatère. Pour chaque forme A , on calcule le meilleur quadrilatère Q_A correspondant à la forme et on mesure le ratio : $E_1(A) = |A|/|Q_A|$.
2. Quelle est la quantité de bruit dans l'objet. On s'attend à un document avec du texte qui se traduit par du bruit au niveau du document. Soit $\mathcal{L}_A = \{X \in \mathcal{S} \mid X \subset A \text{ et } X \text{ est une feuille}\}$ l'ensemble des feuilles du sous-arbre enraciné en A alors :

$$E_2(A) = \frac{\sum_{X \in \mathcal{L}_A} (d(X) - d(A))}{|\mathcal{L}_A|}$$

où $d(X)$ est la profondeur du nœud X

On cherche ensuite la forme qui minimise l'énergie $E_1(X) + E_2(X)$. Notons, que pour une meilleure précision de E_2 , nous commençons par pré-traiter l'image avec un filtre de petit grain pour réduire l'effet du bruit naturel d'acquisition de l'image. Des exemples de résultats de notre méthode sont présentés dans la fig. 12.

Cette méthode (légèrement modifiée pour permettre le *tracking* de document entre les images vidéo) a obtenu la première place du concours. L'évaluation était basée sur le score Jaccard qui mesure la similarité entre l'ensemble des points attendus dans la vérité terrain et ceux retournés par la méthode. Celle-ci a obtenu un score moyen de 0.9716, variant entre 0.9710 et 0.9721 sur la base entière [19], ce qui montre la robustesse de l'approche proposée.

5 CONCLUSION

Nous avons présenté une méthode qui étend l'AdF sur les images multivariées. Contrairement aux approches standards, notre AdFM s'affranchit du choix d'un ordre total

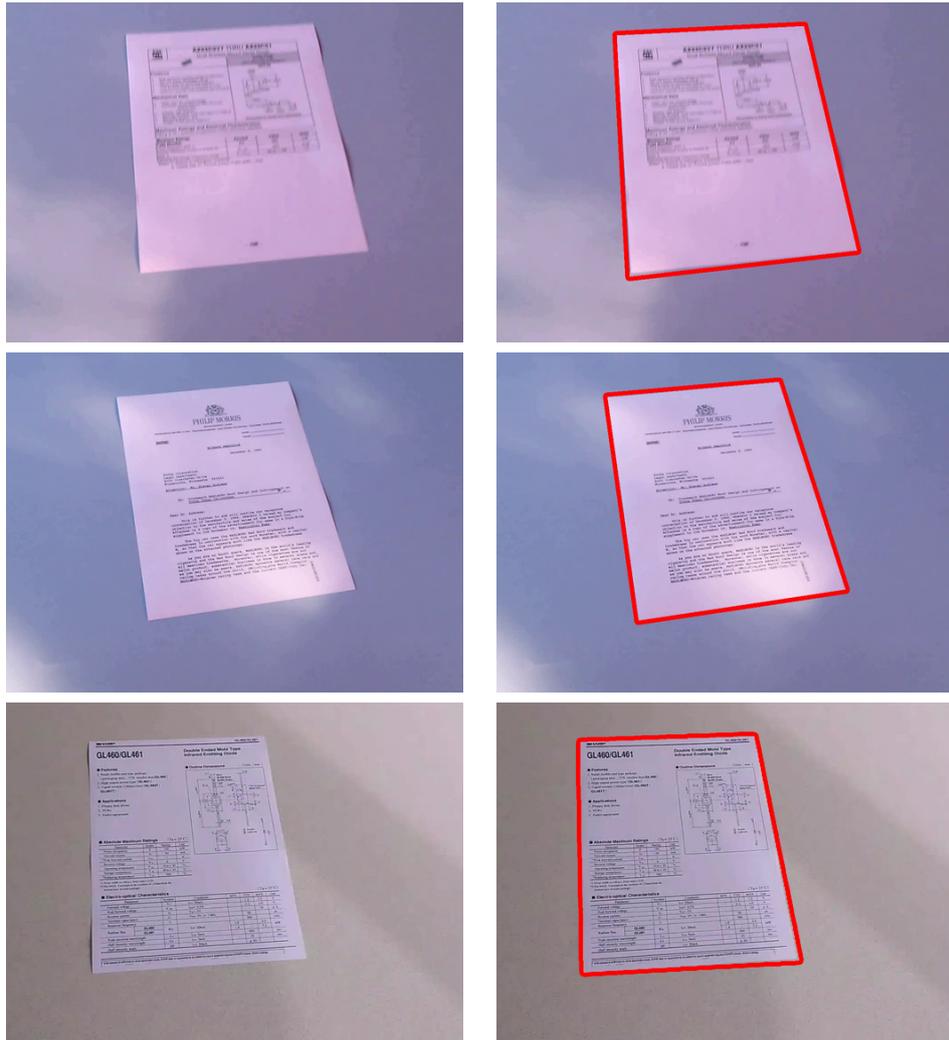


FIGURE 12: Compétition ICDAR sur la détection de document. Ces images montrent la robustesse de notre méthode au flou et aux effets spéculaires de lumière qui déplacent des frontières des objets. Des vidéos sont disponibles en tant que matériel supplémentaire [24].

sur ces données et repose uniquement sur la relation d’inclusion entre les formes. Elle produit ainsi un arbre qui est invariant par changement et inversion de contraste marginal de l’image. Nous avons tenté de mettre en évidence en quoi ces propriétés sont fondamentales en traitement d’images et en vision par ordinateur. Pour y parvenir, nous avons illustré AdFM à travers des applications de simplification, de segmentation et de détection d’objet, montrant des résultats déjà prometteurs et mettant ainsi en avant la versatilité et le potentiel de notre approche.