SORBONNE UNIVERSITÉ

DOCTORAL THESIS

A new minimum barrier distance for multivariate images with applications to salient object detection, shortest path finding, and segmentation

---

Author:
Minh On Vu Ngoc

Supervisor:
Thierry Géraud
Jonathan Fabrizio

Reviewer:
Nicole Vincent, LIPADE, Univ. Paris Descartes
Jean-Christophe Burie, L3I, Univ. La Rochelle

Examiner:
Benoît Naegel, ICube, Univ. de Strasbourg
Béatriz Marcotegui, CMM, Mines ParisTech

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

Laboratoire de Recherche et Développement de l'EPITA
École Doctorale Informatique, Télécommunications et Électronique

June 5, 2020

"If I have seen further than others, it is by standing upon the shoulders of giants."

Isaac Newton

Acknowledgements

It was a great pleasure for me to do my PhD program at LRDE, EPITA. The past three years were full of new knowledge and valuable work experience. I would like to thank all who in one way or another contributed in the completion of this thesis.

I would like to sincerely thank Nicole Vincent and Jean-Christophe Burie, who accepted to review this thesis. I would also like to thank Benoît Naegel and Béatriz Marcotegui, who agreed to be members of the jury.

I am deeply grateful to Theo. He is very patient, knowledgeable, smart and funny. He has been very supportive and given me the freedom to pursue my project without objection. Thank you for having been able to encourage me even when conditions were not the most favorable.

I would like to express my deeply-felt thanks to my co-advisor, Jonathan, for his support, his patience and for pushing me toward my goal. He also gave me a lot of useful advice, not only in work but also in real life. I would also like to thank Nicolas for his time, feedback and interest in my work. He helped me a lot when I wrote my papers and also my thesis.

I warmly thank my friends, Duy, Ludovic, Jim, Zhou, Julie, Michael and all of my colleagues from LRDE, for their countless help, thoughtful suggestions, friendship, and support. I would like to thank Daniela for her support and assistance for helping me going through all the administrative procedures. I also want to thank Clément for his time and his assistance.

My special thanks go to all my friends, Diem, Linh, Nguyen, Hoa, Trinh, Phi, Toan, Tuyen,... for supporting me, encouraging me and always staying by my side whenever I need.

I would like to dedicate this thesis to my beloved family. They always believe in me, support me and love me unconditionally. Without them, I can not become who I am now.

Thank you all!

Abstract

Hierarchical image representations are widely used in image processing to model the content of an image in the multi-scale structure. A well-known hierarchical representation is the tree of shapes (ToS) which encodes the inclusion relationship between connected components from different thresholded levels. This kind of tree is self-dual, contrast-change invariant and popular in computer vision community. Typically, in our work, we use this representation to compute the new distance which belongs to the mathematical morphology domain.

Distance transforms and the saliency maps they induce are generally used in image processing, computer vision, and pattern recognition. One of the most commonly used distance transforms is the geodesic one. Unfortunately, this distance does not always achieve satisfying results on noisy or blurred images. Recently, a new pseudo-distance, called the minimum barrier distance (MBD), more robust to pixel fluctuation, has been introduced. Some years after, Géraud et al. have proposed a good and fast-to-compute approximation of this distance: the Dahu pseudo-distance. Since this distance was initially developed for grayscale images, we propose here an extension of this transform to multivariate images; we call it vectorial Dahu pseudo-distance. This new distance is easily and efficiently computed thanks to the multivariate tree of shapes (MToS). We propose an efficient way to compute this distance and its deduced saliency map in this thesis. We also investigate the properties of this distance in dealing with noise and blur in the image. This distance has been proved to be robust for pixel invariant.

To validate this new distance, we provide benchmarks demonstrating how the vectorial Dahu pseudo-distance is more robust and competitive compared to other MB-based distances. This distance is promising for salient object detection, shortest path finding, and object segmentation. Moreover, we apply this distance to detect the document in videos. Our method is a region-based approach which relies on visual saliency deduced from the Dahu pseudo-distance. We show that the performance of our method is competitive with state-of-the-art methods on the ICDAR Smartdoc 2015 Competition dataset.

Keywords: Tree of shapes, mathematical morphology, hierarchical representation, multivariate images, Dahu pseudo-distance, minimum barrier distance,visual saliency, Document detection, image segmentation.

Résumé

Les représentations hiérarchiques d'images sont largement utilisées dans le traitement d'images pour modéliser le contenu d'une image par un arbre. Une hiérarchie bien connue est l'arbre des formes (AdF) qui encode la relation d'inclusion entre les composants connectés à partir de différents niveaux de seuil. Ce genre d'arbre est auto-duale et invariant de changement de contraste, ce qu'il est utilisé dans de nombreuses applications de vision par ordinateur. En raison de ses propriétés, dans cette thèse, nous utilisons cette représentation pour calculer la nouvelle distance qui appartient au domaine de la morphologie mathématique.

Les transformations de distance et les cartes de saillance qu'elles induisent sont généralement utilisées dans le traitement d'images, la vision par ordinateur et la reconnaissance de formes. L'une des transformations de distance les plus couramment utilisées est celle géodésique. Malheureusement, cette distance n'obtient pas toujours des résultats satisfaisants sur des images bruyantes ou floues. Récemment, une nouvelle pseudo-distance, appelée distance de barrière minimale (MBD), plus robuste aux variations de pixels, a été introduite. Quelques années plus tard, Géraud et al. ont proposé une bonne approximation rapide de cette distance : la pseudo-distance de Dahu. Puisque cette distance a été initialement développée pour les images en niveaux de gris, nous proposons ici une extension de cette transformation aux images multivariées ; nous l'appelons vectorielle Dahu pseudo-distance. Cette nouvelle distance est facilement et efficacement calculée grâce à à l'arbre multivarié des formes (AdFM). Nous vous proposons une méthode de calcul efficace cette distance et sa carte de saillants déduits dans cette thèse. Nous enquêtons également sur le propriétés de cette distance dans le traitement du bruit et du flou dans l'image. Cette distance s'est avéré robuste pour les pixels invariants.

Pour valider cette nouvelle distance, nous fournissons des repères démontrant à quel point la pseudo-distance vectorielle de Dahu est plus robuste et compétitive par rapport aux autres distances basées sur le MB. Cette distance est prometteuse pour la détection des objets saillants, la recherche du chemin le plus court et la segmentation des objets. De plus, nous appliquons cette distance pour détecter le document dans les vidéos. Notre méthode est une approche régionale qui s'appuie sur le saillance visuelle déduite de la pseudo-distance de Dahu. Nous montrons que la performance de notre méthode est compétitive par rapport aux méthodes de pointe de l'ensemble de données du concours Smartdoc 2015 ICDAR.

Mots-clés: Arbre de formes, morphologie mathématique, représentation hiérarchique, images multivariées, pseudo-distance de Dahu, distance de barrière minimale, saillance visuelle, document détection, segmentation de l'image.

Résumé long

Résumé

Les représentations hiérarchiques des images ont été largement utilisée de ces dernières années pour les tâches de segmentation et de filtrage. Elles peuvent être utilisées pour modéliser le contenu d'une image par un arbre. Nous nous concentrons à l'arbre des formes (AdF) qui appartiennent aux arbres basés sur la décomposition de seuil en raison de ses propriétés. Dans cette thèse, nous nous sommes interessés à la distance de barriere minimum. Cette distance s'est avérée robuste pour les images bruitée et floues. Malheureusement, son calcul est coûteux. Par conséquent, nous avons proposé un moyen efficace de la calculer grâce à l'AdF. Cette distance approximative est appelée la pseudo-distance du Dahu. Cette thèse est consacrée à l'étude des propriétés de la pseudo-distance du Dahu et à l'application de cette distance dans plusieurs applications, telles que la détection d'objets saillants, la recherche du plus court chemin, la segmentation des images et la détection d'objets. Nous avons également utilisé cette distance pour détecter des documents dans des vidéos capturées par des smartphones.

1 Introduction

En traitement d'images, vision par ordinateur ou reconnaissance de formes, les objets peuvent apparaître avec différentes tailles et différentes positions dans l'image. Ainsi, pour traiter différentes applications de la vision par ordinateur, il faut tenir compte de la représentation multi-échelle de l'image et de la façon dont les objets sont reliés aux autres. Cela donne lieu à une hiérarchie représentation de l'image, qui est un ensemble d'images connectées du niveau fin au niveau grossier, appelé représentation arborescente. S'appuyer sur les propriétés des arbres, on peut les classer en deux types : arbres de partitions hiérarchiques et représentation de la décomposition des seuils. Dans cette thèse, nous nous concentrons sur la deuxième classe d'arbres.

Cette représentation encode la relation d'inclusion spatiale entre les composantes connectées de différents niveaux de seuil. Min- et Max-tree [1, 2] et l'AdF [3] sont trois méthodes typiques de ce type de représentation. Toute découpe dans ces représentations génère une partition partielle de l'image. De plus, les feuilles de ces présentations correspondent à l'extrema local de l'image. L'AdF est auto-dual et invariant au changement de contraste, ce qui fait de lui une structure bien adaptée au traitements d'images.

De nombreux opérateurs basés sur l'arbre des formes sont généralement utilisés dans le domaine du traitement de l'image. Dans cette thèse, nous prenons en compte une autre approche, appelée transformation à distance, pour traiter de la représentation régionale. Cette transformée de distance est utilisée pour mesurer la dissimilarité entre les pixels de l'image, exprimant ainsi la relation entre l'objet et

l'arrière-plan à travers la définition de la détection des points saillants, ou la relation entre les régions dans le même objet. La fonction de distance a longtemps été étudiée dans la communauté de la morphologie mathématique. L'idée de base de la transformation de distance vient de l'image binaire pour trouver la distance minimale entre un ensemble de chemins entre deux pixels.

Dans cette thèse, nous nous concentrons sur les distances par trajet, où les images peuvent également être vues sous forme de graphes (les sommets sont les pixels de l'image et les arêtes sont induites par la relation de voisinage entre ces pixels). La distance la plus utilisée dans le traitement des images est la distance géodésique [4]. Cependant, cette distance n'est pas assez robuste pour traiter des images bruitées et floues. Dernièrement, une nouvelle pseudo-distance, appelée distance de barriere minimum (MBD) a été proposée dans [5].

La distance de barriere minimum est la valeur minimale de toutes les "intensités" de la barrière (notion définie plus loin) parmi l'ensemble des chemins possibles entre deux pixels. Cette distance est étudiée dans [6] et dans [7]. Le MBD possède de nombreuses propriétés théoriques intéressantes et constitue un outil efficace dans les applications de traitement d'images et de vision par ordinateur, en particulier pour la détection d'objets saillants [8–13], segmentation interactive [14, 15] et localisation d'objet [16].

Récemment, la pseudo-distance du Dahu a été introduite dans le carde de la morphologie mathématique dans le but de se rapprocher de la MBD. Cette distance est calculée en considérant une image comme un paysage (on parle aussi de sa vue topographique). Différente de l'approche de [9] qui calcule le MBD directement dans l'espace image, la pseudo-distance du Dahu peut être efficacement calculée sur une représentation arborescente de l'image (l'arbre de formes). Grâce à cette approche, le calcul de la pseudo-distance du Dahu est très rapide.

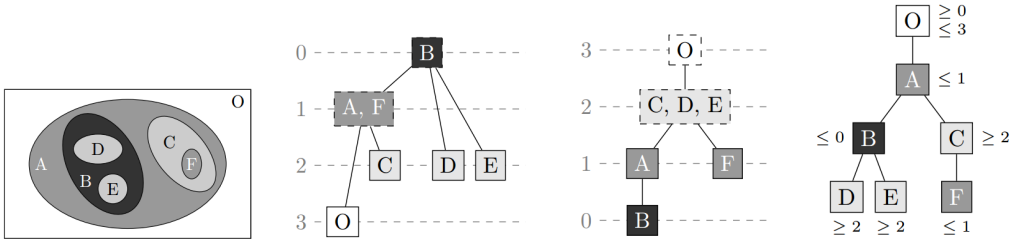
Cette distance ayant été initialement développée pour les images en niveaux de gris, nous proposons dans cette thèse une extension de cette transformation aux images multivariées ; nous l'appelons la pseudo-distance du Dahu vectoriel. Un moyen efficace de la calculer est proposé dans cette thèse. En outre, nous démontrons la robustesse de la pseudo-distance du Dahu vectoriel par rapport à d'autres distances sur plusieurs exemples. Cette distance est prometteuse pour la détection d'objets saillants, la recherche du plus court chemin et la segmentation d'images hiérarchiques. Une combinaison de ces applications déduites de la pseudo-distance du Dahu est intégrée dans un cadre complet pour la détection de documents à partir d'images provenant de caméras.

2 Fondements theoriques

2.1 L'arbre des formes et l'arbre des formes couleur

L'arbre de formes (AdF) est une représentation auto-duale d'une image fusionnée à partir de min-tree et de max-tree. Les nœuds racines du Min- et Max-tree représentent l'ensemble de l'image, tandis que les nœuds feuilles correspondent aux minima, respectivement aux maxima de l'image. Ensuite, la relation d'inclusion est utilisée pour exprimer le lien entre les nœuds et leurs parents.

Une image u est définie comme une fonction: $X \to \mathbb{N}$. Avec une valeur $\lambda \in \mathbb{N}$, les coupes supérieure et inférieure sont définies comme: $[u \geq \lambda] = \{x \in X | u(x) \geq \lambda\}$ et $[u < \lambda] = \{x \in X | u(x) < \lambda\}$. Nous désignons CC comme l'ensemble des



(a) Image (b) Max-tree. (c) Min-tree. (d) ToS.

FIGURE 1.1: Représentations arborescentes basées sur le principe de décomposition.

composantes connectées correspondantes aux coupes inférieure et supérieure de u. Les Max-tree $T_{\geq}(u)$ et Min-tree $T_{<}(u)$ sont ensuite déduits respectivement de ces ensembles de composantes connectées en tant que $T_{\geq}(u) = \{\Gamma \in CC([u \geq \lambda])\}_{\lambda}$ and $T_{<}(u) = \{\Gamma \in CC([u < \lambda])\}_{\lambda}$.

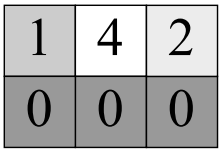
L'arbre des formes est une décomposition d'images en niveaux de gris en composantes connectées, appelées formes, qui peuvent être disposées en un arbre sous la relation d'inclusion. Une forme est une composante connectée (remplissage de cavité) sans trou à l'intérieur. Avec l'opérateur de remplissage de cavité (ou de saturation) indiqué par Sat, nous avons l'ensemble de toutes les formes (arbre de formes) : $\mathfrak{S}(u) = \{Sat(\Gamma); \Gamma \in CC([u < \lambda]) \cup CC([u \geq \lambda])\}_{\lambda}$. Deux lignes de niveau (à des niveaux différents ou non) ne peuvent pas se croiser. Des exemples des Min/Max-tree et de l'AdF sont illustrés dans la Fig. 1.1.

Comme nous l'avons mentionné précédemment, l'arbre des formes est défini dans les images en niveaux de gris. Pour calculer l'arbre des formes des images multivariées, c'est plus difficile. La relation d'ordre des valeurs dans l'arbre des formes doit être totale, sinon les composantes connectés peuvent se chevauchées, et la condition d'inclusion ne tient pas. Dans [17], les auteurs proposent une nouvelle approche pour traiter les images multivariées. Au lieu de construire un ordre total, ils s'appuient sur la relation d'inclusion entre les composantes de l'arbre marginal des formes. Leur algorithme est une procédure en 5 étapes basée sur deux parties principales. La première partie est la construction d'un graphe de formes (GdF) à partir de l'ensemble des AdFs qui est calculé à partir de chaque canal d'image marginal. La seconde est la déduction d'un seul arbre des formes multivariées (AdFM) du GdF basé sur le calcul des attributs sur le GdF.

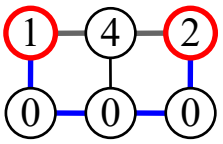
2.2 La distance de barriere minimum

Dans les applications de traitement d'images, un domaine d'image est associé à un graphique dans lequel les sommets représentent des pixels discrets sur l'image. Un chemin dans un graphe X est une séquence $\pi = \langle ..., p_i, p_{i+1} ... \rangle$ (où chaque p_i est un sommet de la valeur). De plus, l'ensemble des chemins allant du sommet x au sommet x' est indiqué par $\Pi(x, x')$. *L'intensité de la barrière* (aussi appelé *distance de la barrière* ou *coût*) τ d'un chemin π dans l'image en niveaux de gris u est défini comme :

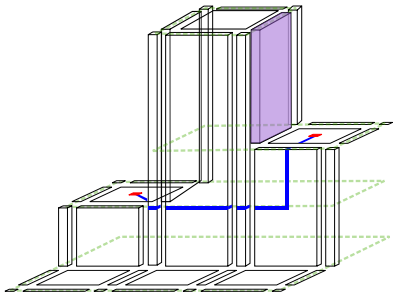
$$\tau_u(\pi) = \max_{p_i \in \pi} u(p_i) - \min_{p_i \in \pi} u(p_i). \tag{1.1}$$



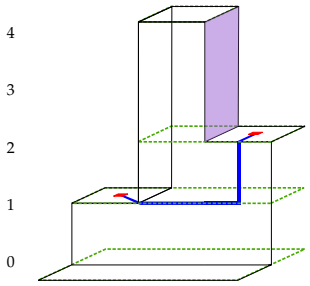
(a) Une image u.



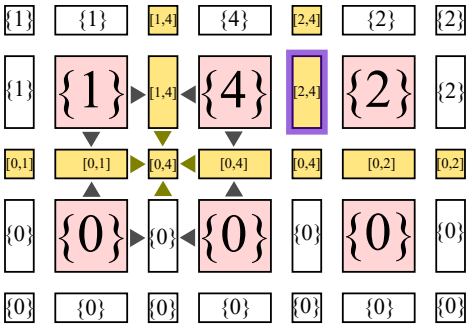
(b) u sous forme de graphe.



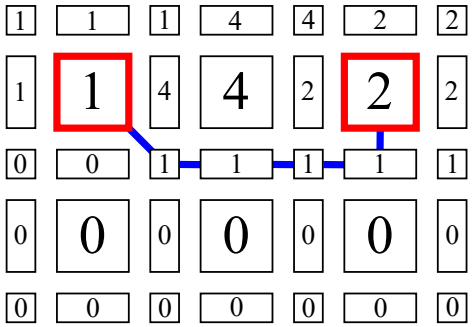
(c) Version 3D de $\widetilde{u}$ donnée en (e).



(d) u en surface.



(e) Image à valeur d'intervalle $\widetilde{u}$.



(f) Un chemin minimal dans un $\overline{u} \Subset \widetilde{u}$.

FIGURE 1.2: Représentation d'images pour le calcul des distances de barrière.

La *minimum barrier distance* d^{MB} (MBD) entre deux sommets x et x' est alors défini par :

$$d_u^{\text{MB}}(x, x') = \min_{\pi \in \Pi(x,x')} \tau_u(\pi), \tag{1.2}$$

La MBD est donc le minimum des intensités de barrière de tous les chemins entre deux sommets donnés. Une illustration de cet opérateur est donnée dans la Fig. 1.2(b). Le chemin bleu, qui correspond à une séquence $\langle 1, 0, 0, 0, 2 \rangle$, est considéré comme le plus court chemin entre ces deux points rouges. La MBD correspondante est alors égal à 2.

2.3 La pseudo-distance Dahu

Une nouvelle version discrète du MBD, appelée la pseudo-distance Dahu, est définie dans [18] et considère une image comme une surface continue (voir Fig. 1.2(d)). Le chemin bleu optimal entre les deux points rouges a une distance égale à un.

Une image en niveau de gris peut être vue comme une fonction $u : \mathbb{Z}^2 \to \mathbb{N}$. Lorsque nous représentons une image à l'aide d'une surface, nous ne pouvons pas utiliser de fonctions scalaires. Plus exactement, dans [19], les auteurs proposent de remplacer le domaine $\mathbb{Z}^2$ par l'espace topologique discret $\mathbb{H}^2$ de Khalimsky 2D

(aussi appelées complexes cubiques), et le codomaine $\mathbb{N}$ par l'ensemble $\mathbb{I}_{\mathbb{N}}$ des intervalles des nombres naturels. Le complexe cubique 2D, qui est illustré dans Fig. 1.2(e) est un ensemble d'éléments 2D, 1D et 0D, dans lesquels les éléments 2D sont les pixels originaux, 1D et 0D sont les inter-pixels qui prennent la valeur d'intervalle à ses voisins 2D. Par exemple, l'élément 1D jaune dans Fig. 1.2(e), qui est délimité par une bordure violette, correspond à la partie verticale violette dans Fig. 1.2(c). A partir d'une image scalaire u, on construit une image à valeur d'intervalle $\widetilde{u}$, qui représente réellement la surface correspondant à u.

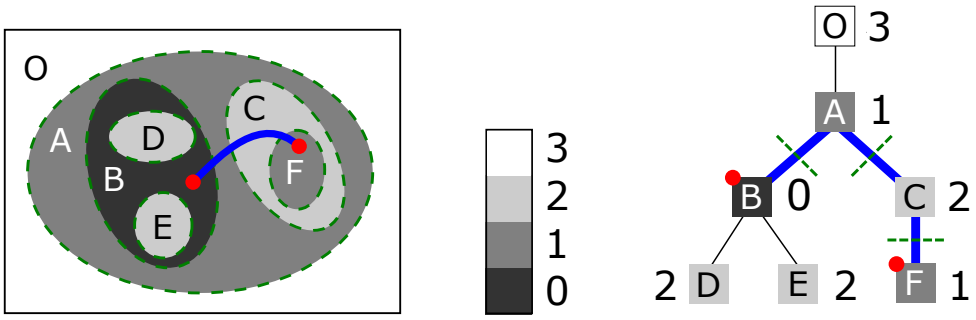
La relation d'inclusion entre une image scalaire et une image à valeur d'intervalle est indiquée par $\Subset$. Le Fig. 1.2(f) représente une image scalaire $\overline{u}$ qui est "incluse" dans l'image à valeur intervalle $\widetilde{u}$ représentée dans la Fig. 1.2(e) ; alors on peut écrire $\overline{u} \Subset \widetilde{u}$. L'adaptation du MBD sur l'image à valeur d'intervalle, appelée la pseudo-distance du Dahu (voir [19]), est notée d^{DAHU}. Ensuite, la pseudo-distance du Dahu entre deux pixels x et x' sur l'image originale u est défini comme :

$$d_u^{\text{DAHU}}(x, x') = \min_{\overline{u} \Subset \widetilde{u}} d_{\overline{u}}^{\text{MB}}(h_x, h_{x'}) \tag{1.3}$$

$$= \min_{\overline{u} \Subset \widetilde{u}} \min_{\pi \in \Pi(h_x, h_{x'})} \tau_{\overline{u}}(\pi), \tag{1.4}$$

où h_x et $h_{x'}$ sont les éléments 2D du complexe cubique correspondant respectivement à x et x'. Cela signifie que nous cherchons un chemin minimal dans le complexe cubique, avec la définition classique du MBD, et considérons toutes les fonctions scalaires possibles $\overline{u}$ qui sont "incluse" dans la carte à valeurs par intervalles $\widetilde{u}$.

2.4 Calcul efficace de la pseudo-distance Dahu à l'aide de l'arbre de formes



(a) Image u. (b) Échelle de gris. (c) L'arbre $\mathfrak{S}(u)$.

FIGURE 1.3: L'arbre des formes d'une image permet d'exprimer et de calculer facilement les cartes de la pseudo-distance du Dahu(voir [19]).

La pseudo-distance du Dahu peut être calculée facilement et efficacement grâce à la représentation arborescente de l'image (l'arbre de formes). Le chemin minimal entre deux points de l'image correspond à un chemin entre deux nœuds de l'arbre. Sur Fig. 1.3(a), le trajet entre deux points (x, x') indiqués par des balle rouges est représenté par une ligne bleue, qui part de la région B, passe par A et C, puis se termine dans la région F. Un tel chemin est minimal parce que chaque chemin dans $\Pi(x, x')$ devrait au moins traverser ce même ensemble de lignes de niveau pour passer de x à x'. En fait, ce chemin dans l'espace image est exactement le même que le chemin (le plus court en nombre de noeuds) dans l'arbre des formes entre les noeuds t_x et $t_{x'}$:

$$\dot{\pi}(t_x, t_{x'}) := \langle t_x, \ldots, \mathrm{lca}(t_x, t_{x'}), \ldots, t_{x'} \rangle,$$

où $\mathrm{lca}(t_x, t_{x'})$ est l'ancêtre commun le plus bas de la paire $(t_x, t_{x'})$ (voir le chemin bleu sur l'arbre représenté dans Fig. 1.3(c)). Notez qu'un chemin dans un arbre est indiqué par $\dot{\pi}$ pour le distinguer des chemins dans l'espace image.

La pseudo-distance du Dahu dans l'espace image entre deux points x et x' peut être écrit comme la distance de barrière minimale entre les deux noeuds t_x et $t_{x'}$ représentant les composantes dans l'arbre des formes contenant respectivement x et x' :

$$d_u^{\mathrm{DAHU}}(x,\, x') = d^{\mathrm{MB}}_{\mathfrak{S}(u)}(t_x,\, t_{x'}) \tag{1.5}$$

$$= \max_{t \in \dot{\pi}(t_x, t_{x'})} \mu_u(t) - \min_{t \in \dot{\pi}(t_x, t_{x'})} \mu_u(t), \tag{1.6}$$

où $\mu_u(t)$ désigne le niveau de gris associé au noeud t. de l'arbre des formes $\mathfrak{S}(u)$ de l'image u. Par exemple, dans Fig. 1.3(c), le chemin bleu donne la séquence des valeurs de noeud $\langle 0, 1, 2, 1 \rangle$, donc le Dahu pseudo-distance est 2. Il n'y a pas besoin de trouver la meilleure image scalaire $\overline{u} \preccurlyeq \widetilde{u}$, ni de trouver le meilleur chemin $\pi \in \Pi(x,\, x')$ dans l'espace image.

3 Aller plus loin avec la pseudo-distance Dahu

3.1 Extension de la pseudo-distance Dahu à des images multivariées

Dans [20], la distance de barriere minimum vectorielle (VMBD) est proposée pour calculer la MBD sur une image multivariée. Cependant, ce VMBD n'est pas facile à calculer directement sur l'image. De plus, le VMBD n'est pas efficace pour calculer les distances entre plusieurs points dans les images. Pour résoudre ce problème, dans cette section, nous présentons la pseudo-distance du Dahu étendue aux images multivariées en utilisant l'arbre des formes.

Considérons que $\boldsymbol{u}$ est une image multivariée, t est un noeud du MToS de $\boldsymbol{u}$, et $\mu_{\boldsymbol{u}}(t)$ est la valeur vectorielle associée au noeud t, i est l'index du canal. On peut alors étendre le Dahu :

$$d_{\boldsymbol{u}}^{\mathrm{DAHU}}(x,\, x') := \textstyle\sum_{i \in \{1..N\}} \alpha_i\, \tau_{\boldsymbol{u}}^{(i)}(\, \dot{\pi}(t_x,\, t_{x'})\,). \tag{1.7}$$

avec :

$$\tau_{\boldsymbol{u}}^{(i)}(\dot{\pi}) := \max_{t \in \dot{\pi}} \mu_{\boldsymbol{u}}^{(i)}(t) - \min_{t \in \dot{\pi}} \mu_{\boldsymbol{u}}^{(i)}(t), \tag{1.8}$$

où α_i est le coefficient de chaque canal.

Pour les images en couleurs RGB, notre équation devient :

$$d_{\boldsymbol{u}}^{\mathrm{DAHU}}(x,\, x') = \frac{1}{3} \textstyle\sum_{i \in \{R,G,B\}} \tau_{\boldsymbol{u}}^{(i)}(\, \dot{\pi}(t_x,\, t_{x'})\,). \tag{1.9}$$

Le MToS est calculé à partir des ToS de chaque canal d'image en fusionnant certaines formes marginales. Le nœud de l'arbre final est associé à plusieurs valeurs de l'image. Par conséquent, un nœud est affecté à une valeur unique calculée à partir

de l'ensemble des valeurs qu'il contient. Dans notre cas, nous réglons chaque nœud du MToS en utilisant la valeur médiane de ses pixels.

3.2 Améliorer la pseudo-distance de Dahu

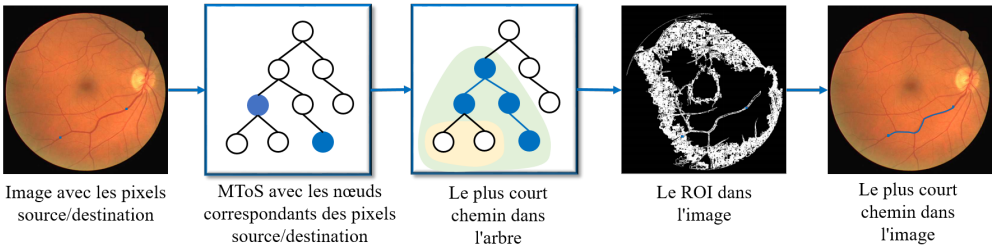


FIGURE 1.4: Schéma pour l'application de recherche du plus court chemin.

Dans cette section, nous présentons une amélioration de la pseudo-distance du Dahu en prenant en compte l'information spatiale entre deux pixels de l'image. Cette amélioration est en fait une procédure "deux-étapes", qui est illustrée dans la Fig. 1.4. Dans la première étape, considérons deux pixels donnés x et x', nous cherchons le plus court chemin au sens de la pseudo-distance du Dahu dans l'espace de l'arbre entre deux noeuds t_x et $t_{x'}$, qui correspondent aux deux pixels donnés (voir les noeuds bleus sur l'arbre représenté dans Fig. 1.4.

Notez que chaque noeud t_x sur l'arbre représente une composante connectée $CC(t_x)$ dans le domaine image. Nous désignons $\Re^*(t_x)$ la région qui est l'union des composantes connectées qui correspondent aux descendants du noeud t_x, et $\Re(t_x)$ l'union des $\Re^*(t_x)$ et la composante connectée $CC(t_x)$ du noeud t_x lui-même. Après avoir calculé le chemin le plus court $\dot{\pi}(t_x, t_{x'})$, on trouve une région dans l'espace image qui relie deux pixels x et x'. Nous appelons cela $ROI(t_x, t'_x)$. Ce ROI est en fait l'ensemble de tous les chemins possibles entre les deux points donnés dans l'espace image minimisant la pseudo-distance du Dahu.

Dans la deuxième étape, nous voulons trouver le plus court chemin (dans l'espace de l'image) entre les deux pixels x et x', qui appartientent au $ROI(t_x, t_{x'})$, pour qu'il ait la longueur la plus courte dans l'espace image. Ce chemin optimal a différentes significations. Ce chemin n'est pas seulement le chemin le plus court dans l'espace des couleurs mais aussi le chemin le plus court dans l'espace image. Un exemple du chemin optimal est représenté dans Fig. 1.4. Le chemin le plus court se trouve dans cette région en utilisant l'algorithme heuristique A^* (voir [21]).

3.3 Détection d'objets saillants basée sur la pseudo-distance Dahu

Pour utiliser la pseudo-distance du Dahu dans la détection des objets saillants, nous adoptons deux hypothèses sur le fond des images naturelles, *bordure* et *connectivité*, qui sont proposés dans [22]. La première hypothèse indique que la bordure du domaine de l'image est principalement du fond. Dans la deuxième hypothèse, les auteurs supposent que les régions d'arrière-plan sont grandes et homogènes, et que les éléments d'arrière-plan ont tendance à se connecter avec la bordure de l'image.

Nous pouvons définir la carte de saillance sur la base de la pseudo-distance du Dahu de la manière suivante :

$$S_u^{\text{DAHU}}(x, X') := \min_{x' \in X'} d_u^{\text{DAHU}}(x, x'),$$

où X' est un ensemble de points du domaine de l'image u.

Définissons l'ensemble correspondant des noeuds sur $\mathfrak{S}(u)$ de X' :

$$T_{X'} := \{ t_{x'} ; x' \in X' \}. \tag{1.10}$$

Ensuite, nous obtenons la carte des saillants:

$$S_u^{\mathrm{DAHU}}(x, X') = S_{\mathfrak{S}(u)}^{\mathrm{MBD}}(t_x, T_{X'}), \tag{1.11}$$

3.4 Segmentation interactive basée sur la pseudo-distance de Dahu

Dans cette section, nous proposons un modèle amélioré de segmentation interactive utilisant la pseudo-distance de Dahu. Nous appliquons une approche statistique pour obtenir plus d'informations sur les régions d'avant-plan et d'arrière-plan à partir de marqueurs. Le schéma de notre méthode est le suivant présenté dans Fig. 1.5.

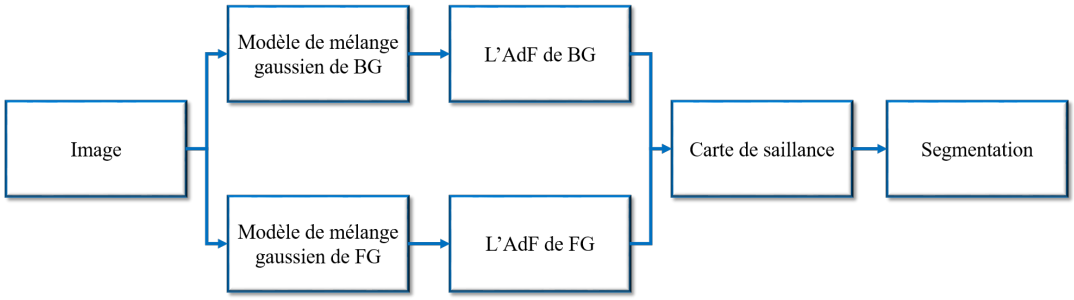


FIGURE 1.5: Segmentation interactive basée sur la pseudo-distance de Dahu.

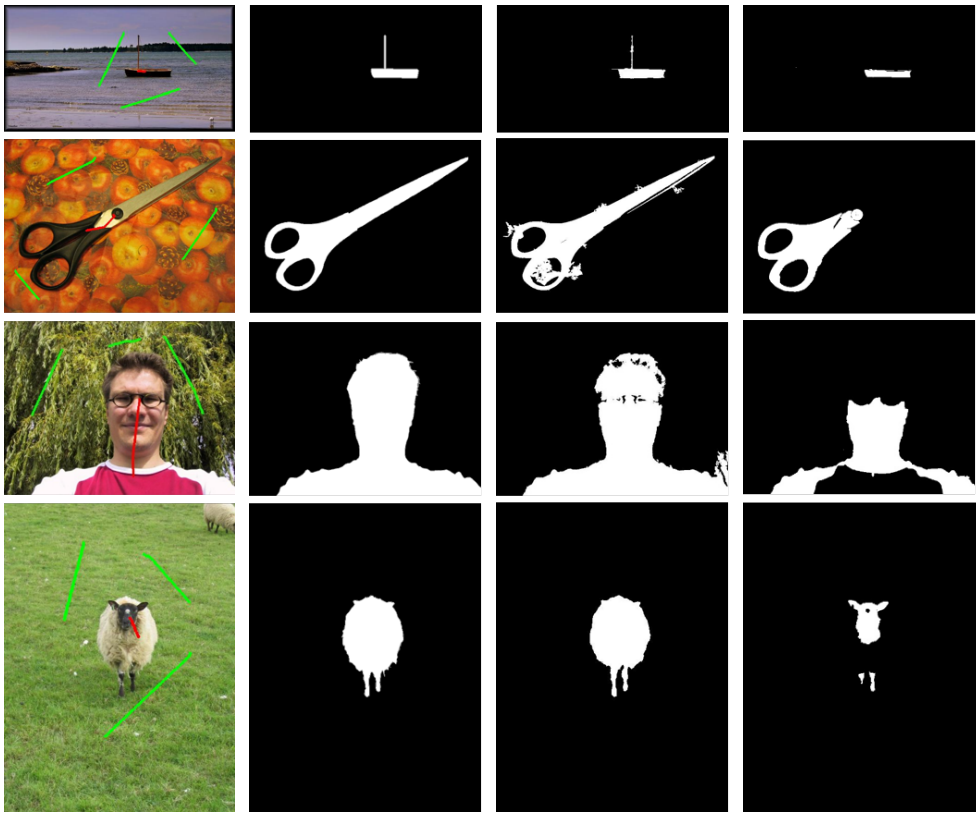
Après avoir ajusté les modèles GMM, nous estimons une probabilité de chaque pixel. Dans l'étape suivante, nous construisons deux AdFs pour représenter ces deux cartes de probabilité. Nous marquons le nœud de l'arbre qui correspond aux marqueurs. Ensuite, la pseudo-distance de Dahu est utilisée pour calculer la carte des saillances à partir des nœuds marqués. Ces deux cartes de distance sont comparées l'une à l'autre pour déterminer l'étiquette du pixel de l'image. Ensuite, l'image avec les étiquettes est reconstruite.

Nous présentons quelques résultats qualitatifs de notre méthode par rapport à la méthode Grabcut [23] dans la Fig. 1.6. Les résultats de notre méthode comparé à Grabcut sont illustrés respectivement in Fig. 1.6(c) and Fig. 1.6(d).

TABLE 1.1: Les résultats de la segmentation interactive

Distance metrics	Geodesic	MBD	MSD16	MSD32	Grabcut	Our
Weighted F	0.6469	0.6166	0.6821	0.6807	0.6392	**0.7143**

Table 1.1 présente quelques résultats quantitatifs de notre méthode par rapport aux approches les plus récentes. Pour évaluer la qualité des méthodes de segmentation interactive, nous utilisons le F-score . Notez que, les résultats de la géodésique, la MBD et la MSD sont extraites de [24]. Notre méthode permet d'obtenir de meilleurs résultats que la méthode Grabcut, qui est habituellement utilisée dans le cadre de nombreuses applications interactives de segmentation.



(a) Ori (b) GT (c) GMM-Dahu (d) Grabcut

FIGURE 1.6: Comparaison de la segmentation interactive entre la méthode proposée et la méthode Grabcut [23].

3.5 Détection de documents basée sur la pseudo-distance Dahu

Dans cette section, nous nous concentrons sur la segmentation automatique des documents dans les photos ou les vidéos issues de smartphones à l'aide de la saillance visuelle. Notre méthode est une méthode basée sur la saillance, qui se compose de quatre étapes principales.

Au début, nous supposons que nous avons un contraste élevé entre le document et l'arrière-plan. Ainsi, nous considérons les pixels le long de la bordure de l'image comme des nœuds de graine pour calculer la carte de saillance visuelle [22]. Nous calculons la carte de saillance de la même manière que la section précédente en utilisant la Eq. (1.11). Cette méthode nous permet de connaître la position du document dans l'image.

Pour segmenter la région du document, nous proposons d'utiliser une segmentation hiérarchique de l'image en parallèle avec le calcul de la carte de saillance. Notre méthode commence avec l'algorithme SLIC [25] pour partitionner une image en plusieurs régions appelées super-pixels. La segmentation hiérarchique d'image segmente une image en plusieurs partitions, ce qui réduit le nombre d'éléments de l'image ce qui reduit l'espace de rechercher. Nous utilisons une méthode de simplification et de segmentation de l'image basée sur la distance de Dahu.

Ensuite, nous combinons le résultat de la carte de saillance de Dahu et la segmentation hiérarchique de l'image pour obtenir une carte finale de saillance. La valeur finale de chaque région R_i est la moyenne de la carte de saillance de chaque pixel de la région. Dans cette carte de saillance finale, les pixels de la région du document

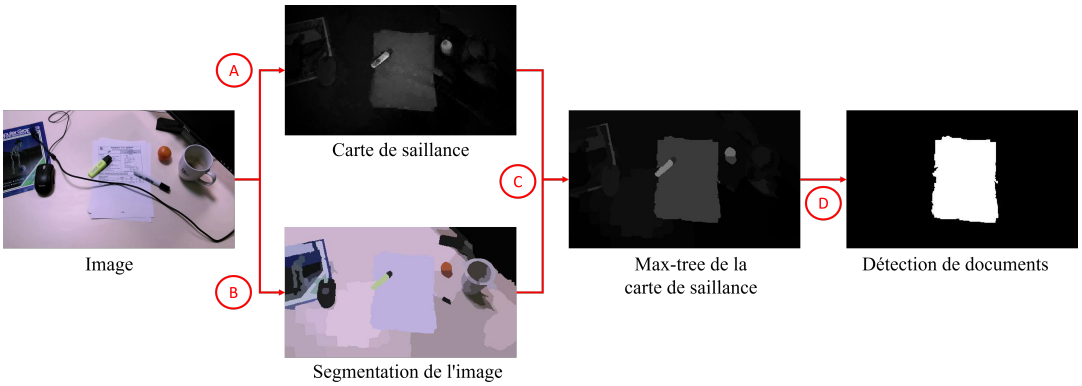


FIGURE 1.7: Schéma efficace pour la détection de documents.

sont plus clairs que les autres pixels. En d'autres termes, la région du document est mise en surbrillance dans l'image. Par conséquent, nous construisons un max-tree de cette carte de saillance.

Method	Bg 1	Bg 2	Bg 3	Bg 4	Bg 5	Overall	Runtime
A2iA-1	0.972	0.801	0.912	0.635	0.189	0.779	?
A2iA-2	0.960	0.806	0.912	0.826	0.189	0.809	?
ISPL-CVML	0.987	0.965	0.985	0.977	0.856	0.966	?
LRDE [26]	0.987	0.978	0.989	0.984	0.861	0.972	**1min**
NetEase	0.962	0.955	0.962	0.951	0.222	0.882	?
SEECS-NUST	0.888	0.826	0.783	0.781	0.011	0.739	?
RPPDI-UPE	0.827	0.910	0.970	0.365	0.216	0.741	?
SmartEngines [27]	0.989	0.983	0.990	0.979	0.688	0.955	?
L. R. S. Leal [28]	0.961	0.944	0.965	0.930	0.412	0.895	0.43s
LRDE-2 [29]	0.905	0.936	0.859	0.903	?	?	0.04s
Ours	0.985	0.982	0.987	0.980	0.848	0.97	**3.7s**
Smartdoc ave. [30]	0.9465	0.9031	0.9377	0.8122	0.4041	0.8552	?

TABLE 1.2: Résultats quantitatifs sur les données des concours Smartdoc 2015. La couleur rouge (resp. bleue) indique le meilleur (resp. le second) résultat dans chaque cas. Notre méthode obtient la deuxième meilleure note globale. Elle est au même niveau que avec la méthode LRDE [26], mais environ 16 fois plus rapide que leur méthode.

Finalement, supposons que le document candidat soit représenté dans le max-tree, le problème de segmentation du document est alors de trouver le document dans l'espace arborescent. Pour ce faire, nous attribuons un attribut à chacun d'eux qui correspond à un nœud de max-tree. Ici, nous utilisons une hypothèse préalable qui est le document a une forme quadrilatérale. Pour calculer notre attribut, nous calculons séquentiellement l'attribut sur chaque noeud de l'arbre et nous observons dans quelle mesure ces attributs correspondent aux critères du document. L'idée est de considérer les maxima locaux de la carte d'energy comme des candidats pour la détection de documents. Nos critères sont les suivants :

1. À quel point la bordure de la forme correspond à un quadrilatère:

$$E_f(A) = \frac{|A \cap Quad(A)|}{|A \cup Quad(A)|} \tag{1.12}$$

2. Les angles entre les lignes du haut (resp. du bas), indiqués par TL (resp. BL), et entre les lignes de gauche (resp. la droite), notées LL (resp. RL) :

$$E_a(A) = \frac{\cos(TL, BL) + \cos(LL, RL)}{2} \tag{1.13}$$

3. La valeur de la carte de saillance de chaque nœud de l'arbre :

$$E_s(A) = S_u^{\text{DAHU}}(A) \tag{1.14}$$

L'attribut final est calculé par cette équation :

$$E(A) = E_f(A) \times E_a(A) \times E_s(A) \tag{1.15}$$

Une fois que l'attribut $E(A)$ est disponible, nous pouvons rechercher le nœud "le plus probable" de l'arbre qui maximise cette fonction d'attribut.

Pour étendre la détection de documents dans un flux vidéo, une méthode simple de suivi compare les positions des formes dans des images consécutives.

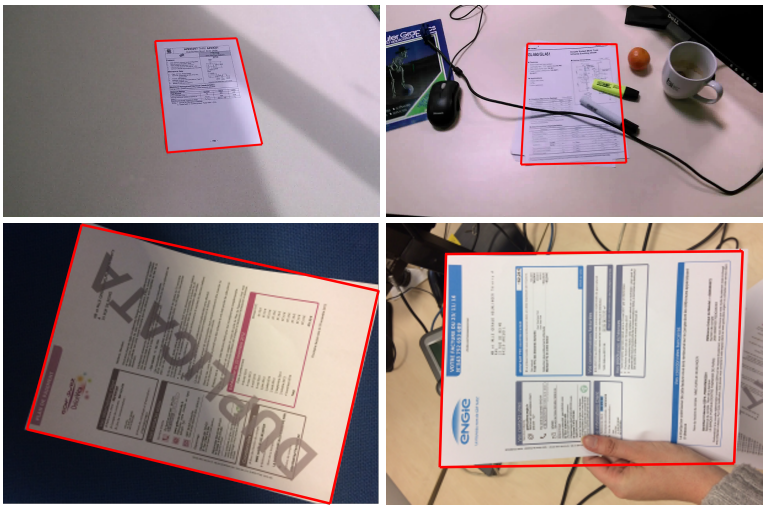


FIGURE 1.8: Quelques résultats qualitatifs de notre méthode. Ces images montrent la robustesse de notre méthode au flou ou au document partiellement courbé.

Pour évaluer, nous utilisons l'index de Jaccard entre le document et la vérité de terrain. Dans la Table 1.2, notre méthode obtient la deuxième meilleure note globale sur 12 méthodes. La différence avec la méthode du premier rang (LRDE) est négligeable (0,972 vs 0,97), mais nous sommes environ 16 fois plus rapides (1 min vs 3,7s).

Dans la Fig. 1.8, nous montrons les résultats de notre méthode sur quelques images difficiles.

La Fig. 1.9 démontre le compromis entre le temps d'exécution du processus et le score global. Si nous augmentons le paramètre de mise à l'échelle et diminuons le nombre de super-pixels, le temps d'exécution est beaucoup plus court, tandis que la précision reste acceptable.

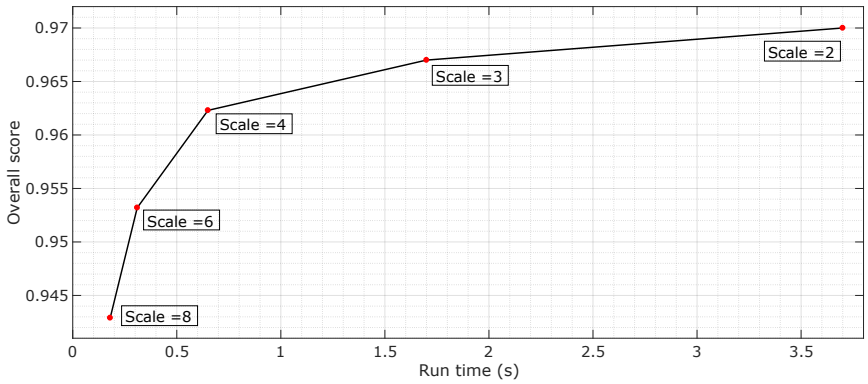


FIGURE 1.9: Le compromis entre le temps d'exécution (résolution de l'image, i.e. l'échelle de l'image) et la précision globale. Même à faible résolution, notre méthode permet d'obtenir une note globale de 0,962 pour un temps de fonctionnement égal à 0.65s.

4 Conclusions

Dans cette thèse, la représentation hiérarchique de l'image, en particulier la représentation arborescente basée sur la décomposition par seuil, a été présentée comme un domaine de recherche très prometteur. De nombreuses méthodes de traitement ont été appliquées à la hiérarchie pour prouver l'efficacité de ce type de représentation d'images. Dans cette thèse, nous prenons en compte une autre approche basée sur la transformation de la distance pour enrichir les capacités applicables de représentation hiérarchique de l'image.

En outre, nous avons étudié la pseudo-distance de Dahu et nous avons introduit de multiples améliorations de cette pseudo-distance. Tout d'abord, nous avons introduit une extension vectorielle de la pseudo-distance de Dahu capable de traiter des images multicanaux. Evidemment, cette pseudo-distance vectorielle de Dahu peut gérer des images couleur qui sont déjà une grande amélioration, mais ne se limite pas aux images à trois canaux. Deuxièmement, nous avons amélioré la pseudo-distance de Dahu en combinant la pseudo-distance du Dahu avec des informations sur le domaine spatial des images. Nous avons également prouvé que notre pseudo-distance vectorielle de Dahu est moins affectée par le bruit dans l'image que les autres pseudo-distances basées sur MB.

Nous avons proposé une nouvelle méthode de détection des documents dans les vidéos capturées par les smartphones, avec très peu de connaissances a priori sur les documents et les images. Nous avons démontré l'efficacité de la pertinence visuelle pour la détection de documents.

Contents

Acknowledgements iii

Abstract v

Résumé vii

Résumé long ix
1 Introduction . . . ix
2 Fondements theoriques . . . x
2.1 L'arbre des formes et l'arbre des formes couleur . . . x
2.2 La distance de barriere minimum . . . xi
2.3 La pseudo-distance Dahu . . . xii
2.4 Calcul efficace de la pseudo-distance Dahu à l'aide de l'arbre de formes . . . xiii
3 Aller plus loin avec la pseudo-distance Dahu . . . xiv
3.1 Extension de la pseudo-distance Dahu à des images multivariées xiv
3.2 Améliorer la pseudo-distance de Dahu . . . xv
3.3 Détection d'objets saillants basée sur la pseudo-distance Dahu . xv
3.4 Segmentation interactive basée sur la pseudo-distance de Dahu xvi
3.5 Détection de documents basée sur la pseudo-distance Dahu . . xvii
4 Conclusions . . . xx

1 Introduction 1
1.1 Image representation . . . 1
1.2 Distance transform . . . 3
1.3 Main contributions . . . 4
1.4 Manuscript organization . . . 4

2 Theoretical Background 9
2.1 Image Representation . . . 9
2.1.1 Digital image . . . 9
2.1.2 Classical Image Representation . . . 10
2.1.2.1 Pixel-based representation . . . 10
2.1.2.2 Region-based representation . . . 11
2.2 Hierarchical Partition Trees . . . 12
2.2.1 Quadtree . . . 13
2.2.2 Minimum spanning tree . . . 14
2.2.3 α-Tree . . . 16
2.2.4 Binary Partition Tree . . . 17
2.2.5 Conclusion . . . 18
2.3 Tree based on the threshold decomposition . . . 19
2.3.1 Min Tree and Max Tree . . . 19
2.3.2 Tree of shapes . . . 19

2.3.3 Multivariate Tree of shapes . . . 21
2.3.4 Conclusion . . . 23
2.4 Tree simplification . . . 23
2.4.1 Tree filtering approach . . . 23
2.4.1.1 Increasing Criterion . . . 24
2.4.1.2 Non-increasing Criterion . . . 24
2.4.2 Hierarchical image segmentation . . . 25
2.5 Image segmentation . . . 28
2.5.1 Superpixel segmentation . . . 28
2.5.2 Contour-based segmentation . . . 30
2.5.3 Watershed . . . 31
2.5.4 Graph-based segmentation . . . 32
2.5.4.1 Normalized cut . . . 32
2.5.4.2 Graph cut . . . 33
2.5.4.3 Felzenswalb and Huttenlocher method . . . 34
2.5.5 Interactive segmentation . . . 36
2.6 Distance function . . . 37
2.6.1 Definitions and examples . . . 38
2.6.2 Geodesic distance . . . 40
2.6.3 Minimum Barrier distance . . . 42
2.6.4 Distance map based on the minimum barrier distance . . . 42
2.6.5 MB-based distances . . . 43
2.6.6 The Dahu pseudo-distance . . . 47
2.6.7 Efficient Dahu pseudo-distance computation using the tree of shapes . . . 48
2.6.8 Saliency map based on the Dahu pseudo-distance . . . 49
2.6.9 Conclusion . . . 50
2.7 Visual saliency detection . . . 50
2.7.1 Bottom-up methods . . . 51
2.7.1.1 Contrast prior . . . 51
2.7.1.2 Center Prior . . . 53
2.7.1.3 Boundary and Connectivity Prior . . . 53
2.7.1.4 Graph-based Approach . . . 54
2.7.2 Top-down methods . . . 55
2.8 Document detection . . . 56
2.9 Conclusions . . . 57

3 Dahu pseudo-distance improvements and applications 59
3.1 Dahu pseudo-distance improvements . . . 59
3.1.1 Improvement of speed performance: simultaneous computations of the Dahu pseudo-distance and the tree of shapes . . . 59
3.1.2 Extending the Dahu pseudo-distance to multivariate images . . . 61
3.2 Dahu pseudo-distance applications . . . 64
3.2.1 Shortest path finding based on the Dahu pseudo-distance . . . 65
3.2.2 Salient object detection based on the Dahu pseudo-distance . . . 67
3.2.3 Interactive segmentation based on the Dahu pseudo-distance . . . 69
3.2.3.1 A simple version for interactive segmentation based on the Dahu pseudo-distance . . . 69
3.2.3.2 An extended version for interactive segmentation based on the Dahu pseudo-distance . . . 70
3.2.4 Image segmentation based on the Dahu pseudo-distance . . . 72

3.2.5 Document detection based on the Dahu pseudo-distance . . . 74
3.2.5.1 A simple version for document detection based on the Dahu pseudo-distance . . . 74
3.2.5.2 An extended version for document detection based on the Dahu pseudo-distance . . . 75

4 Validation of the Dahu pseudo-distance **79**
4.1 Visual saliency detection . . . 79
4.1.1 Comparison of saliency maps obtained by the usual Dahu pseudo-distance on separate channels and by our vectorial Dahu pseudo-distance . . . 81
4.1.2 Comparison of saliency maps of the vectorial Dahu pseudo-distance with state-of-the-art methods . . . 86
4.2 Efficiency and robustness of the algorithm . . . 88
4.2.1 Ability to distinguish object and background . . . 88
4.2.2 Robustness against noise . . . 89
4.3 Speed performance . . . 90

5 Applications and Evaluations **93**
5.1 Shortest path in images . . . 93
5.2 Dahu pseudo-distance on multimodal images and hyperspectral images 95
5.2.1 Multimodal images . . . 95
5.2.2 Multispectral images . . . 96
5.3 Dahu distance in interactive segmentation . . . 98
5.3.1 Dahu pseudo-distance in interactive segmentation on synthetic images with taking into account noise and position of seeds . . . 98
5.3.2 Dahu pseudo-distance in interactive segmentation concerning the numbers of markers . . . 100
5.3.3 A simple interactive segmentation based on the Dahu pseudo-distance on natural images . . . 101
5.3.4 An extension in interactive segmentation based on the Dahu pseudo-distance . . . 102
5.4 Image segmentation based on the Dahu pseudo-distance . . . 104
5.5 Document Detection based on the Dahu pseudo-distance . . . 106
5.5.1 Simple saliency based method for document detection . . . 106
5.5.1.1 Experiment setting . . . 106
5.5.1.2 Experimental Results . . . 107
5.5.1.3 Limitation . . . 108
5.5.2 Extended saliency based method for document detection . . . 109
5.5.2.1 Dataset and Evaluation . . . 109
5.5.2.2 Experiments and Results . . . 109

6 Conclusion **113**

List of Figures

1.1 Représentations arborescentes basées sur le principe de décomposition. . . . xi
1.2 Représentation d'images pour le calcul des distances de barrière. . . . xii
1.3 L'arbre des formes d'une image permet d'exprimer et de calculer facilement les cartes de la pseudo-distance du Dahu(voir [19]). . . . xiii
1.4 Schéma pour l'application de recherche du plus court chemin. . . . xv
1.5 Segmentation interactive basée sur la pseudo-distance de Dahu. . . . xvi
1.6 Comparaison de la segmentation interactive entre la méthode proposée et la méthode Grabcut [23]. . . . xvii
1.7 Schéma efficace pour la détection de documents. . . . xviii
1.8 Quelques résultats qualitatifs de notre méthode. Ces images montrent la robustesse de notre méthode au flou ou au document partiellement courbé. . . . xix
1.9 Le compromis entre le temps d'exécution (résolution de l'image, i.e. l'échelle de l'image) et la précision globale. Même à faible résolution, notre méthode permet d'obtenir une note globale de 0,962 pour un temps de fonctionnement égal à 0.65s. . . . xx

2.1 Graph of pixels, where nodes represent pixels in the image and edges correspond to connectivities between adjacent pixels. . . . 10
2.2 Region adjacency graph. . . . 11
2.3 An illustration of the hierarchy of image. . . . 12
2.4 An image and a its Quadtree representation. . . . 13
2.5 An image and its MST representation. . . . 14
2.6 An example of an α-tree. . . . 16
2.7 Example of BPT construction using a region merging algorithm by a priority queue. . . . 18
2.8 Tree computation of the Max- and Min-tree based on the immersion algorithms (2-steps procedure). The result of the sorting step is given over the arrow, and the tree is constructed in the inverse order. . . . 20
2.9 The computation of the tree of shapes. The result of the sorting step is given over the arrow. . . . 21
2.10 The 5-steps process for computing the MToS. Images are extracted from [130]. . . . 22
2.11 An example of the hierarchical cut. . . . 25
2.12 Optimal Cuts Pyramids: Optimal cuts using Mumford-Shah function, shown for different λ_s. Images are taken in [48]. . . . 26
2.13 Image segmentation using SLIC algorithm with the size of each superpixel are respectively equal 64, 256, and 1024 pixels [25]. . . . 29
2.14 Hierarchical segmentation from contours. Images are extracted in [52] 31
2.15 Minima, catchment basins, and watersheds on the topographic representation of a gray-scale image. . . . 32

2.16 The segmenting results by using normalized cut algorithm with different value of number of segmentation k. The image is extracted from [160]. . . . 33
2.17 A simple 2D segmentation example for a 3×3 image. The cost of each edge is reflected by the edge's thickness. . . . 34
2.18 A baseball scene (432×294 grey image), and the segmentation results produced by FH algorithm ($\sigma = 0.8$, $k = 300$). Image extracted from [90]. . . . 35
2.19 Comparison of some matting and segmentation tools. The top row shows the user interaction required to complete the segmentation or matting process. These methods are: Magic Wand [166], Intelligent Scissors [167], Bayes matting [168], Knockout 2 [169], Graph Cut [170], GrabCut [23]. The bottom row illustrates the resulting segmentation. . . 36
2.20 Discrete distance function calculated from the central pixel of an image . . . 38
2.21 Skeleton application by using distance transform approach. . . . 39
2.22 Distance function and erosion: the set X eroded by a diamond shaped structuring element of size 6 is obtained by thresholding the 4-connected distance D on X. Image is extracted from [180]. . . . 39
2.23 Segmentation of overlapping blobs by watershedding WS the complement C of their distance function D. . . . 40
2.24 Geodesics between p and q in a connected set S, and between p and X. 41
2.25 Image representations for computing barrier distances. . . . 43
2.26 The MB-based distances are used in salient object detection (see [8]). The left image is the original image, the right one is the saliency map of all pixels in the image by considering that pixels on the border of the image are the background. . . . 44
2.27 The MB-based distances are used in interactive segmentation (see [20]). The left image is the original image and the right image is the result of interactive segmentation. . . . 44
2.28 The MB-based distances are used in object localization (see [16]) and refocusing application (see [189]). . . . 45
2.29 The MB-based distances are used in object segmentation (see [192]) and superpixels segmentation (see [188]). . . . 46
2.30 The tree of shapes of an image allows to easily express and compute the Dahu pseudo-distance and distance maps (see [19]). . . . 48
2.31 An example of visual comparison between eye fixation modeling and saliency detection. . . . 51
2.32 Images and pixel-level annotations from four salient object datasets. . 52
2.33 Average annotation maps of six datasets used in benchmarking. Images taken from [212]. . . . 53
2.34 Examples from [22] showing the paths of background (in magenta) and foreground (in green) from the boundary in the top row. Bottom row shows saliency maps retrieved by their algorithm. . . . 54

3.1 The computation of the MBD and the vectorial Dahu pseudo-distance in a color image. . . . 62
3.2 The Dahu pseudo-distance in the grayscale image and in the color image. . . . 64
3.3 A scheme for shortest path finding application. . . . 65
3.4 Shapes on the cubical complex. Image is taken from [130]. . . . 65

3.5 Boundary and connectivity priors [22]. 67
3.6 Finding the shortest path from every pixel in the image to the seed set. 67
3.7 Interactive segmentation scheme on the MToS. 70
3.8 GMM model for interactive segmentation based on the Dahu pseudo-distance. 71
3.9 Image segmentation based on the Dahu pseudo-distance. 72
3.10 Simple scheme for document detection. 74
3.11 Effect of fusing four side-specific maps using Eq. (3.17). 75
3.12 An extended scheme for document detection based on the Dahu pseudo-distance. 76
3.13 Document detection from the max-tree. A document candidate tends to have a quadrilateral shape, also the top line is parallel with the bottom line (respectively with the left line and the right line). On the other hand, the document region is brighter in the saliency map. . . . 77

4.1 Comparison between saliency maps obtained using the vectorial Dahu pseudo-distance and using the Dahu pseudo-distance on separate channels. From top to down are four datasets: MSRA-10K, DUTOMRON, ECSSD, PASCAL-S. From left to right are three evaluation metrics: (a) Precision-recall curves, (b) F_β-measure, (c) Percentage curves. "Color" is the *color* saliency map computed using our vectorial Dahu pseudo-distance applied directly on color image, "Gray" is the saliency map obtained using the Dahu pseudo-distance computed on the grayscale image and "Combination" is the saliency map obtained by averaging saliency maps computed on separate red, green and blue channels. The three different measures show that our vectorial Dahu pseudo-distance leads to a much better saliency map. 82
4.2 Several saliency maps of the vectorial Dahu pseudo-distance on color images and the Dahu pseudo-distance on separate channels. Note that image (c) and (d) are respectively the *vivo* and *viso* Dahu pseudo-distances on the color image. The Dahu pseudo-distance on the color image highlights the object over the background, whereas, when only one channel is used, the saliency map only spots a part of the object. . 83
4.3 Different versions of saliency map deduced from Dahu pseudo-distances computed on the original color image (a) or the corresponding grayscale image (b); The saliency map when the seeds are the border pixels deduced from: the Dahu pseudo-distance (c) computed on the grayscale image and the vectorial Dahu pseudo-distance (d) computed directly on color image (with *vivo* (e) an additional color visualization of this latter); The saliency map when the seed is the center pixel deduced from: the Dahu pseudo-distance (f) computed on the grayscale image and the vectorial Dahu pseudo-distance (g) computed directly on color image (with *vivo* (h) an additional color visualization of this latter). 84
4.4 Comparison on color images of saliency maps deduced from our vectorial Dahu pseudo-distance on color images with saliency maps deduced from state-of-the-art methods. 86
4.5 An example image to investigate noise stability of the Dahu pseudo-distance and MB-based distance. The points p_1 and p_2 belong to the background, when p_3 is inside the object (this picture comes from the MSRA dataset (see [209])). 89

4.6 Stability of the inter- and intra-distances using the vectorial Dahu pseudo-distance or other MB-based methods against Gaussian noise. 90
4.7 Execution time (in milliseconds) to compute numerous distances between two points using the (pseudo-)distances presented in this thesis. 91

5.1 Shortest path finding in images. The input images and the end points (depicted in red) of the path we want to find are shown on each picture. Result are given for Dahu pseudo-distance, Waterflow-MBD and MST-MBD. Images are extracted from [246] and from [247]. 94
5.2 A scheme for object segmentation on multimodal/multispectral images. 95
5.3 White matter segmentation using the vectorial Dahu pseudo-distance. Images are taken from [248]. 96
5.4 Hyperspectral images 97
5.5 Interactive segmentation on synthetic images with taking into account the seed point positions and noise. 99
5.6 On the sensitivity to the number and position of seeds. 100
5.7 Interactive segmentation. 101
5.8 Failed Interactive segmentation. 102
5.9 The qualitative results of our extension method for interactive segmentation. The original images along with the scribbles are presented in column (a); (b) and (c) respectively represent the probability of every pixel in the image with regard to the background and foreground scribbles; the segmentation results are illustrated in column (d) which are close to the ground truth in column (e). 103
5.10 Comparison on interactive segmentation between our proposed method and Grabcut method [23]. 104
5.11 Image segmentation. 105
5.12 Comparison of our saliency maps with other classical or state-of-the-art methods. 107
5.13 Numerical comparison of saliency maps. 108
5.14 Some failure cases of the Dahu-based approach. 109
5.15 Quantitative results on Smartdoc 2015 competitions data. The red (resp. blue) color denotes the best (resp. second) result in each background. Our method gets the second highest overall score. It is competitive with the LRDE method [26], but about 20 times faster than their method. 110
5.16 Some qualitative results of our method. These images show the robustness of our method to illumination, blur and curled document. 111
5.17 The compromise between the executed time (image resolution i.e. image scale) and the overall accuracy. Even at low resolution, our method achieves an overall score of 0.962 for a run time equal to 0.65s. 111

List of Tables

1.1 Les résultats de la segmentation interactive xvi
1.2 Résultats quantitatifs sur les données des concours Smartdoc 2015. La couleur rouge (resp. bleue) indique le meilleur (resp. le second) résultat dans chaque cas. Notre méthode obtient la deuxième meilleure note globale. Elle est au même niveau que avec la méthode LRDE [26], mais environ 16 fois plus rapide que leur méthode. xviii

2.1 Algorithmic approaches to interactive segmentation. 36

4.1 Comparison between saliency maps obtained using the vectorial Dahu pseudo-distance and using the Dahu pseudo-distance on separate channels using $F_\beta{}^{max}$ measure and EMD score. "Color" is the *color* saliency map computed using our vectorial Dahu pseudo-distance applied directly on color image, "Gray" is the saliency map deduced from the Dahu pseudo-distance computed on the grayscale image, R, G and B are the saliency maps deduced from the Dahu pseudo-distance computed on each channel separately and "Combination" is the saliency map obtained by averaging the three saliency maps R, G and B. The best result is in bold form and the worst is in underlined. The three different measures show that our vectorial Dahu pseudo-distance leads to a much better saliency map. 81
4.2 Numerical comparison of saliency maps deduced from the vectorial Dahu pseudo-distance applied on color images and different MB-based distances adapted to manage color images. The comparison is performed using $\overline{F_\beta}$ measure and EMD score. Best scores are in bold. Results of all methods are comparable and variations among them are negligible. 87
4.3 A comparison of ratio of inter- and intra-distances between the Dahu pseudo-distance and other MB-based methods. 88

5.1 The segmentation results of the synthetic images (5.5). The percentage of incorrectly labelled pixels is presented in the form of the mean values and standard deviations. The best scores are in bold. 99
5.2 A comparison of interactive segmentation between our proposed method and several state-of-the-art methods. 103

Chapter 1

Introduction

Hierarchical image representations have been the major trends in recent years for segmentation and filtering tasks. They can be used to model the content of an image by the their structure. Because of the multi-scale representation properties, hierarchical image representations are able to capture the object of interest at various scales. Moreover, the topological relationship between objects in the image can be performed through the edges of the tree. In this context, we take an interest in studying such hierarchical representations as useful tools for several computer vision applications, including salient object detection and object segmentation.

In the field of Mathematical Morphology, there exist two types of hierarchical image representations with two different semantics: partition trees and trees based on threshold decomposition. In this thesis, we are particularly interested in the tree of shapes which belongs to the second class because of its properties. This kind of tree is self-dual and contrast-change invariant. Therefore, we use this structural representation to address a new approach, called distance transform to deal with the region-based representation.

Distance transform aims to measure the dissimilarity between pixels in an image. In our work, we focus on the path-wise distances in general and the minimum barrier distance in particular. This distance has been proved to be robust for noisy and blurred images. Unfortunately, its computation is expensive. To overcome this limitation, we proposed an efficient way to compute it thanks to the tree of shapes. This approximated distance is called the Dahu pseudo-distance. This thesis is dedicated to investigate the properties of the Dahu pseudo-distance and to apply this distance in several applications, such as salient object detection, shortest path finding, image segmentation and object detection. Specially, we employed this new distance for document detection in the videos captured by smartphones.

1.1 Image representation

A digital image is a two-dimensional signal which captures the information from cameras. An image can be considered to be a graph, based on the relation between adjacent pixels, which are the smallest elements in the image. The pixel-based representation, also called "*pixel adjacent graph*" (PAG), has been studied in the early stage of image processing [31].

Despite the simplicity of a PAG, the representation relied on the pixel level is not sufficient. It leads us to the higher level representations, region-based representations or "*region adjacent graph*" (RAG). In these representations, an image is modeled by a set of superpixels which are group of pixels that share similar properties. The RAG, which can be obtained by fine segmentation algorithms, takes into account the local context (regions) and global properties (spatial connection between

objects). The relation between adjacent regions in the image is expressed by the edge weights which represent the dissimilarity measures between them. Depending on the application, one can choose appropriate dissimilarity measures, for example color, texture or gradient value. The number of elements in region-based representation is less than the pixel-based one, consequently, it reduces the search space for object detection applications.

Objects can have various sizes and different positions in the image. Therefore, to adapt to many computer vision applications, we need to consider the multi-scale representation of the image. That gives rise to hierarchical representation of image, which is a set of connected components from fine to coarse level, called tree-based representation. Relying on the properties of the trees, we can categorize two types: hierarchical partition trees and threshold decomposition trees.

- The former representation begins with image partition algorithms. It fuses small regions to form a bigger one, thereby generating a set of partitions going from fine to coarse. It can be represented by a tree structure, in which the root node corresponds to an entire image while the leaf nodes represent initial regions in the fine segmentation. This type of representation can be found on several examples, in particular, Quadtrees [32], α-trees [33] and Binary partition trees [34]. These representations are used in various applications in computer vision [34–36].

- The latter representation is based on the threshold decomposition, which encodes the spatial inclusion relationship between connected components from different thresholded levels. The Min- and Max-trees [1, 2] and the Tree of shapes [3] are three typical trees of this kind of representation. Contrary to the partition trees, the trees based on threshold decomposition are contrast-invariant. Any cut in these trees generates a partial partition in the image. Additionally, the leaf nodes on these presentations correspond to the local extrema of the image. Trees based on threshold decomposition have been used in numerous applications, for instance, image filtering and segmentation [37, 38, 1, 39, 40, 26], video segmentation [41], image representation [18, 17], pattern recognition [42, 43], image registration [44] , image compression [2] and data visualization [45].

The above representations are application-driven and are generally used in image processing domains. Their multi-scale structures are able to capture different object regions in the image. There are many methods which have been proposed to deal with the tree. A popular approach, the tree simplification, is used to reduce the number of nodes in the tree which correspond to the small areas or do not contain meaningful information. The nodes on the tree can be removed or preserved depending on their attributes which are based on the size, contrast, shape or texture of the connected component. In addition, the parenthood relationships between parent nodes and their descendants are considered as well.

Another approach aims to search for the "*best cut*" that generates partitions for image segmentation. This method is based on a global optimization model of the energies computed on each node in the hierarchy. A partition which is generated from the best cut is a union of different nodes from different levels in the tree. The most well-known energy function is the Mumford-Shah function that is first proposed in [46] and used on the hierarchies in many researches such as [40, 47, 48].

Recently, many supervised-learning based methods are proposed to obtain the best cut in the hierarchy [49–51]. These methods rely on the low-level features, for examples, color, gradient and texture cue [52], and mid-level features, which are based on graph partition and Gestalt properties [51].

In this thesis, we address another approach, called distance transform, which is widely used to measure the dissimilarity between pixels in the image, thereby expressing the relations between objects and background in the image. This new distance function will be presented in Section 1.2.

1.2 Distance transform

Distance functions have been long studied in the mathematical morphology community, typically, in fundamental morphological operators such as erosion, dilation or skeleton application [53]. Recently, distance transform and the notion of saliency maps, which is deduced from the distance function, are generally used in image processing and computer vision [22, 9, 10, 8, 6]. In general, distance functions can be classified into two categories: point-wise and path-wise. Point-wise distances are computed relative to the domain of an image, while path-wise distances involve the topographical view of the image.

Here, we focus on path-wise distances, where an image can also be seen as a graph (the vertices are the pixels of the image and the edges are induced by the neighborhood relationship between these pixels). The usual method to find the path-wise distance between two pixels is thus to compute the length of the shortest path in the graph that goes from one of these pixels to the other one. The most used path-wise distance in image processing is the geodesic distance (see [4]). However, this distance is not robust enough to deal with noisy and blurred images. Lately, a new pseudo-distance, called minimum barrier distance (MBD) has been proposed in [5].

The minimum barrier distance is the minimum value of all the barrier "strengths" (a notion defined later) among the set of possible paths between two given points. This distance is first studied in [6] and in [7]. The MBD has many interesting theoretical properties and is an effective tool in image processing and computer vision applications, especially in salient object detection (see [9, 10, 8, 11–13]), interactive segmentation (see [14, 15]) and object localization (see [16]). Some works show that the minimum barrier distance outperforms the geodesic distance on noisy and blurred images (see [9, 5]).

Recently, the Dahu pseudo-distance has been introduced in a Mathematical Morphology fashion (see [19]) with the aim to approximate the MBD. This Dahu pseudo-distance is computed by considering an image as a landscape (we also speak about its topographical view). Different from the approaches of [9] and of [8] which compute the MBD directly in the image space, the Dahu pseudo-distance can be computed efficiently on a tree-based representation of the image (the tree of shapes). Thanks to this approach, the computation of the Dahu pseudo-distance is very fast.

Since this distance was initially developed for grayscale images, in this thesis, we propose an extension of this transform to multivariate images; we call it vectorial Dahu pseudo-distance. An efficient way to compute it is provided in the following chapters. Additionally, we demonstrate the robustness of the vectorial Dahu pseudo-distance compared to other MB-based distances across several benchmarks. This distance is efficient for salient object detection, and hierarchical image segmentation. A combination of these two applications which are derived from the Dahu

pseudo-distance is integrated in a full framework for document detection in videos captured by smartphones.

1.3 Main contributions

Our main contribution in this thesis is the proposition of a framework for document detection relying on the region-based representation, which applies the new MBD that belongs to the domain of mathematical morphology. Additionally, we investigate several interesting properties of the Dahu pseudo-distance:

- We propose an extension of the Dahu pseudo-distance to multivariate images, and we introduce a new way to compute it faster.
- We extend the Dahu pseudo-distance to a more "clever" version which combines tree-based and spatial representations to give better results (especially to find the shortest path between two points in the image space). Then we do not only look for the shortest path in a tree but also in the domain of the image using the geodesic distance.
- We explore the properties of the Dahu pseudo-distance via several experiments: we compare the vectorial Dahu pseudo-distance with the Dahu pseudo-distance computed on separate channels, we analyze the noise stability of the vectorial Dahu pseudo-distance, and we study the contrast of the Dahu pseudo-distance when computed on the ratio between inter- and intra-distances.
- We demonstrate the robustness of the vectorial Dahu pseudo-distance in some applications, such as salient object detection, interactive segmentation and shortest path finding. Many experiments confirm the improvement brought by the multivariate extension of the Dahu pseudo-distance over other common strategies using the MBD on color images.
- We introduce the multivariate Dahu pseudo-distance on multimodal/multispectral images and we provide experiments to validate the usability of this Dahu pseudo-distance on such kind of images.
- We propose an efficient method for interactive segmentation using the Dahu pseudo-distance through a consideration of the background and foreground information by employing a statistical approach.
- We propose a fast method for image segmentation based on our new distance. This method demonstrates again the robustness and effiency of the Dahu pseudo-distance in image processing application.
- We use the Dahu pseudo-distance on automatic segmentation of documents in smartphone photos or videos using the visual saliency approach.

1.4 Manuscript organization

This thesis is divided into four main parts. The **first part** presents the theoretical background that relates to our works including a review of hierarchical representations provided by the mathematical morphology community, tree simplification and segmentation, a review of traditional image segmentation methods, distance transformation and visual saliency detection.

- Section 2.1 discusses the digital image and review several classical methods to represent the image by covering some fundamental notations and definitions. These image representations are implemented through a concept that considers an image to be a graph which can be based on the pixels or the regions in the image.

- A review of the first type of hierarchical, namely the hierarchical partition trees is presented in Section 2.2. We recall several well-known structures such as the Quadtree [54], Minimum spanning tree [55], α-tree [33] and the Binary Partition Tree [34]. We also present the algorithms to build these representations from a digital image, the properties and the applications of each hierarchical partition tree.

- Section 2.3 introduces the other type of hierarchical representation of images, namely tree-based on the threshold decomposition. Three classical inclusion trees that we review in this section: the min- and max-tree [2], and the Tree of Shape [56]. We also review an extension of Tree of Shapes to multivariate images [17].

- In the previous sections, we discussed different types of tree-based representation. However, a tree may contain a great deal of nodes. Therefore, in Section 2.4, tree simplification is introduced to reduce the number of nodes in the tree. Several operators to simplify the tree are presented in [34, 57, 2, 58].

- We highlights the most popular methods for image segmentation in Section 2.5, which is used to partition a digital image into multiple meaningful segments. Image segmentation is used to simplify or change the image representation in order to analyse the image easier. Generally, this method is used as an intermediary step for many computer vision applications such as object detection and recognition.

- Some well-known distances are investigated in Section 2.6. Then, a recent distance called Dahu pseudo-distance is presented, which belongs to the mathematical morphology domain.

- Section 2.7 gives several visual saliency detection methods in both top-down and bottom-up methods.

The **second part** presents the main contributions of our work. We propose several approaches to enrich the properties of the Dahu pseudo-distance. This chapter is divided into two sections: Dahu pseudo-distance improvements and applications.

- In Section 3.1, we present a method to efficiently compute the Dahu saliency map by using a tree-based structure. Additionally, we extend the Dahu pseudo-distance to multivariate images, and present a method to compute it efficiently based on the tree of shapes.

- In Section 3.2, we propose several frameworks based on the Dahu pseudo-distance to solve many problems in image processing and computer vision, for example, the shortest path finding, salient objects detection, interactive segmentation and image segmentation. These methods are integrated together into a document detection application.

In the **third part**, we explore the properties of the vectorial Dahu pseudo-distance via several experiments in the sense of visual saliency detection. Furthermore, some experiments are implemented to analyze the stability w.r.t noise of the vectorial Dahu pseudo-distance and MB-based distances, and the contrast of the Dahu pseudo-distance based on the ratio between inter-distance and intra-distance. The inter-distance is the distance from a marker outside the object to a marker inside the object and the intra-distance is the distance between two markers in the same object. Lastly, we provide a comparison between the executed time of the Dahu pseudo-distance and some MB-based distances.

- Section 4.1 demonstrates the robustness of the vectorial Dahu pseudo-distance. In this section, we implement multiple experiments with the vectorial Dahu pseudo-distance. Firstly, we compare the vectorial Dahu pseudo-distance with the Dahu pseudo-distance on separate channels. Thereafter, we compare our new distance with state-of-the-art MB-based distances.
- We investigate the contrast of the Dahu pseudo-distance based on the ratio between inter-distance and intra-distance and analyze the stability of the vectorial Dahu pseudo-distance w.r.t noise in Section 4.2.
- Section 4.3 compares the speed performance of the Dahu pseudo-distance with some state-of-the-art MB-based distances.

In the **fourth part**, we validate the efficiency of the Dahu pseudo-distance in many applications. The first application is the shortest path finding between two pixels in the image. Next, we demonstrate the great potentiality of the Dahu pseudo-distance in multimodal medical images, consequently, confirm the usability of the Dahu pseudo-distance in multivariate images. Another application to endorse the ability of the Dahu pseudo-distance on multivariate images is multi-spectral imaging. Finally, the Dahu pseudo-distance is exploited to detect a document in the videos captured by smartphones.

- In Section 5.1, an application for finding the shortest path between two chosen markers of the Dahu pseudo-distance in the image is explored. We compare the shortest path between our distance and other MB-based distances.
- Section 5.2 presents the application of the vectorial Dahu pseudo-distance on the multi-modality medical imagery and multispectral satellite imagery. We use the same strategy to deal with the multimodal and multispectral images.
- We examine the stability of the Dahu pseudo-distance w.r.t marker positions for object segmentation in Section 5.3. First, we investigate the Dahu pseudo-distance stability, and how this stability influences the results of distance-based image segmentation. Secondly, we study the inter-dependence of the number of distance seeds on interactive segmentation in natural images. Thirdly, we apply a simple method using the Dahu pseudo-distance in interactive segmentation on the Gulshan dataset [59]. Finally, we evaluate our improving model for interactive segmentation by comparing to other state-of-the-art approaches.
- In Section 5.4, we apply the proposed fast method for image segmentation based on the Dahu pseudo-distance. Our approach belongs to the hierarchical image segmentation class.

- Section 5.5 validates the usability of the Dahu pseudo-distance for automatic segmentation of documents in smartphone photos or videos using visual saliency (VS). In the first part, we compare our method with different VS methods. We show that our saliency maps are competitive with state-of-the-art visual saliency methods, and that such approach is very promising for use in identity document detection and segmentation, even without taking into account prior knowledge about document contents. In the second section, we evaluate our extended version which is based on the hierarchical image segmentation. Our method is able to accurately segment the document region at high speed.

Chapter 6 concludes this dissertation. We present a quick review of our method to extend the Dahu pseudo-distance to multivariate images. We also introduce several improvements and advantages of our proposed methods to deal with many applications in image processing. Finally, we discuss some limitations of the Dahu pseudo-distance and open some directions for future works.

Chapter 2

Theoretical Background

2.1 Image Representation

In this chapter, we talk about the digital image and review several classical image representations by covering some fundamental notations and definitions. These image representations are implemented through a concept that considers an image to be a graph. The representation can be based on the pixels or the regions in the image. They are efficient and used widely in different applications in image processing.

2.1.1 Digital image

An image is a two-dimensional plane which is captured from the optical devices, in particular, the cameras. The captured continuous image is then digitized to an array of digital numbers, thereby representing the data in both coordinates (sampling) and amplitude (quantization). Therefore, an image can be regarded as well a two-dimensional function $f(x, y)$, $\mathbb{Z}^2 \rightarrow \mathbb{N}$, where x and y indicate the spatial coordinates, and the amplitude f at any pair of coordinates (x, y) represents the intensity value of that point [60]. The smallest representation of the image is the pixel (picture element), which is stored and displayed in the grid map or the matrix [61].

We present in the following several notations, which are used in this thesis including pixel connectivity, path between pixels, connected component and regions in the image.

Pixel p in the image is linked to its neighbors by some classical pixel adjacency relations. For example, a pixel p at coordinates (x, y) has four horizontal and vertical neighbors whose coordinates are given by: $(x+1, y), (x-1, y), (x, y+1), (x, y-1)$.

A path $\pi(p_0, p_n)$ from p_0 to p_n is a sequence of pixels $\langle p_0 ..., p_i, p_{i+1} ... p_n \rangle$, where p_i and p_j are two adjacent neighbors.

Let call $\{S\}$, the set of all pixels in the image, and $\{S_X\}$, a subset of pixels so that, $\{S_X\} \subseteq \{S\}$. Two pixels p and q are said to be connected in $\{S_X\}$ if there exists a path of pixels in $\{S_X\}$ that connects p and q. For any pixel p in $\{S_X\}$, the set of pixels that are connected to p in $\{S_X\}$ is called a connected component of $\{S_X\}$. If there is only one connected component, then set $\{S_X\}$ is called a connected set [60].

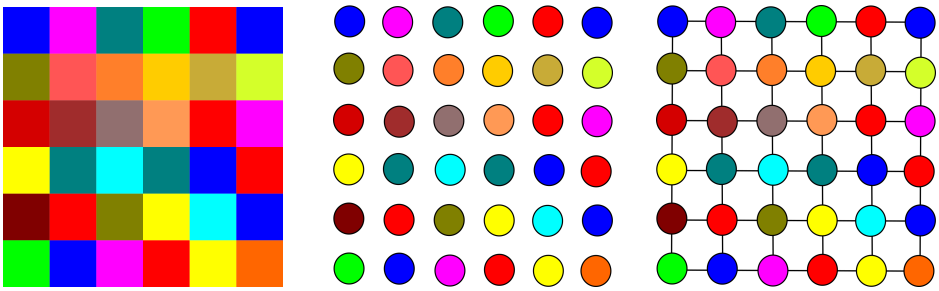
We call a subset of pixels $\{R\}$, the region in the image if $\{R\}$ is a connected set. Two regions R_i and R_j are called adjacent if their union forms a connected set. Regions that are not adjacent are said to be disjoint ($R_i \cap R_j = \emptyset$). A boundary of region $\{R\}$ is a set of pixels which are adjacent to pixels that belong to the complement of region $\{R_c\}$ (set of pixels that are not in $\{R\}$) [60].

2.1.2 Classical Image Representation

In the previous section, we recalled the definition of the digital image, which can be examined as a matrix of pixels. The matrix theory is used for many operations between images. Here, we discuss other classical image representations such as pixel-based representation, graph-based representation and hierarchical representation. The kind of image representation is chosen depending on the application.

2.1.2.1 Pixel-based representation

A graph is a good way to represent a set of data where some pairs of data are connected to each other [62]. This representation is intuitively based on the fact that a pixel is a unit element of the image. The pixel-based representation is also called the "*pixel adjacent graph*" (PAG).



(a) Image. (b) Nodes. (c) Nodes and edges.

FIGURE 2.1: Graph of pixels, where nodes represent pixels in the image and edges correspond to connectivities between adjacent pixels.

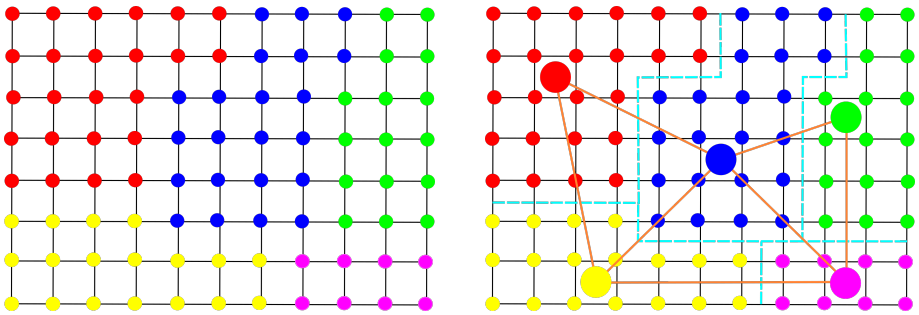
In image processing, an image domain is associated with a graph of pixels, which is defined as $G(V, E)$, where vertices V represent pixels in the image, and E is the set of edges. Each edge E_{ij} joins two pixels v_i and v_j. The first example of dealing with this graph is defining the relation between neighboring pixels. 4- and 8-connectivity in 2D are defined in [63]. Although these two connectivities are popular, their topological properties do not hold the Jordan curve theorem, which states that a closed curve separates a 2D space into two regions (exterior and interior) [64]. The Khalimsky plane is proposed in [65] to solve this problem. An example of the pixel-based representation is illustrated as Fig. 2.1.

The PAG is also used to segment an image into regions [31]. In this paper, the authors proposed a method to measure the difference between regions, thereby producing satisfying image segmentation results. The graph is constructed by considering the edge weights, which express the similarities between neighboring pixels. In particular, the edge weight is the absolute intensity difference between the pixels connected by an edge. The complexity of graph algorithms mostly depends on the number of edges and the number of vertices in the graph.

Despite the simplicity of this representation, there is a large number of elements to be examined. In addition, applications in image processing and computer vision are dramatically increased in recent decades. The representation relied on the pixel level is not sufficient. Therefore, to better analyse the scene, we need to use the higher representation.

2.1.2.2 Region-based representation

In region-based representation, an image is modeled by a set of regions or set of superpixels, where each superpixel is a group of neighboring pixels that share some similar characteristics. A region is more intuitive, easier for visualization, and contains more information than an individual pixel. A region-based representation can be obtained by segmentation algorithms that partition an image into different regions upon specific similarity criterion. Besides, this representation also takes into account the spatial information and the dependence of neighboring pixels.



(a) A fine partion of the image.

(b) Region adjacency graph of the fine partition

FIGURE 2.2: Region adjacency graph.

This kind of representation is called "*region adjacency graph*" (RAG). An example of RAG is illustrated in Fig. 2.2. In this graph, the nodes represent regions in the image, and edges connect adjacent regions. The regions are generated by using a fine segmentation or unsupervised clustering methods such as superpixel segmentation [25], DBSCAN [66] or watershed transform [67, 68]. Since this fine segmentation is an initial step of the graph-based representation, the quality of the superpixel segmentation is essential. In particular, the fine segmentation has to contain meaningful superpixels, whose boundaries should appear on the contour of the object. Besides, the fine segmentation has to capture small objects in the image. The number of regions is reduced compared to pixel-based representations, while the representation accuracy can be kept [34]. The edge weights in this graph indicate the dissimilarities between adjacent regions in the image. The dissimilarity can be computed as the difference between color, texture or average gradient value. In addition, it should bare in mind both the spatial and the value sense [69]. The choice of the dissimilarity is especially important and depends on the application.

As discussed in [52], image segmentation is still a big problem in image processing. The objects in the image can appeared in various sizes. Therefore, a good segmentation method should consider the multi-scale representation of the image. To overcome this problem, we need to build a hierarchical representation, which is a collection of segmentation from fine to coarse level. That leads us to the definition of the hierarchical representation of image.

Based on the properties of the nodes and parenthood relationship, we can classify them into two types of the tree-based representation: Trees based on the threshold decomposition of the image and partition trees, which are known as hierarchies of segmentation.

- **Trees based on the threshold decomposition**: In this representation, a tree node denotes a particular connected component of the image level sets and

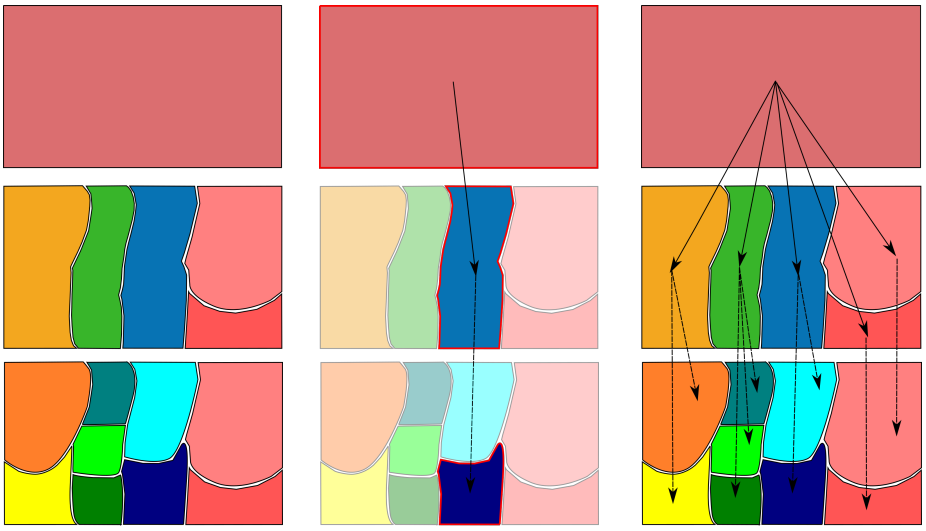
parenthood between nodes maps the relationship of spatial inclusion between components at different levels [43]. In general, any cut of this tree forms a partial partition of the image. Specifically, Max- and Min-tree [70], and Tree of Shapes (ToS) [18] are three typical trees of this representation.

- **Partitioning trees** are initialized from an image partition. They merge regions from a finer scale to form a bigger region in coarser scale. Any cut in this representation yields an image partition. Some typical examples of this representation are Binary Partition Trees (BPT) [34], Minimum Spanning Tree (MST) [55], Quadtree [54], and alpha-tree [33].

Contrary to the hierarchical partition tree, the union of leaf nodes of the tree based on the threshold decomposition does not cover the whole image. Instead, they represent the local extrema of the image. This type of tree will be discussed in Section 2.3, while the partition tree is the topic of Section 2.2.

2.2 Hierarchical Partition Trees

In this section, we recall several well-known hierarchical partition trees, the algorithms to build them from a digital image, the properties and the applications deduced from hierarchical partition trees.



(a) Set of partitions. (b) Inclusion relationship. (c) Dendrogram.

FIGURE 2.3: An illustration of the hierarchy of image.

A hierarchical partition tree is a chain of image partitions from fine levels to coarse levels. It is similar to the tree structure, in which the leaf nodes correspond to the initial image partition, and the root node represents the whole image. The initial image partition can be the result of any image segmentation, such as flat-zones of the image [36] or superpixels segmentation [25].

Let H be a set of partitions P_i. An example of the chain of image partition is illustrated in Fig. 2.3(a). We denote H being a hierarchy of partition if it satisfies the inclusion order condition:

$$0 \leq i \leq n, \forall j,k, 0 \leq j \leq k \leq n \Rightarrow P_j \subseteq P_k \tag{2.1}$$

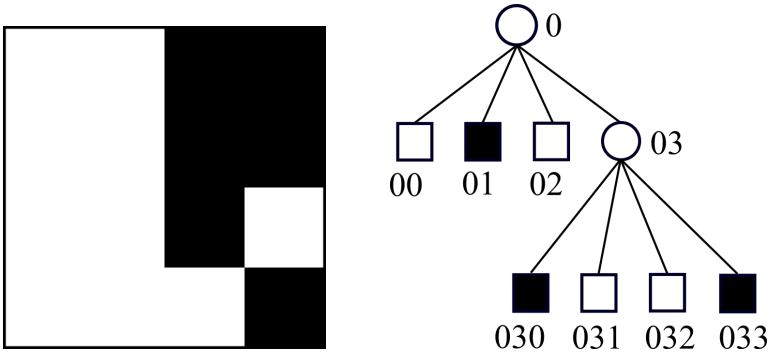
where P_0 is the finest partition and P_n corresponds to the whole image. A partition P_j is finer than the partition P_k if all the regions of P_j are included in the partition P_k. This property is depicted in Fig. 2.3(b).

Besides, the intermediary node is created by merging regions from its finer level. In other words, the hierarchy is constructed by an iterative merging algorithm until they remain only one region in the image. Therefore, a dendrogram is usually used to describe hierarchical partition trees. Fig. 2.3(c) illustrates one example of the dendrogram of the image. One advantage of these representations is that it is a multi-scale representation of the image. Therefore, it is able to cover variable-size objects in the image. It leads to the fact that the hierarchy of segmentation is compatible with object detection and segmentation application. Another advantage of this hierarchy is reducing the number of elements in the image; in other words reducing the search space for candidate regions.

There exist different hierarchies of partition to adapt to various applications. However, fully exploiting the properties of hierarchical representation, incorporating multiple information (context, color, texture, gradient, etc) from the regions in the partition, and also finding a best meaningful segmentation from the hierarchy are still big questions in the image processing community. "There is no such thing as a free lunch". Otherwise speaking, there is no such thing as the best hierarchy image representation for all the applications. That is why in the following sections, we review several famous partition trees.

2.2.1 Quadtree

This section is a review of the Quadtree, which is used in the early stage of the image processing history [54]. It is a spatial data structure that allows representing the image content. Leaf nodes on the Quadtree represent areas in the image. A Quadtree is constructed in a recursive way to cut the regions in the image into quadrants relied on a given split criterion. Therefore, a Quadtree is a hierarchical representation of different levels of resolution [32]. Each node on the tree can be a rectangle or a square region along with a specific color, which is the dominant color that has a higher percentage of occurrences inside the region. A parent node always has four descendants named respectively 1, 2, 3 and 4. The color of all sibling nodes which have the same parent should be different.



(a) Image. (b) The Quadtree.

FIGURE 2.4: An image and a its Quadtree representation.

The constructed algorithm of the Quadtree usually begins at the root node, which represents the whole initiate image. The next step is choosing the split criterion,

which can be color or texture homogeneity or the number of feature points in the quadrant. If the parent node does not satisfy the given split criterion, then four descendant nodes are divided from the parent node. The algorithm is repeated until every leaf node on the tree conforms the criterion.

Quadtree has been used widely in the 1990's in various applications such as content-based image retrieval [71–75], image compression [76–81], graphic [82], and image segmentation [83]. Especially in the image segmentation application, several sibling nodes may be merged to form a larger region. However, this process does not keep Quadtree properties anymore. Fig. 2.4 presents an image and its corresponding Quadtree representation. Index 0 identifies the root node that represents the whole image. Whereas, the indices 1, 2, 3 and 4 respectively correspond with four descendant of each parent node.

2.2.2 Minimum spanning tree

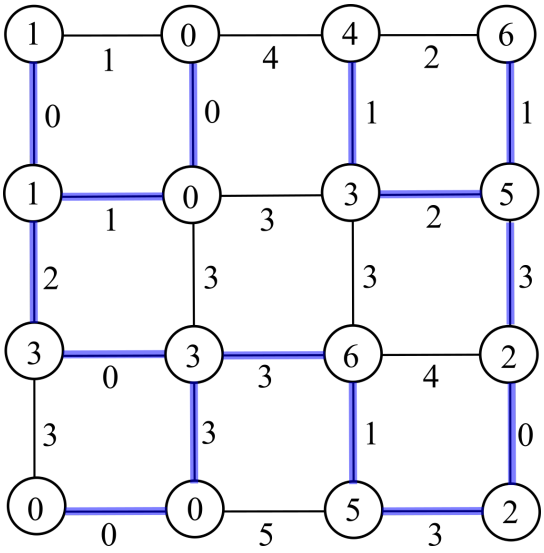


FIGURE 2.5: An image and its MST representation.

Although the minimum spanning tree does not belong to the tree-based representation of the image, it is used in many image processing applications. Given an edge-weighted connected graph $G(V, E, w)$, a spanning tree is a subset of a graph G, which contains all the vertices from G covered with the minimum possible number of edges. Hence, a spanning tree does not have cycles, and it cannot be disconnected. The minimum spanning tree $MST(G)$ is then a spanning tree whose sum of edge weights is as small as possible.

$$MST(G) = \underset{T \in ST}{argmin}\Big(\sum_{e_{ij} \in E_T} w(e_{ij})\Big) \tag{2.2}$$

where ST is a set of all spanning trees of G.

In general, there may exist several MSTs of a graph. In the particular case where all edge weights are different, the MST is unique. An example of MST is given in Fig. 2.5. The MST is illustrated by the blue lines in this figure.

Several algorithms are proposed to compute MST for undirected graph. Here, we present three popular methods, including Boruvka's algorithm [84], Prim's algorithm [85] and Kruskal's algorithm [55]. All of them are greedy algorithms.

Algorithm 1: Boruvka's algorithm to compute MST.

Data: A graph $G(V, E, w)$
Result: A $MST(G)$
Initialize all vertices as individual components (or sets) ;
Set $MST = \{\}$;
while *There are more than one components* **do**
    Consider next component ;
    Find the closest weight edge that connects this component to any other component ;
    Add this closest edge to MST if not already added;
end

The study on constructing an exact MST starts with Boruvka's algorithm [84]. This algorithm begins with each vertex of a graph being a tree. Then for each tree, it iteratively selects the shortest edge connecting the tree to the rest and combines the edge into the forest formed by all the trees, until the forest is connected. The computational complexity of this algorithm is $O(ElogV)$, where E is the number of edges, and V is the number of vertices in the graph. Boruvka's algorithm is illustrated in Algo. 1.

Algorithm 2: Prim's algorithm to compute MST.

Data: A graph $G(V, E, w)$
Result: A $MST(G)$
Choose arbitrarily a start node s ;
Set $MST = \{\}$, $S = \{s\}$;
while $S \neq V$ **do**
    Find an edge e such that:
    - e starts in S and ends out of S
    - e has the minimal weight of edges

    Add e to MST ;
    Add the vertex at the end of e to S ;
end

One of the most typical examples is Prim's algorithm, which was proposed by [85]. It starts with an empty spanning tree. The main idea is maintaining two sets of vertices: the first one contains vertices that have been already included in the MST, while the second one is not. It first arbitrarily selects a vertex as a tree, and then repeatedly adds the shortest edge that connects a new vertex to the tree, until all the vertices are included. The time complexity of Prim's algorithm is $O(ElogV)$. If the Fibonacci heap is employed for finding the shortest edge, the computational time is reduced to $O(E + VlogV)$ [86, 87]. Prim's algorithm is given in Algo. 2.

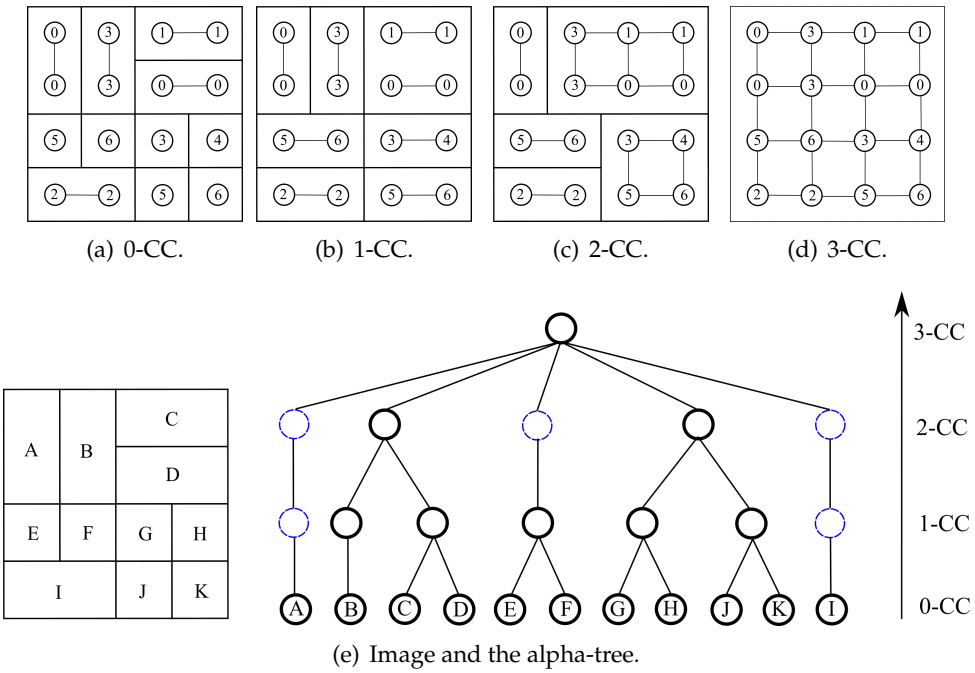
Algorithm 3: Kruskal's algorithm to compute MST.

Data: A graph $G(V, E, w)$
Result: A $MST(G)$
Initially, sort edges in ascending order of weight;
Set $MST = \{\}$;
for *each edge* $e \in E$ **do**
  if $\tilde{G} = (V, MST \cup \{e\})$ *does not contain a cycle then* **then**
    Add e to the MST;
  end
end

Kruskal's algorithm is one of the most used algorithms to construct the MST [55]. In this algorithm, all the edges are sorted by their weights in non-decreasing order. It starts with each vertex being a tree and iteratively combines the trees by adding edges in the sorted order, excluding those leading to a cycle until all the trees are combined into one tree. The running time of Kruskal's algorithm is $O(ElogV)$. Algo. 2 describes the Kruskal's algorithm to compute the MST.

The MST is a well-known problem in graph theory and has been broadly applied in many image processing and computer vision applications. For example, the MST is adopted in image segmentation [88–92], images analysis [93, 94] cluster analysis [95–98], classification [99], density estimation [100], and salient object detection [10].

2.2.3 α-Tree



(a) 0-CC. (b) 1-CC. (c) 2-CC. (d) 3-CC.

(e) Image and the alpha-tree.

FIGURE 2.6: An example of an α-tree.

The α-tree [33] is a multiscale representation of the image, also known as quasi-flat zone hierarchy [101]. This tree is based on the α-connectivity. To easily understand the α-tree, we begin with the notion of the flat zone in the image. In a digital

image, flat zones $0-CC$ are defined as connected sets of pixels sharing the same value:

$$0-CC(x)=\{x\}\cup\{y|\exists\pi(x,y):\\ \forall x_i\in\pi(x,y)\wedge x_i\neq y\Rightarrow d(x_i,x_{i+1})=0\} \quad (2.3)$$

However, the flat zone definition has a problem because of its extreme over-segmentation. To overcome this limitation, the quasi-flat zone definition is proposed in [101]. The most widely used definition of quasi-flat zones is called α-zone, that leads to the definition of the α-connectivity.

$$\alpha-CC(x)=\{x\}\cup\{y|\exists\pi(x,y):\\ \forall x_i\in\pi(x,y)\wedge x_i\neq y\Rightarrow d(x_i,x_{i+1})\leq\alpha\} \quad (2.4)$$

The higher the value of α is, the larger the quasi-zone $\alpha-CC(x)$ is. The $\alpha-CC(x)$ may be merged from two lower α-zones. Therefore, we can get a hierarchy representation from this chain or partitions.

However, the simple α-zone definition also has a drawback. In the case of a low gradient or blur image, this quasi-flat zone may merge different regions to the same node in the tree. To deal with this problem, an additional ω parameter is proposed in [102] to control the growth of the quasi-flat zone. The ω parameter is defined as the difference value between the maximum and minimum value in the quasi-flat zone. The (α,ω)-$CC(p)$ is defined as:

$$(\alpha,\omega)-CC(p)=\vee\{\alpha_i-CC(p)|\alpha_i\leq\alpha,R(\alpha_i-CC(p))\leq\omega\} \quad (2.5)$$

where $R(\alpha_i-CC(p))$ denotes the maximal dissimilarity within α-$CC(p)$.

An example of the α-tree is given in Fig. 2.6. In this figure, the α-level between two adjacent regions is defined as the minimum edge weight that connects these two regions. Other edges connecting these regions are not important. This property is similar to the MST construction, in particular, the Kruskal's algorithm. For that reason, Kruskal's algorithm allows an effective way to compute the α-tree [103]. The α-tree is applied as well to some applications such as image simplification and segmentation [104, 105], object detection [106], and hyperspectral images [102].

2.2.4 Binary Partition Tree

The Binary Partition Tree (BPT) [34] is a hierarchical representation of an image based on the similarity between adjacent regions. The root node represents the entire image. The leaf nodes in BPT are regions in the initial partitions, which are able to capture small objects in the image, thereby describing the very local information of the image. On the other hand, the nodes, which are closed to the root node, contain the global description as they correspond to large regions in the image. As previously said, classical object detection methods use an exhaustive search algorithm to scan all possible candidate objects in the image concerning positions and scales. Because of its structural properties, the BPT is a good way to reduce the search space for object detection.

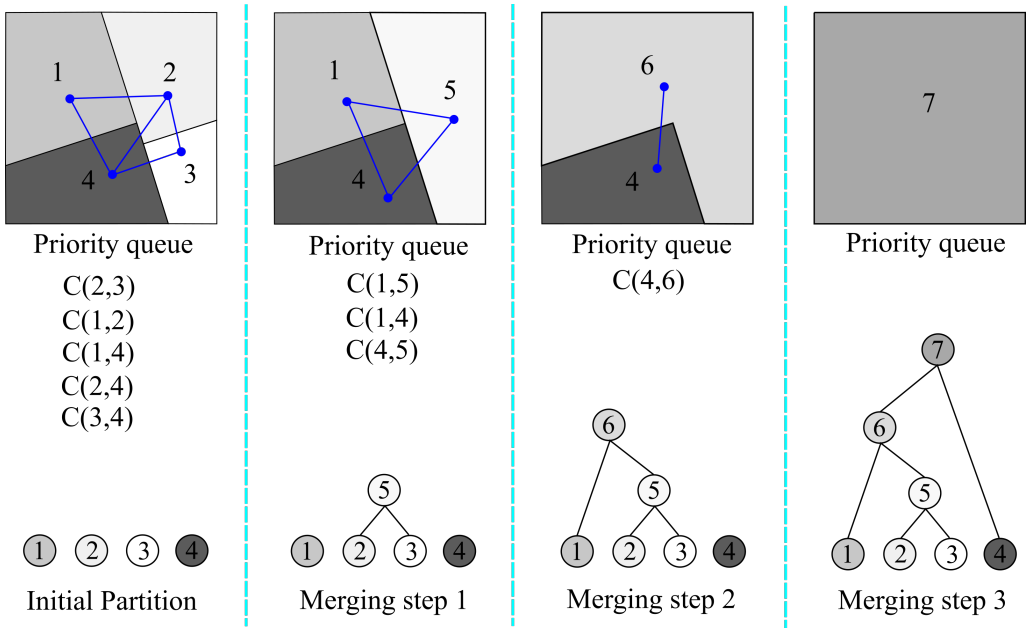


FIGURE 2.7: Example of BPT construction using a region merging algorithm by a priority queue.

The BPT reflects the similarity between neighboring regions [36]. The BPT construction is usually based on a recursive algorithm. It starts from an initial partition of the image (pixels, flat zones or over-segmentation algorithm). At each step, the similarity scores between all adjacent regions are calculated, and only the most similar pair of regions will be merged (thus the hierarchy is a binary tree). A priority queue is used to track the weight between adjacent regions in the image. After merging two regions, we renew the information of the parent node and update the weights between the parent node and its the adjacent regions. The merging procedure repeats until only one component left. The BPT is obtained by keeping track of this merging process. Therefore, the merging criterion plays an important role to construct this tree. It defines the similarity between regions and determines the order where regions are going to be merged. The choice of merging criterion depends on the application. An example of the BPT construction is illustrated in Fig. 2.7.

BPT is used in various applications in image segmentation [34, 35], and object detection [36, 107]. It has been extended to remote sensing domain such as hyperspectral and SAR images [57, 108–112].

2.2.5 Conclusion

In this chapter, we review the first class of tree-based representation in the mathematical morphology domain: hierarchical partition trees. We do not enumerate all types of the hierarchical partition tree, but instead present several popular classical approaches: Quadtree, Minimum Spanning Tree (MST), α-tree, and Binary Partition Tree (BPT). Hierarchical partition trees are multi-scale representations of images. In other words, they provide both the local and global descriptions of the images. The constructions of the trees are based on the similarity between regions in the image. Therefore, the merging criterion plays an important role in the process. The hierarchical partition trees also bare in mind the adjacency between adjacent regions. A cut from this tree generates a partition in the segmenting image. In the next section, we

present the second class of the tree-based representation, namely tree-based on the threshold decomposition, which involves the spatial inclusion relationship instead of the adjacency.

2.3 Tree based on the threshold decomposition

In this chapter, we discuss the trees based on the threshold decomposition. A node in these representations corresponds to a connected component of the image level sets. These connected components are arranged into the tree thanks to the inclusion relationship. The three typical threshold decomposition trees are the Min- and Max-tree, which are first presented in [1], and the tree of shapes, which is a self-dual representation of an image can be seen as a "merge" of these two above trees [113]. The Min- and Max-tree are dual in the sense that the Min-tree of an image is the Max-tree of its complementary and vice versa. They are constructed so that their leaf nodes orient toward the extrema in the image. These trees will be presented in Section 2.3.1. Otherwise, the tree of shape, which is also called topographic map, will be presented in Section 2.3.2.

2.3.1 Min Tree and Max Tree

We start with the Min- and Max-tree, which are the simplest threshold decomposition trees. A connected component in the gray-level image is defined as a connected set of pixels, which is obtained by using the notion of threshold decomposition [1]. The Min- and Max-tree are then derived from this component tree [2]. The root nodes in the Min- and Max-tree represent for the whole image, while the leaf nodes correspond with the minima, respectively maxima in the image. Then the inclusion relationship is used to express the link between nodes and their parents. These connected components do not create any new contour so that it is appropriate with filtering application [2].

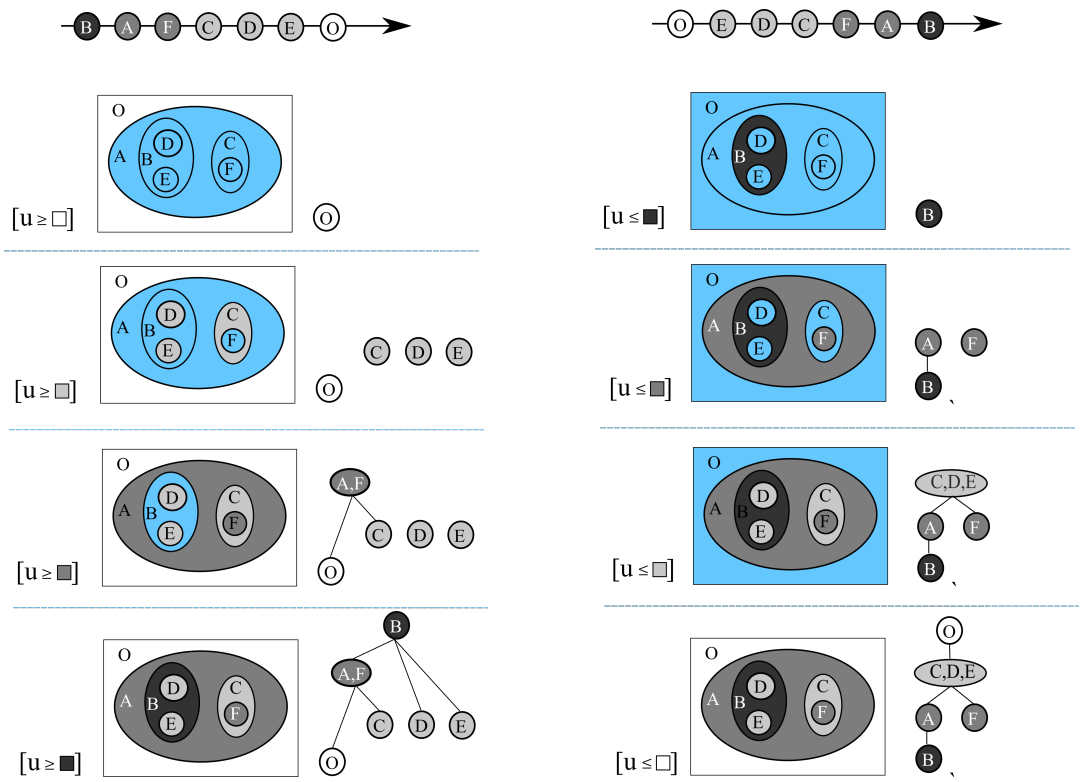
An image u is defined as a function: $X \rightarrow \mathbb{N}$. With a value $\lambda \in \mathbb{N}$, the upper and lower level sets (cuts) are defined as $[u \geq \lambda] = \{x \in X | u(x) \geq \lambda\}$ and $[u < \lambda] = \{x \in X | u(x) < \lambda\}$. We denote CC as the set of connected components corresponding with the lower and upper cuts of u. The Max-tree $T_{\geq}(u)$ and Min-tree $T_{<}(u)$ are then deduced respectively from these sets of connected components as $T_{\geq}(u) = \{\Gamma \in CC([u \geq \lambda])\}_{\lambda}$ and $T_{<}(u) = \{\Gamma \in CC([u < \lambda])\}_{\lambda}$. The Max- and Min-tree represent the inclusion relationship between the connected components at different levels of λ.

In [114], several algorithms to construct the Max/Min-tree have been presented, including the flooding algorithms [2, 115–117], the immersion algorithms [70], and the merge-based algorithms [118, 119]. In this thesis, we only focus on the second algorithm, which is a two-step procedure based on Tarjan's union-find algorithm [120]. An example of the Min/Max tree is illustrated in Fig. 2.8.

Various applications deduced from the Max/Min-tree have been proposed, for example image representation [2, 1, 70], image filtering and segmentation [121, 2], pattern recognition [122, 43] and remote sensing [34].

2.3.2 Tree of shapes

The tree of shapes (ToS) [56, 123] is a self-dual representation of an image, which is obtained by merging from the min-tree and the max-tree. It is self-dual because it



(a) Max-tree. (b) Min-tree.

FIGURE 2.8: Tree computation of the Max- and Min-tree based on the immersion algorithms (2-steps procedure). The result of the sorting step is given over the arrow, and the tree is constructed in the inverse order.

does not care about the contrast of objects (the dark object inside the light region or vice versa), thereby eliminating the redundancy of information contained in those trees. The tree of shapes is a decomposition of gray-level images into connected components, called shapes, which can be arranged into a tree under the inclusion relationship. A shape is a filled-in connected component (cavity-fill-in) without hole inside (its boundary is then an iso-level line). A cavity of a set $S \in X$ (X is the image domain) is called a "hole"; it is a connected component of $X \backslash S$ which is not the "exterior" of S [18]. With the cavity-filling (or saturation) operator denoted by Sat, and CC, the set of connected components (the tree of shapes) is defined as: $\mathfrak{S}(u) = \{Sat(\Gamma); \Gamma \in CC([u < \lambda]) \cup CC([u \geq \lambda])\}_\lambda$. Two iso-level lines (at different levels or not) can not cross each other (under some particular constraints). A very strong consequence is that shapes are either disjoint or nested, which explains that the tree of shapes is a tree and not a graph with cycles.

The first ToS construction algorithm has been proposed by Monasse et al. [113], called Fast Level Line Transform (FLLT), computes and merges the Min- and Max-tree. Its extended version, called Fast Level Set Transform (FLST) is presented in [123] relying on a region-growing approach. In [124], Song et al. have proposed a top-down approach by tracking the level lines starting from the border. A simple and efficient method to compute the ToS has been proposed by Géraud et al. [18]. Their method turns a ToS computation problem into a Max-tree computation problem. It is based on the Union-Find algorithm that computes the ToS in quasi-linear time

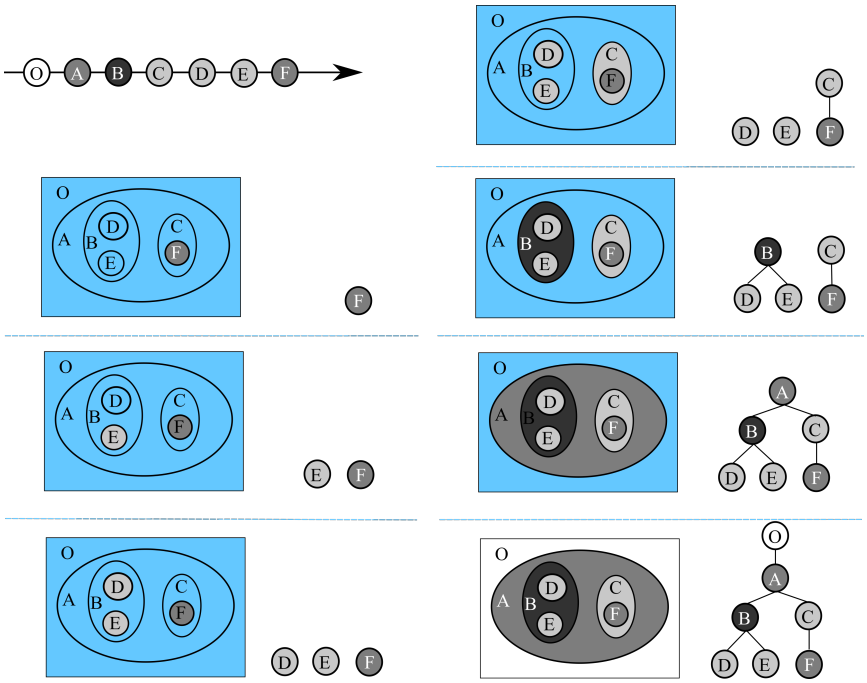


FIGURE 2.9: The computation of the tree of shapes. The result of the sorting step is given over the arrow.

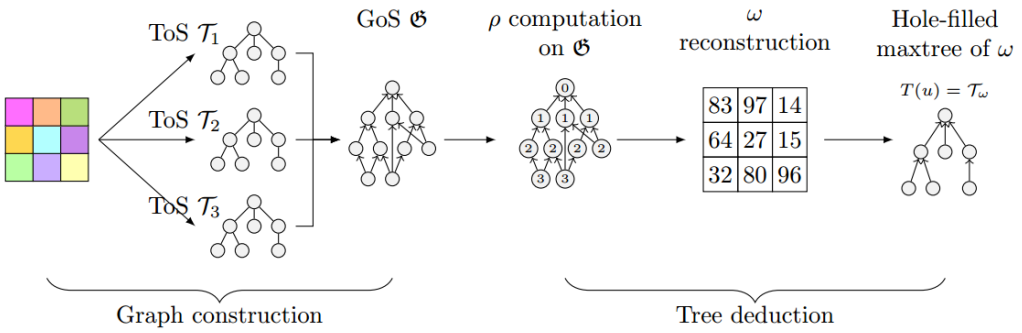
in four steps: interpolation, immersion, pixel sorting and tree construction. This method transforms an image to an interval-valued map in Khalimsky grid. A parallel version of this approach has been presented in [125]. The construction procedure of the ToS is depicted in Fig. 2.9.

This representation which is invariant to contrast changes and to contrast inversion, has been proved very useful in image processing and pattern recognition tasks such as image segmentation [126, 39, 40, 127, 26], object detection [42], remote sensing [128] and salient object detection [129].

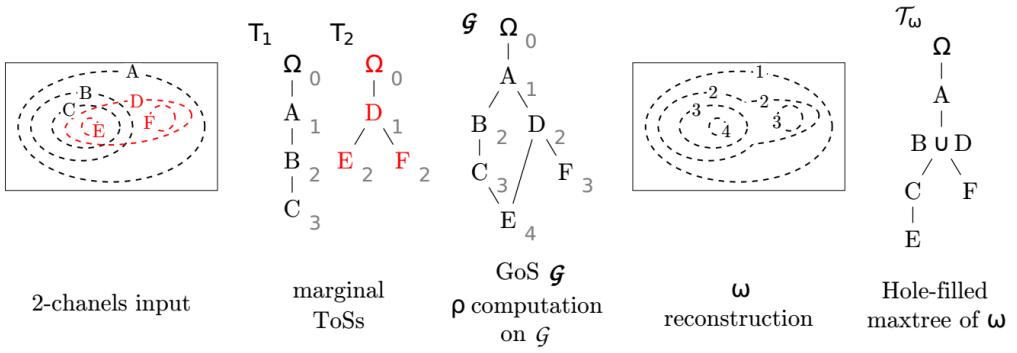
2.3.3 Multivariate Tree of shapes

As we mentioned before, the tree of shapes is only defined in the gray-scale images [18]. To compute the tree of shapes of the multivariate images, it is getting more complicated. In the case of partition trees, their computations are based on the distances which define the dissimilarity between pixels or regions, so that there is no challenge to extend those hierarchies to multivariate images. On the other hand, the ordering relation of values in the tree of shapes has to be total, otherwise the connected components may overlap, and the inclusion condition does not hold. To deal with the multivariate images, two methods exist. The first one constructs the ToS on each marginal image channel separately as considering each channel is an independent signal. This method has a limitation because, at the end of the tree construction, we have to handle several trees. Moreover, it does not consider the relationship between different channels in the multivariate image [130].

In this thesis, we only focus on the second method, which is based on vectorial processing. It defines an ordering on the vectorial value space. An advantage of this approach is that it only has one structure to process. A survey of the computation on the multivariate image is presented in [131]. In [17], the authors proposed a



(a) Computation steps of MToS.



(b) An example of MToS.

FIGURE 2.10: The 5-steps process for computing the MToS. Images are extracted from [130].

new approach to deal with multivariate images. Instead of building a total order, they rely on the inclusion relationship between components on the marginal tree of shapes. Their algorithm is a 5-steps procedure based on two main parts. The first part is the construction of a graph of shapes (GoS) from the set of ToS's, which is computed from each marginal image channel. The second one is the deduction of a single multivariate tree of shapes (MToS) from the GoS based on the computation of attributes on the GoS. The whole process is illustrated in Fig. 2.10. The details of the algorithm are presented as follows.

Contruction a GoS

The algorithm begins with computing the marginal ToS's $T_1, T_2, ..., T_n$ of each image channel, in which each ToS T_i is associated with a set of shape S_i. A set of components $S = \bigcup S_i$ from multiple trees is defined as the initial shape set. Here, we need to think about the relation between these marginal trees, specifically, the inclusion relationship between shapes from different trees. Therefore, a GoS G is created by merging all the components from marginal trees concerning their inclusion relation, $G = \bigcup T_i$. The GoS merging procedure is illustrated as the first part in Fig. 2.10. In the GoS, two marginal shapes from different trees may overlap, in other words, a pixel may belong to several nodes which are not in the same tree. The problem now is to extract a unique tree from this GoS.

Deduction a MToS from a GoS

The tree, which is deduced from the GoS, has to hold an algebraic decreasing shape attribute ρ, so that:

$$\forall A, B \in S, A \subset B \Rightarrow \rho(A) > \rho(B) \tag{2.6}$$

where A and B are shapes which belong the set of component S.

The chosen attribute is the depth attribute. The depth of a shape is defined as the length of the longest path from a shape to the root. The depth image ω is computed as:

$$\omega(x) = \max_{x \in X, X \in G} \rho(X) \tag{2.7}$$

where x is the pixel which belongs to the shape X. An example of the depth attribute ρ is illustrated in Fig. 2.10(b). The depth of each shape is computed on the marginal trees and the graph of shapes. In the end of the process, an depth map ω is reconstructed. Note that, after computing the depth attribute, component B and D are set to the same level and they are merged in the depth map ω.

By denoting h being the thresholded level, we can consider $M(x)$ as the max-tree of ω, $M(x) = \{\Gamma \in CC([\omega(x) \geq h])\}_h$. However, because M may form components with holes, the hole filled max-tree of the depth image is constructed. It is also the final MToS T_ω of the image. The tree deduction procedure is depicted in the second part in Fig. 2.10(a).

A GoS is a complete representation of an image, whereas the MToS is not, since it lost information while computing the depth attribute. A node in the MToS may contain different color values. Therefore, to reconstruct an image from the tree, we assign each node to a median or average value of all color values belonged to that node [17].

2.3.4 Conclusion

In this section, we present the second tree-based representation of an image: tree-based on the threshold decomposition. Typical examples are the Min- and Max-tree [132] and the tree of shapes [18]. These trees are constructed from the connected components thanks to the inclusion relation. Each node in the tree represents a particular connected component in the image. These trees are designed to represent bright and dark structures in the image. Therefore, the leaf nodes in these trees correspond with image extrema. We also introduce an extension of the ToS in the multivariate images, called MToS. We use this tree in Chapter 3 to compute our new distance.

2.4 Tree simplification

In the previous section, we discussed different types of tree-based representation. However, a tree may contain a lot of nodes. Therefore, tree simplification is important to reduce the number of nodes in the tree, thereby reducing the runtime for later tasks. In this section, we review several simplifying operators on the tree.

2.4.1 Tree filtering approach

Tree-based representation is a good way to represent an image and provides an organization between components in a hierarchical relation. However, the tree nodes,

which are close to the leaf nodes, usually correspond to small areas in the image and do not contain much semantic information. That requires us to merge these nodes to simplify the tree for later processing steps. In other words, we eliminate some connected components and then reconstruct a new image from the remaining tree nodes. We call these steps as tree filtering operators. Based on the strategies preserving or removing nodes in the tree, we can classify these operators into two classes: the tree pruning and non-pruning [58]. The former cut the sub-branches of the tree. Therefore, if a node is filtered, all its descendants are also pruned. Whereas, in the latter case, the descendant nodes of the filtered node can be preserved. To filter out or preserve the nodes in the tree, we rely on their criteria stated in the following subsections. Nodes in the component trees can be simplified based on regional maxima, minima, or extrema, while in the case of partition trees, we employ the same criteria as the ones used for tree construction, such as the size, contrast, texture or shape of the component [58].

Moreover, the criteria can be classified into two classes: increasing and non-increasing criterion. The criterion C on the connected component R is called increasing if it holds this condition, otherwise it is called non-increasing:

$$\forall R_1 \subseteq R_2 \Rightarrow C(R_1) \leq C(R_2) \tag{2.8}$$

2.4.1.1 Increasing Criterion

This criterion depends on the attribute of each node in the tree. If the criterion value of the node is lower than the threshold value, the node is filtered out, and its corresponding region is merged to the parent node. The increasing criterion guarantees that if a node is pruned, all of its descendants are also removed [58].

The increasing criterion is widely used to simplify the tree T. Several increasing criteria of a region X in the image f are given as follows:

- Area$(X) = \{\#p | p \in X\}$ (# is the number of pixels p in X);
- Height$(X) = \max_{p \in X} f(p) - \min_{p \in X} f(p)$;
- Volume$(X) = \sum_{p \in X} \left(\max_{p \in X} f(p) - f(p) \right)$;
- Diagonal length of the smallest bounding rectangle.

2.4.1.2 Non-increasing Criterion

Besides the increasing criterion, several non-increasing criteria are also proposed as follows:

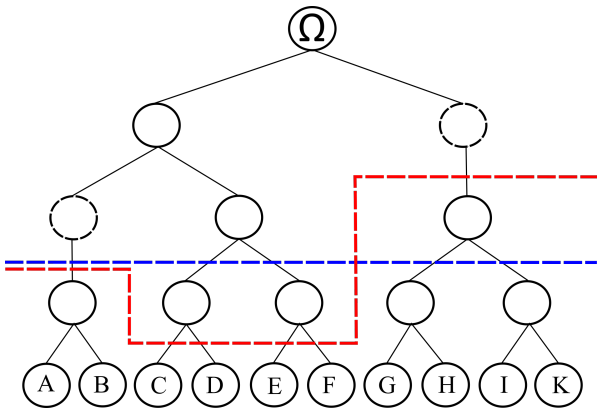
- Perimeter $P(X)$
- Compactness$(X) = \frac{4\pi Area(X)}{P^2(X)}$
- Elongation$(X) = \frac{l_{\max}(X)}{l_{\min}(X)}$,where l_{max} and l_{min} are major and minor axes of the minimum covering ellipse of X.

On the contrary to the increasing criterion, the descendants of the removed node in case of non-creasing criterion A, can be preserved. Several tree filtering approaches have been proposed to deal with this situation [58]:

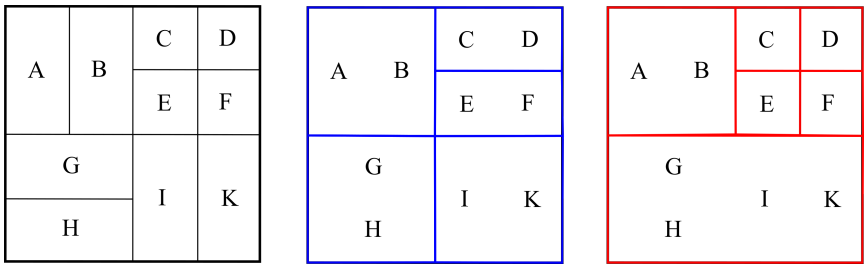
- Min: A node is kept if it passes the criterion and all of its ancestors are kept.
- Max: A node is kept if it passes the criterion or any of its descendants is kept.
- Viterbi: The removal and preservation of nodes are considered as optimization problems. For each leaf node, the path with the lowest cost to the root node is selected.
- Direct: A node is kept if it passes the criterion. The pixels belonging to the nodes that have been removed are merged to the node of their first ancestor that has to be preserved.
- Subtractive: is the same as the direct rule except that the gray levels of surviving descendants of removed nodes are also lowered, so that the contrast with the local background remains the same.

2.4.2 Hierarchical image segmentation

A major difference between the tree based on the threshold decomposition T_t reviewed in Section 2.3, and the hierarchy of segmentations T_h reviewed in Section 2.2 is that any cut of a type T_h gives a partition of the image domain, whereas any cut (except the root) of a type T_t yields a subset of the image domain. Here, we focus on the "cut" in a hierarchy of segmentation.



(a) Horizontal and non-horizontal cut.



(b) Left: The finest partition. Middle: Partition from the blue cut. Right: Partition from the red cut.

FIGURE 2.11: An example of the hierarchical cut.

Hierarchical segmentation is widely used in image segmentation by providing high quality results through multi-scale image analysis. It is able to generate a subset

of all possible partitions π_i (set of distinct regions that do not overlap) of an image from fine to coarse.

Given a hierarchy representation of an image H, a method for obtaining any partition π_i from H is defined as a "cut" in a hierarchy. In other words, a cut of a hierarchy H is a subset of H such that every path from the leaf node to the root node contains one and only one node in the cut slice. This cutting condition is presented in [26]. There exist two types of cut of a hierarchy: "horizontal cut" and "non-horizontal cut" which are illustrated in Fig. 2.11. The first one, also the simplest method, is "horizontal cut", which is a partition of a hierarchy at a fixed level. Examples of the horizontal cut can be found in [52] as using a given threshold to the ultrametric contour, or in [102] as applying a value of α to the α-tree.

The horizontal cut is quite simple but usually does not achieve a remarkable performance. Alternatively, a definition of "non-horizontal cut", which is based on an optimization model of the energies calculated on each node of the hierarchy, is proposed [69]. In the non-horizontal cut, a partition is generated by associating different disjointed regions from different levels of the hierarchy that satisfies the cutting condition. The number of nodes is finite, so is its set of cuts. The "best cut" is the one that minimizes the energies function. It is unique so that it holds the global property. Moreover, this optimal also satisfies the local property since the energy of each cut node is lower than the energy of its parent or its children. Hence, the optimal cut is both local and global [48].

In non-horizontal cut case, a hierarchy can be segmented by adopting a scale parameter λ. A general form of the energy function E_λ of a region R is:

$$E_\lambda(R) = E_\phi(R) + \lambda E_\partial(R) \tag{2.9}$$

where E_ϕ and E_∂ represent respectively a fidelity term and a regularization term, λ is a positive value. The coarse level of the segmentation depends on the value of λ. The higher value of λ is, the coarser segmentation of hierarchy is.

FIGURE 2.12: Optimal Cuts Pyramids: Optimal cuts using Mumford-Shah function, shown for different λ_s. Images are taken in [48].

The optimal cut can be calculated using the dynamic programming by climbing up in one ascending pass on the hierarchy. The energy of each node is compared to the energy on its parent and children node. The one that has less energies is kept for

continuing the pass. To compare the energy between nodes, three modes of composition including addition, supremum and infimum, are defined as the following:

- Addition: $\sum_{R_i \in childof(R)} E_\phi(R_i) + \lambda E_\partial(R_i)$
- Supremum: $\bigvee_{R_i \in childof(R)} E_\phi(R_i) + \lambda E_\partial(R_i)$
- Infimum: $\bigwedge_{R_i \in childof(R)} E_\phi(R_i) + \lambda E_\partial(R_i)$

If the energy of the node corresponding to region R is lower than the energy of its children node, we remove these children nodes out of the hierarchy. On the contrary, we remove the node corresponding to region R, and update the parent relationship on the tree. The algorithm is stopped when the root node is reached and all the energy conditions are satisfied.

The most popular scale parameter function is the Mumford-Shah function, which is first proposed in [46]. The Mumford-Shah function has been studied extensively in the last decades [40, 47, 48]. This function is defined as:

$$E(R) = \sum_{x \in R} \|f(x) - m(R)\|^2 + \lambda \sum (|\partial R|) \tag{2.10}$$

where the first term (fidelity term) is the variance between function f and its average $m(R)$ in the region R. For example, the function f can be the color function of an image, or the scalar luminance $f = (r + g + b)/3$ which is presented in [48]. The second term (regularization term) is equal to contour length $|\partial R|$ of each node R. Some examples of the optimal cut using Mumford-Shah function are shown in Fig. 2.12. These figures illustrate the segmenting results with respect to different value of λ. The higher value of λ gives the coarser segmentation.

Recently, many supervised learning-based methods are proposed to achieve the best cut in the hierarchy [49–51]. In these papers, the hierarchies are constructed from local information, such as multiscale local brightness, color, and texture cues [52]. Besides these low-level features, several mid-level features, which are based on graph partition, region, and Gestalt properties, are used to create a classification model for predicting the best segmentation. In [49], they train a classifier model to predict the probability for each clique (a set of parent node R and a union of its children node R_i) in the hierarchy. A label $l_i = +1$ or $l_i = -1$ is assigned to a parent node to indicate whether its children are merged. Hence, all the leaf nodes are assigned with label $l_i = +1$. The set of nodes, which are labeled as $+1$ and their parents are labeled as -1, are considered as the final segmentation.

In [51], they consider regions, which belong to the optimal cut, are properly-segmented region. Regions upper (resp. lower) the optimal cut are over-segmented (resp. under-segmented) regions. A label $x = \{-1, 0, 1\}$ is assigned to each node v_i to indicate the 3 classes of node based on its scale (under-segmented, properly-segmented and over-segmented). They employ a regression model to predict the scale of regions using mid-level features. Then an optimize function is formulated to select the best cut as the set of regions that better balance between over-segmentation and under-segmentation.

2.5 Image segmentation

Over the past several decades, image segmentation has been widely used as an intermediary step in computer vision, image processing, and pattern recognition. This technique aims to partition a digital image into multiple meaningful segments (set of superpixels). It helps to facilitate the analysis through simplifying image or changing the image representation. The segmentation algorithm assigns different labels to pixels in the image, such that the pixels with the same label share some common characteristics such as color, intensity or texture. In the end of the process, we obtain a set of segments that cover the entire image. In other words, image segmentation is a multi-classes labelling problem.

Although image segmentation has long been studied, it still poses many problems. Firstly, a segment may belong to single or multiple connected components in the image. Secondly, image segmentation is oriented toward a specific application. In other words, different applications require different methods and features. The third challenge with image segmentation is that we can not achieve a reasonable unique image segmentation. Given an image, if we ask a set of people to segment the image, we will get multiple results of what is a good segmentation. For example, in the Berkeley segmentation dataset [133], the ground truth image segmentation from five different people are different.

Image segmentation attracts a deep and rich research, and various algorithms are proposed to deal with different tasks. In many literature surveys [134, 135], image segmentation methods are classified into unsupervised and supervised classes. The former methods are implemented without using any knowledge about the object or user's inputs. On the other hand, the latter methods use the prior information in the training datasets or user's input in the testing datasets. Therefore, a large training dataset may give better performance. Because of high complexity and computation, these methods require sufficient hardware for calculation.

In this thesis, we do not try to enumerate all the image segmentation methods. Instead, we highlight a few of the most popular unsupervised methods.

2.5.1 Superpixel segmentation

This section presents a review about superpixel segmentation, which is an oversegmentation of an image into few tens to thousand segments. This technique aims to generate a new representation of an image, which is much easier and faster to process. The superpixels are split as long as they are not divided by object contours. Thus, the objects can be recovered at later procedures. Moreover, superpixel segmentation is an efficient way to get local features instead of calculating directly from the original image. An example of the superpixel is illustrated in Fig. 2.13. In this image, different segmentation results are provided w.r.t the number and the compactness of superpixels.

Due to its simplicity, superpixel segmentation is used in various applications in the image processing community. Instead of looking for all possible candidates in the images, the object proposal methods, which are based on the superpixels, are presented to reduce the search space. These approaches can be found in many papers [136–139]. Furthermore, superpixel segmentation is employed for semantic segmentation [139, 52, 50, 49], salient object detection [22, 140, 141], and object tracking [142].

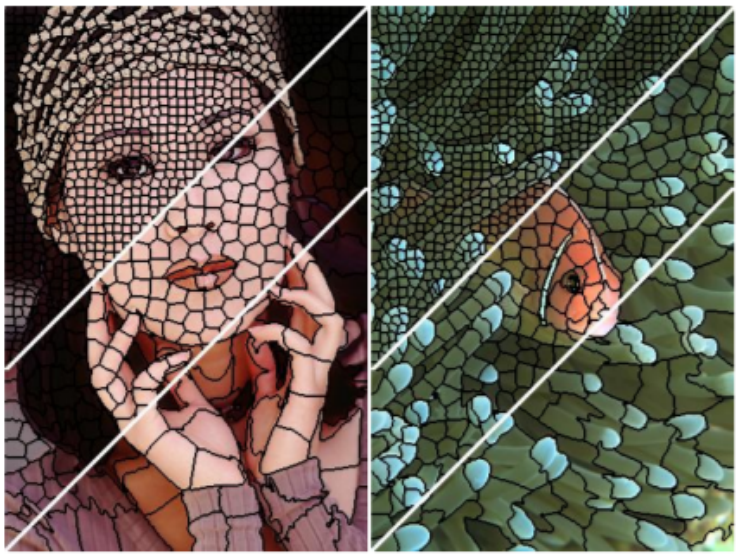


FIGURE 2.13: Image segmentation using SLIC algorithm with the size of each superpixel are respectively equal 64, 256, and 1024 pixels [25].

Several superpixel segmentation methods are proposed in the state of the art. Specially, in this thesis, we employ the simple linear iterative clustering (SLIC) algorithm [25] as an intermediary step in our method.

The basic idea of the SLIC method is similar to the k-mean algorithm [143]. This method is also a pixels clustering approach based on calculating the color similarity and proximity in the image domain [25]. The five-dimensional feature $[labxy]$ is used to measure the distance between pixels, where $[lab]$ is the pixel color in the CIELAB color space, and $[xy]$ is the coordinate of the pixel in the domain. The dissimilarity between two pixels p_i and p_j is computed as:

$$d_{lab} = (l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2 \tag{2.11}$$

$$d_{xy} = (x_j - x_i)^2 + (y_j - y_i)^2 \tag{2.12}$$

Clearly, the distance d_{lab} is limited while d_{xy} depends on the size of the image. Thus, a normalization of these two distances is necessary to incorporate both of them.

$$D = \sqrt{d_{lab} + \frac{m^2}{S^2} d_{xy}} \tag{2.13}$$

where $S = \sqrt{N/k}$ normalizes the size of superpixels, such that N is the number of pixels and k is the number of superpixels. m is used here as a compactness parameter between the color dissimilarity and spatial proximity. The higher the value of m is, the more compact between the clusters [25] are.

Algorithm 4: SLIC superpixels.

Initialize cluster centers c_i by sampling pixels at regular grid interval S;
Move cluster centers to lowest gradient location within 3×3 neighborhood;
Set label $label(i) = -1$ for each pixel i;
Set distance $d(i) = \infty$ for each pixel i;
repeat
  for *each cluster center* C_k **do**
    for *each pixel* i *in a* $2S \times 2S$ *region around* C_k **do**
      Compute the distance D between C_k and i ;
      if $D < d(i)$ **then**
        Set $d(i) = D$;
        Set $l(i) = k$;
      end
    end
  end
  Compute new cluster centers ;
  Compute residual error E ;
until $E \leq threshold$;

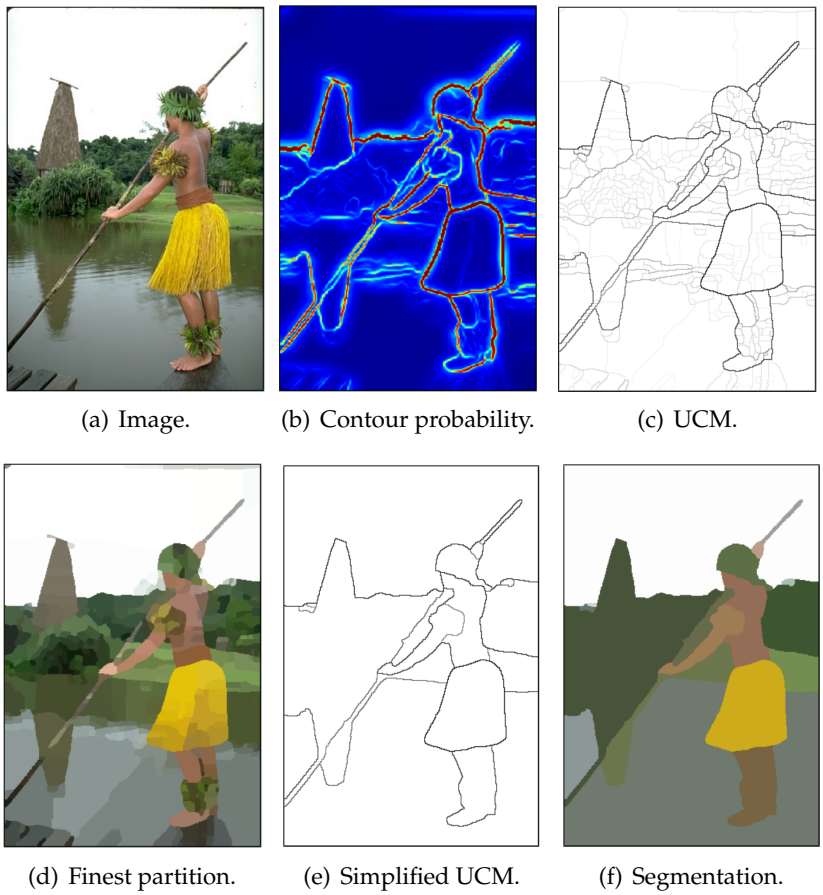
The SLIC algorithm is presented in Algo. 4. This algorithm iteratively associates pixels with the nearest cluster center, then updates the new position of the center. This process is repeated until the error converges. Finally, a post-processing step is used to reassign pixels to nearby superpixels.

2.5.2 Contour-based segmentation

Contour detection and image segmentation have been studied since the early state of image processing. These two methods are related, but not identical. In general, the contour detection methods do not guarantee to have closed contours, thereby usually providing unsatisfying segmentation image. Classical contour detection methods are based on the intensity changes between adjacent pixels in the image. This technique can be categorized into two basic local approaches: first and second-order differentiation. In the first-order approach, the gradient image is generated by convolving the image with a gradient mask. Edge is considered as the local maxima among pixels in the gradient image. Roberts [144], Sobel [145] and Canny [146] are some well-known methods for edge detection. On the other hand, the second-order class searches for the optimal edges where the second derivative is zero. A well-known isotropic generalisation of the second derivative to two dimensions is the Laplacian [147] This method searches for the zero-crossing place, where a pixel value is positive and the others are negative (or vice versa).

Recently, the hierarchical segmentation is used to extract the soft boundary image, called ultrametric contour (UCM). In [148, 149], the authors take into account the local contour cues along the boundary regions, then integrate with region attributes to achieve the ultrametric contour. In [150], the posterior probability of a contour is computed w.r.t the coordinates of the pixel and the orientation of the boundary. This approach is then developed in [52] by using the multiscale model. A globalized probability of a boundary is then computed from local and spectral information. An example of this method is illustrated in Fig. 2.14.

Another method to detection the contour is presented in [151]. It is a learning method by using random decision forest to capture the structured information. The



(a) Image. (b) Contour probability. (c) UCM.

(d) Finest partition. (e) Simplified UCM. (f) Segmentation.

FIGURE 2.14: Hierarchical segmentation from contours. Images are extracted in [52]

contour detection is considered as predicting local segmentation masks given input image patches.

2.5.3 Watershed

In this section, we present a well-known image segmentation algorithm, namely watershed, which is based on the growth of the region. The basic idea of this method is to simulate the flooding process. Imagine that we have a surface with holes and water is falling on it. Different labels are assigned to different regional minima (holes). Each regional minima is gradually filled by the water, then the unlabeled pixels are assigned to the closest basin. Some constraints are set to prevent water merging from different holes. The basins are progressively grown until all pixels in the image are filled. That leads to the ridges between basins, which we call the watersheds [152].

The most famous watershed algorithm is seeded watershed segmentation or also known as marker controlled watershed. The markers are put on the image to define different regions. The segmenting result is highly depended on the position of the started markers. In [153], the initial seeds are the local gradient minima. The seeded region is grown thanks to the front propagation method by using the priority queue that involves the order of pixels which are calculated from a distance function between the current pixel and the nearest seed. As a consequence, a catchment basin

around a seeded region is defined as a set of pixels that are closer to this region than the others.

The watershed is firstly proposed by Digabel and al [154], and then improved in [155]. Several watershed segmentation algorithms have been reviewed in [156], in particular, watershed by immersion [157] or by topographical distance [158]. The former algorithm is presented in [157], which has two steps: **1)** sorting pixels in increasing order of gray value, **2)** flooding step starts from the regional minima. In latter one, several shortest path algorithms are used to perform watershed segmentation [67, 159].

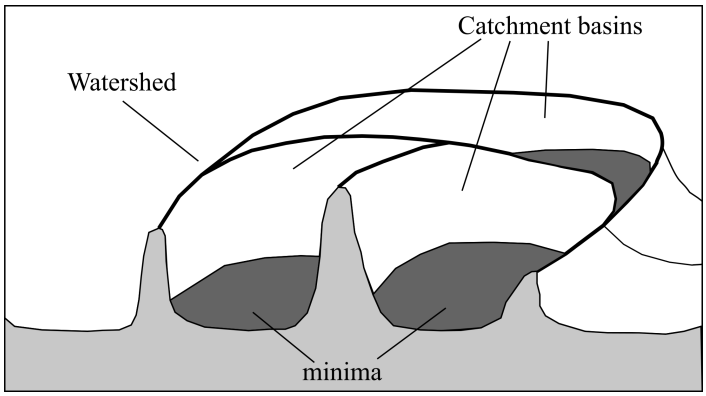


FIGURE 2.15: Minima, catchment basins, and watersheds on the topographic representation of a gray-scale image.

However, the performance of watershed segmentation highly depends on the algorithm that is used to compute the gradient image. Therefore, the watershed transform usually produces an over-segmented image, with many irrelevant regions. An example of the watershed on the topographic representation of the gray-scale image is illustrated in Fig. 2.15.

2.5.4 Graph-based segmentation

Graph-based approaches have been studied for decades. An image can be considered to be a weighted graph $G = (V, E, w)$, where vertices V in the graph represent pixels in the image and edges E correspond to connectivities between adjacent pixels. The edge weight w represents the dissimilarity between pixels. The basic idea of graph-based image segmentation is separating a set of nodes V into different subset nodes $V_1, ..., V_m$ such that the similarities among nodes in the same subset are higher than those across different subsets.

2.5.4.1 Normalized cut

In [161], Shi et al. proposed an algorithm to segment an image into multiple partitions, called Normalized cut. Denoting $w(i, j)$ as a weight between two nodes v_i and v_j in the image, a cut in the graph that partitions an image into two disjoint sets A and B is formulated as the total weight of the edges that have been removed.

$$cut(A, B) = \sum_{i \in A, j \in B} w(i, j) \tag{2.14}$$

(a) $k = 5$. (b) $k = 15$. (c) $k = 30$. (d) $k = 45$.

FIGURE 2.16: The segmenting results by using normalized cut algorithm with different value of number of segmentation k. The image is extracted from [160].

The goal of this method is minimizing the total edge weights along with the cut. However, the segmentation based on this formulation can lead to a problem, that is, it tends to generate small regions cause the cut function increases along with the number of edges across the bipartition. To overcome this limitation, the normalized cut is proposed w.r.t the number of nodes in the graph:

$$cut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} \tag{2.15}$$

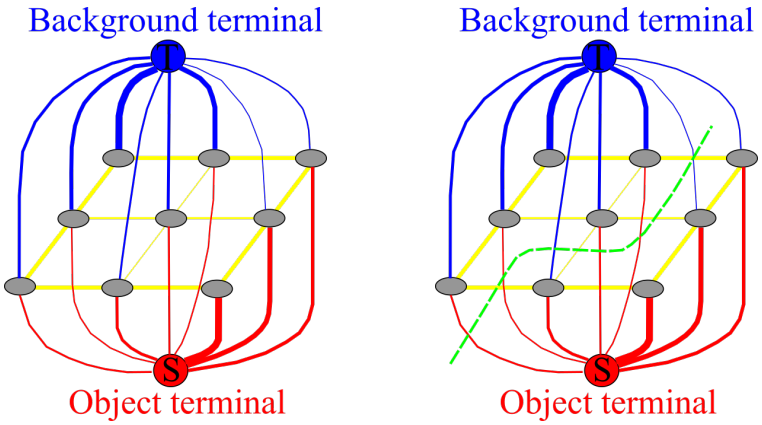
where $assoc(A,V) = \sum_{i \in A, j \in V} \mathrm{w}(i,j)$ is the total connection from the nodes belonging to A to all nodes in the graph. This technique aims to optimize the cut function while also maximizing the associativity in the graph. However, solving this problem needs an NP-complete complexity algorithm. An approximated solution is proposed by Shi et al. in [161]. This method converts the cut problem into a Rayleigh quotient to approximate the minimal normalized cut. It is shown that the second smallest eigenvector y is a solution to the normalized cut problem. This method can be repeated iteratively for the case of segmenting an image into multiple partitions. An example is illustrated in Fig. 2.16. This is the result of the normalized cut algorithm according to different values of the number of segments.

2.5.4.2 Graph cut

In this section, we present another graph-based method for image segmentation. In this method, the source S and sink T node are defined for a weighted $S-T$ graph. The notion cut in $S-T$ graph is described as a set of edges so that there is no path from the source to the sink node. The graph cut algorithm aims to optimize the cost of the $S-T$ cut, which is introduced in Eq. (2.14).

Several approaches are proposed to solve the equation above such as max-flow [162–164], push-relabel [165]. In this section, we discuss the method that is presented in [164] for interactive segmentation.

The illustration of this method is shown in Fig. 2.17, where the red, blue and gray nodes respectively represent the source S, sink T and pixel p in the image. The t-link ($\{p,S\}$ (red edges) or $\{p,T\}$ (blue edges)) connects each node to the terminal, in other words, encodes the regional term between non-seed to seed pixels. Whereas, the n-link represents the neighborhood relation N between adjacent pixels which is



(a) A graph with source and sink terminal.

(b) A cut on a graph.

FIGURE 2.17: A simple 2D segmentation example for a 3×3 image. The cost of each edge is reflected by the edge's thickness.

defined as connectivity term. Therefore, the set of edges E is defined as:

$$E = N \underset{p \in P}{\cup} \{\{p, S\}, \{p, T\}\} \tag{2.16}$$

In Fig. 2.17, the thickness of the t-link edge depends on the weight of each edge. In [163], the authors state that the minimum weight of a cut is equal to the maximum value of the flow from S to T. The optimal cut can be computed by adapting the new max-flow version which is presented in [164]. This cut is illustrated as the green dash line in Fig. 2.17.

2.5.4.3 Felzenswalb and Huttenlocher method

Another well-known graph-based segmentation is presented in [90] by Felzenszwalb and Huttenlocher, namely FH algorithm. Again, an image is considered to be a graph, in which the edge weight w_{ij} between two pixels p_i and p_j in the image I, can be computed as: $w_{ij} = I(p_i) - I(p_j)$. Based on the edge weight, a minimum spanning tree (MST) that represents every pixel in the image is constructed using Kruskal's algorithm. The basic idea of the image segmentation in this case is equivalent to clustering the MST.

The method to cut the MST is based on the boundary condition which decides whether two regions should be connected. It is defined as:

$$D(C_1 - C_2) = \begin{cases} true, \text{ if } Dif(C_1, C_2) > MInt(C_1) \\ false, \text{ otherwise} \end{cases} \tag{2.17}$$

where $Dif(C_1, C_2)$ is defined as,

$$Dif(C_1, C_2) = \underset{v_i \in C_1, v_j \in C_2}{min} w_{ij} \tag{2.18}$$

and $MInt(C_1, C_2)$

$$MInt(C_1, C_2) = min((Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2))), \tag{2.19}$$

FIGURE 2.18: A baseball scene (432 × 294 grey image), and the segmentation results produced by FH algorithm ($\sigma = 0.8, k = 300$). Image extracted from [90].

where $Int(C)$ is the maximum value of the edge weight of the connected component C in the MST and $Dif(C_1, C_2)$ is the difference between two adjacent connected components. The value $MInt(C_1, C_2)$ is the minimum internal of either C_1 or C_2. The parameter $\tau(C)$ is presented to control the relative difference between inter- and intra-component.

$$\tau(C) = \frac{k}{|C|} \tag{2.20}$$

where $|C|$ is the total pixels in C. The parameter k is set to avoid small connected component generated from this method.

Algorithm 5: Image segmentation using FH algorithm.

Data: A graph $G(V, E, w)$, m edges, n vertices x, parameter k
Result: A segmentation $S = (C_1, ..., C_r)$

Initially, sort edges e_i in ascending order of weight;
Set $S^0 = (\{x_1\}, ..., \{x_n\})$, each cluster contains one vertex;
for *each* $t = 1, ..., m$ **do**
    Let x_i and x_j be the vertices connected by e_t. ;
    Let $C_{x_i}^{t-1}$ be the connected component containing point x_i on iteration $t-1$. Likewise for $C_{x_j}^{t-1}$. ;
    $\tau(C_{x_i}^{t-1}) = \frac{k}{|C_{x_i}^{t-1}|}$;
    $\tau(C_{x_j}^{t-1}) = \frac{k}{|C_{x_j}^{t-1}|}$;
    if $|e_t| < \min\left\{(Int(C_{x_i}^{t-1}) + \tau(C_{x_i}^{t-1}), Int(C_{x_j}^{t-1}) + \tau(C_{x_j}^{t-1}))\right\}$ **then**
        Merge $C_{x_i}^{t-1}$ and $C_{x_j}^{t-1}$;
    end
    $S = S_m$;
end

The whole algorithm is a simple greedy method presented in Algo. 5. It is actually an iterative method that merges smaller regions to bigger regions such that these regions satisfy with the boundary condition.

An image and its segmentation result based on FH method are illustrated in Fig. 2.18. The method runs in $O(mlogm)$ time for m graph edges.

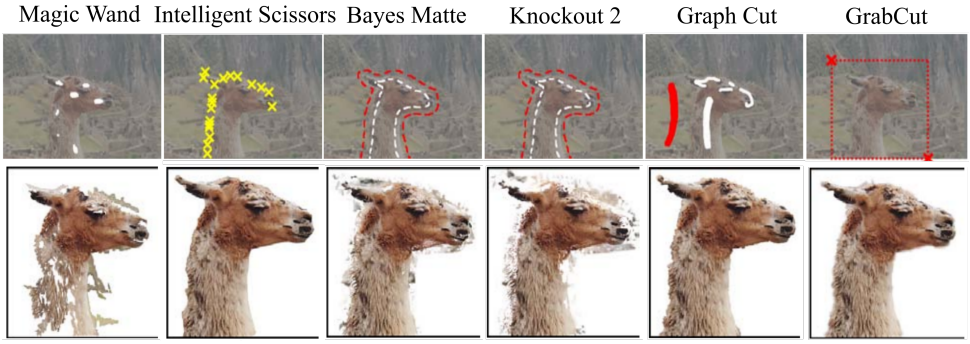


FIGURE 2.19: Comparison of some matting and segmentation tools. The top row shows the user interaction required to complete the segmentation or matting process. These methods are: Magic Wand [166], Intelligent Scissors [167], Bayes matting [168], Knockout 2 [169], Graph Cut [170], GrabCut [23]. The bottom row illustrates the resulting segmentation.

2.5.5 Interactive segmentation

We have presented in previous sections several methods for automatic image segmentation. In this section, we discuss another type of segmentation, called interactive segmentation, which involves human intervention to supply high-level information. Human assisted segmentation has been long studied and can be employed into many applications to extract the object regions in the image. This application requires users to provide additional information for the objects and background regions. Generally, users can put some markers to define what is the foreground and background. The human annotation can be chosen among scribbles, bounding box, or even a point. Then these constraints are put to an optimized model to produce an initial segmentation. Interactive segmentation is an iterative process where the user is in a loop. It is a target driven task so that the users can add or remove markers to refine the current segmentation until they get satisfying results. The goal of interactive segmentation is to provide a way to extract the foreground region quickly and accurately.

Various approaches have been proposed to solve this problem. Each method is created with different specific domain. For instance, active contour [171] and a similar approach [172], are generally used in medical images. Other methods such as Graphcut [170], Grabcut [23] are used for photo editing in natural image. Depending on the application, we can choose the appropriate method to obtain satisfying results.

Here, we highlight several well-known approaches in the interactive segmentation domain. These methods are presented in Table 2.1.

TABLE 2.1: Algorithmic approaches to interactive segmentation.

Method	Example Algorithm
Region Growing	Seeded Region Growing [173]
Classifiers	Simple Interactive Object Extraction [174]
Graph and MRF Models	Interactive Graph Cuts [170, 23]
Hierarchy	Tree-based representation [130]
Shortest path	Dijkstra [59, 175–177]

In [173], the authors propose an inexpensive and straightforward algorithm which is based on the color similarity between adjacent pixels. The input of the method is a set of seed points which has been categorized into two sets: one for object and another one for the background. Each pixel is assigned to a label (object or background) depending on the distance between the color of pixel and the average color of two classes.

The interactive segmentation based on the classifier model is presented in [174]. In the beginning, the original image is transformed to LAB color space which is close to our perception. The basic idea of this method is based on the color signature [178] of the known object and background from user markers. Relying on the generated color signatures, represented as a weighted set of cluster centers, the pixel is then classified depending on their distance to the foreground and background color signatures.

The Graphcut algorithm [170] proposed by Boykov considers interactive segmentation as a globally optimal solution using a fast min-cut/max-flow algorithm. This method is generally used to segment the object in natural images. Then its extended version is presented in [23] by two enhancements: "iterative estimation" and "incomplete labeling" to reduce the degree of user interaction. In [179], viewing an image as a weighted graph, the authors prove the connection between watersheds and graph cut, and use it for interactive segmentation framework.

Another approach is based on the shortest path algorithm [59, 175–177]. They search for the shortest path from every pixel to the two sets of marker, and compute the path-wise distance. A label of each pixel is then assigned to the closest marker. The Dijkstra algorithm is a well-known method for solving this kind of problem. In [167], *Intelligent scissors* method is proposed to find the object contour via shortest paths in a graph near the boundary of the target clicked by the user.

In [130], the authors propose to use the hierarchical representation of an image to perform interactive segmentation. Firstly, some markers are put in the image. Then the nodes on the tree that correspond to the markers in the image are labeled. Depending on the distance from every node on the tree to the assigned nodes, we can decide whether it belongs to the object or background. The segmenting image can be reconstructed from the label of every node on the tree.

2.6 Distance function

The distance function is used widely in image processing and computer vision, especially in mathematical morphology, because it measures the dissimilarity between pixels in the image [53]. The distance map which is deduced from the distance function, is a gray-level image that looks similarly to the binary image, except the fact that the intensity values of the pixels inside the object region are changed proportionally to the distance between the pixel to the boundary of the object. Besides, a distance map provides a way to get another representation of the original image or bring more information to the original image.

In this section, we review several well-known distances, then present a new distance called Dahu pseudo-distance, which is computed based on mathematical morphology background. Our work will be mainly based on this new distance.

2.6.1 Definitions and examples

A distance in 2D image is a function: $\mathbb{Z}^2 \rightarrow \mathbb{R}^+$. Denoting p, q, and r are pixels in the image, the distance function satisfies these following conditions:

- $d(p,q) = 0 \Leftrightarrow p = q$
- $d(p,q) = d(q,p)$
- $d(p,q) \leq d(p,r) + d(r,q)$

As presented in previous sections, a digital image is also considered to be a graph, where vertices represent pixels in the image. A path-wise distance between two pixels p and q is defined as the shortest path that connects these two pixels. This shortest path $\pi(p,q)$ is associated with a metric d_S to represent the dissimilarity between these two pixels.

$$d_S = \min_{\Pi(p,q)} d(p,q) \tag{2.21}$$

where $\Pi(p,q)$ is a set of all possible paths that connect p and q.

6	5	4	3	4	5	6
5	4	3	2	3	4	5
4	3	2	1	2	3	4
3	2	1	0	1	2	3
4	3	2	1	2	3	4
5	4	3	2	3	4	5
6	5	4	3	4	5	6

FIGURE 2.20: Discrete distance function calculated from the central pixel of an image

Several well-known discrete distances used in mathematical morphology are city-block distance d_{cb}, chessboard distance d_{ch} and Euclidean distance d_{eu}, which are defined as follows:

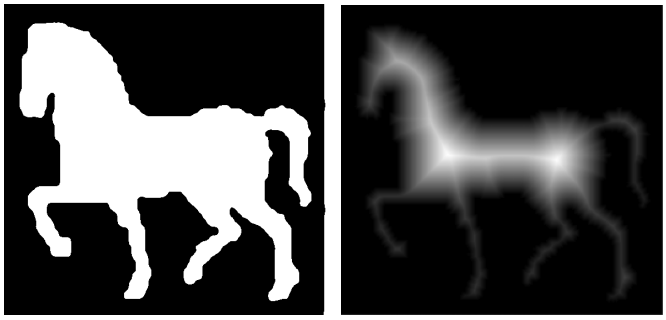
$$d_{cb}[(x_1,y_1),(x_2,y_2)] = |x_2 - x_1| + |y_2 - y_1| \tag{2.22}$$

$$d_{ch}[(x_1,y_1),(x_2,y_2)] = \max\{|x_2 - x_1|, |y_2 - y_1|\} \tag{2.23}$$

$$d_{eu}[(x_1,y_1),(x_2,y_2)] = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2.24}$$

where (x_i, y_i) are the coordinates of a pixel. The Euclidean distance is computed without considering the neighborhood relationships between adjacent points in the image domain. Note that, the shortest path between two pixels is not necessarily unique. There may have more than one shortest path in the image between two pixels [53]. An example of the 4-connected distance map, which is computed from the central pixel, is illustrated in Fig. 2.20.

The distance functions are used in several applications in mathematical morphology for analysing the object in the binary image [53]. The first application of



(a) Binary image. (b) Distance transform.

FIGURE 2.21: Skeleton application by using distance transform approach.

the distance function that we present in this section is the skeleton application. A skeleton morphology is a way to extract a region-based shape feature representing the general form of an object. A skeleton is thin so that it contains fewer pixels than an object image. The skeleton also represents local object symmetries and the topological structure of the object. An example of the skeleton of the object is illustrated in Fig. 2.21. In this image, a distance map is computed, where the seed pixels belong to the background of the image by propagating from the seed pixels to the all pixels in the image. A priority queue is used for this procedure. The intensity value of pixels in the resulting image increase from the boundary to the pixels inside the object, except pixels belong to the ridges, which also have higher intensities than adjacent pixels. These ridges are a set of points where the propagation process from different directions meet. They are also the skeleton of the object in the image.

(a) Set X. (b) $D_4(x)$. (c) $\varepsilon_{\Diamond}^{(6)}(X)$.

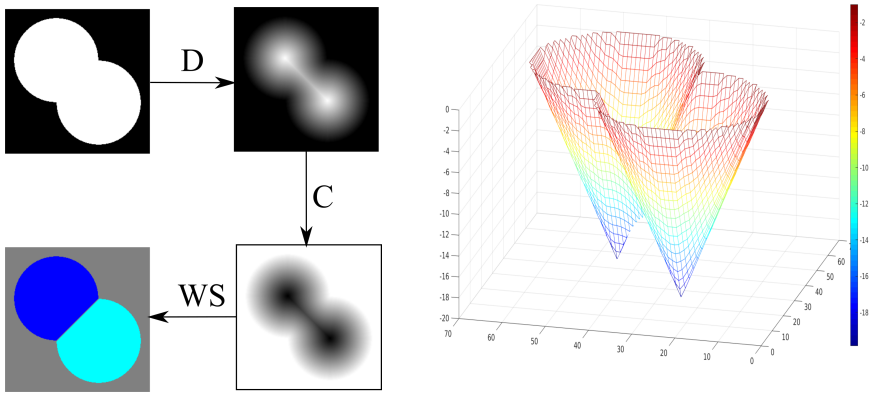
FIGURE 2.22: Distance function and erosion: the set X eroded by a diamond shaped structuring element of size 6 is obtained by thresholding the 4-connected distance D on X. Image is extracted from [180].

The distance transform is also used in binary erosion and dilation, which are two fundamental operations in mathematical morphology. Here, we analyze the erosion operator. The dilation operator can be implemented similarly. Normally, erosion operator of a set X, denoted as $\varepsilon_B(X)$ is implemented by using a structuring element B: $\varepsilon_B(X) = \{x | B_x \subseteq X\}$. Besides, erosion (respectively dilation) can be applied using the distance transform. An illustration is depicted in Fig. 2.22. From the binary image, a distance map is computed using the city-block distance with regard to the seed pixels are put in the background. Depending on the size n of the erosion operator, thresholding is applied to the distance map to get all values strictly greater than n. Besides, depending on the kind of structure element (diamond $\Diamond$ or

square □), 4-connectivity or 8-connectivity are used. The definitions of the erosion operator based on the distance transform are expressed as follows:

$$\varepsilon_{\Diamond}^{(n)}(X) = \{x \in X | D_4(x) > n\} = T_{[n+1,t_{\max}]}[D_4(n)] \tag{2.25}$$

$$\varepsilon_{\square}^{(n)}(X) = \{x \in X | D_4(x) > n\} = T_{[n+1,t_{\max}]}[D_4(n)] \tag{2.26}$$



(a) Procedure of marker-controlled segmentation.

(b) Topological view of distance transform.

FIGURE 2.23: Segmentation of overlapping blobs by watershedding WS the complement C of their distance function D.

Another application of the distance transform that we present in this section is marker-controlled segmentation for separating overlapping blobs. In mathematical morphology, thresholding is a popular method and widely used to segment objects with high contrast in the image. However, this method has a limitation since it sometimes provides overlapped blobs between different objects. An example is illustrated in Fig. 2.23. As we can see, two blobs are connected in the foreground of the image. This distance map is computed, where the intensity values are the distance from every pixel to the background of the image. The distance map is complemented so that the minima of the image corresponds to the center of the objects. Finally, a watershed algorithm [67] is applied to segment these two objects.

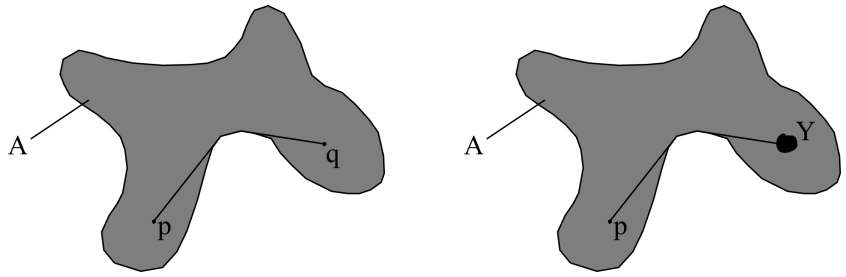
2.6.2 Geodesic distance

Finding the shortest path between two vertices is one the most common problem in graph theory. One famous solution to find the shortest path is using Dijkstra's algorithm [181]. In [53], the "geodesic distance" $d_G(p,q)$ between two pixels p and q is firstly defined in the connected set S. It is the minimum of length L of the path $\pi(p,q) = (p_1, p_2, ..., p_n)$ joining p and q:

$$d_G(p,q) = min\{L(\pi(p,q)) | p_1 = p, p_n = q, \pi(p,q) \subseteq S\} \tag{2.27}$$

The geodesic distance $d_G(p,X)$ between a pixel p and a subset X is defined as the smallest geodesic distance between p and any pixel q that belongs to X:

$$d_G(p,X) = \min_{q \in X} d_G(p,q) \tag{2.28}$$



(a) Geodesic between p and q. (b) Geodesic between p and set Y.

FIGURE 2.24: Geodesics between p and q in a connected set S, and between p and X.

The geodesic distance between two points and between a point and a set are depicted in Fig. 2.24.

The geodesic distance is then extended with considering the edge weight W between 2 pixels in the image. The geodesic distance d_G is the minimum integral of a weight function among a set of all possible paths between two pixels in the image [176].

$$d_G(p,q) := \min_{\pi_{p,q} \in \Pi_{p,q}} \int_p^q \left| W(x).\pi_{p,q}(x) \right| dx \tag{2.29}$$

where $\pi_{p,q}(x)$ is a path connecting the pixels p, q, and $\Pi(p,q)$ is a set of possible paths between two pixels in the image.

In discrete form, the geodesic length L of a path $\pi(p,q) = \{p_1 = p, p_2, ..., p_n = q\}$ between two pixels p, q in the discrete image can be computed by this equation:

$$L(\pi) = \sum_{i=1}^{n-1} w(\pi(p_i, p_{i+1})) \tag{2.30}$$

The weight $w(\pi(p_i, p_{i+1})) = |I(p_i) - I(p_{i+1})|$, which is the gradient between two adjacent pixels in the image. Then the geodesic distance is:

$$d_G(p,q) = \min_{\pi \in \Pi(p,q)} L(\pi) \tag{2.31}$$

Several algorithms are presented to compute the geodesic distance. The first simple algorithm, which computes the geodesic distance d between all pair of adjacent pixels, is the algorithm of Floyd [182]. The complexity of this algorithm is $O(N^3)$ operations, where N is the number of pixels in the image. Another algorithm is Dijkstra's algorithm. This algorithm computes the geodesic distance based on a front propagation in the image. They use a priority queue, in which the order of pixels relies on their geodesic distance to the seed pixel. The complexity of this algorithm is $O(VN + Nlog(N))$, where V is the size of neighborhood of pixels. Therefore, to calculate the geodesic distance between all pair of pixels in the image, the complexity is $O(N^2 log(N))$ operations. In [4], the authors propose an approximated approach to compute the geodesic distance, namely raster scanning. This method visits sequentially every pixel in the image in the forward direction and then backward direction. The iteration is repeated until there is no distance value changed. In practice, this

algorithm usually outputs a satisfying result in a few iterations, and thus it can be regarded as having linear complexity in the number of image pixels.

The geodesic distance is used in robotics and video games to compute an optimal path in an environment that has some obstacles [183]. In addition, it is used to detect curvilinear features and perform segmentation [184, 185]. In [176, 175, 59, 177], geodesic distance gives potential results for interactive segmentation applications. It is also employed in salient object detection [22, 186, 141, 187].

2.6.3 Minimum Barrier distance

In this section, we recall the mathematical background necessary to define the MBD in details and we show how to derive a distance map using the MBD.

In image processing applications, an image domain is associated with a graph in which vertices represent discrete pixels on the image and the set of edges on the graph corresponds with the adjacency relations $\mathcal{N}$ between pixels. A gray-level image (Fig. 2.25(a)) is then represented as a vertex-valued graph (Fig. 2.25(b)).

A path in a graph X is a sequence $\pi = \langle ..., p_i, p_{i+1} ... \rangle$ (where each p_i is a vertex of the graph), with $p_i \in X$ and $p_{i+1} \in \mathcal{N}_X(p_i)$. Also, the set of paths going from the vertex x to the vertex x' is denoted by $\Pi(x, x')$. The *barrier strength* (also called *barrier distance* or *cost*) τ of a path π in the given gray-level image u is defined as:

$$\tau_u(\pi) = \max_{p_i \in \pi} u(p_i) - \min_{p_i \in \pi} u(p_i). \tag{2.32}$$

The *minimum barrier distance* d^{MB} (MBD) between two vertices x and x' in u is then defined by:

$$d_u^{\text{MB}}(x, x') = \min_{\pi \in \Pi(x, x')} \tau_u(\pi), \tag{2.33}$$

The MBD is thus the minimum of the barrier strengths of all the paths between two given vertices. An illustration of this operator is depicted in Fig. 2.25. The blue path, which corresponds to a sequence $\langle 1, 0, 0, 0, 2 \rangle$, is considered to be the shortest path between these two red vertices. The corresponding MBD is then equal to 2.

Note that, instead of its name, the MBD is not a distance, because it can exist some x, y such that $x \neq y$ and $d_u^{\text{MB}}(x, y) = 0$.

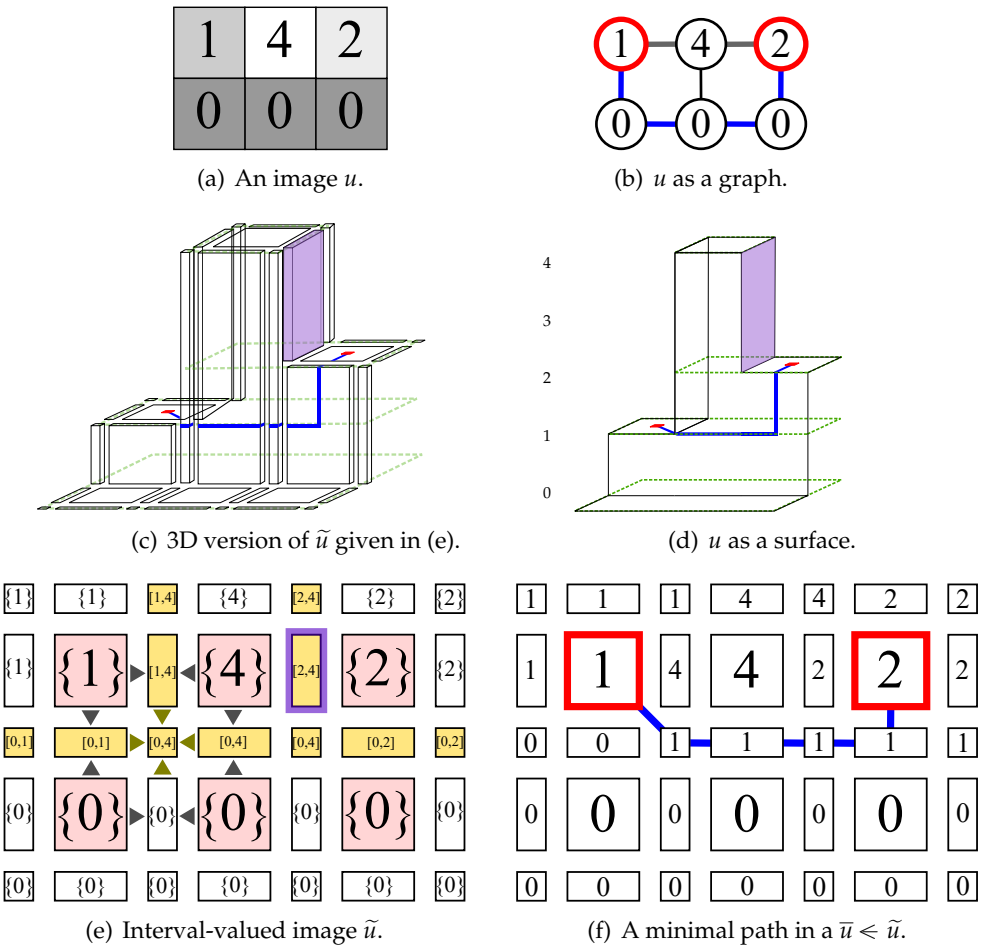
2.6.4 Distance map based on the minimum barrier distance

It is possible to derive a distance map from the MBD (and it is a common usage). Given a minimum barrier strength function and a set X' of seed points, a distance map S^{MBD} can be computed by:

$$S_u^{\text{MBD}}(x, X') = \min_{x' \in X'} d_u^{\text{MB}}(x, x'). \tag{2.34}$$

A distance transform[1] can be used to compute a distance map from a set of seed points X'. Therefore, choosing the appropriate seed points is an essential step to detect objects in the image. Saliency maps presented in [9], in [10] and in [8] are

[1] We call *distance transforms* all the distances or pseudo-distances between two points presented in this thesis, and *distance maps* the family of distances (between a set of points and a point) induced by these distance transforms; saliency maps are an example of distance maps.



(a) An image u. (b) u as a graph.

(c) 3D version of $\widetilde{u}$ given in (e). (d) u as a surface.

(e) Interval-valued image $\widetilde{u}$. (f) A minimal path in a $\overline{u} \Subset \widetilde{u}$.

FIGURE 2.25: Image representations for computing barrier distances.

particular cases of that. In these papers, the seeds are in the border (since the image border is considered to be background). The saliency map is then the distance from every point of the image to the border of the image. For every point, the MBD looks for the optimal path that connects a point to the "nearest" border. This saliency map has already been computed many times relying on many fast MBD (approximation) computations such as raster scan in [9], minimum spanning tree in [10], water flowing in [8].

2.6.5 MB-based distances

The MBD has been firstly introduced in [5] as a minimum value of the barrier strength among the set of possible paths between two pixels in the image. The MBD has been used in several applications in image processing and computer vision, for instance, in salient object detection (see [9, 10, 8, 11–13]), in object localization (see [16]), in superpixel segmentation (see [188]), in interactive segmentation (see [6, 14, 15, 20], refocusing [189]), object proposals generation (see [190]) and in object segmentation (see [191, 192]).

In salient object detection, the goal is to compute a saliency map that highlights the most important objects in the image. To proceed, the *boundary connectivity prior*, which is presented in [22], assumes that boundary regions are usually large, homogeneous and mostly background. The MBD estimates a distance from every pixel

FIGURE 2.26: The MB-based distances are used in salient object detection (see [8]). The left image is the original image, the right one is the saliency map of all pixels in the image by considering that pixels on the border of the image are the background.

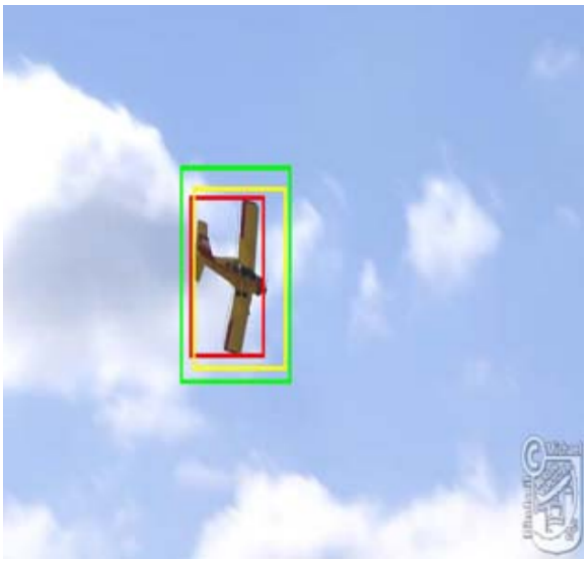
in the image to the image boundary while considering that image boundary is regarded as the background seeds. An example is illustrated in Fig. 2.26.

FIGURE 2.27: The MB-based distances are used in interactive segmentation (see [20]). The left image is the original image and the right image is the result of interactive segmentation.

Besides, the MBD has been also used for interactive segmentation (see [5, 20]), which is illustrated in Fig. 2.27. In this application, the user tags some pixels belonging to the object to segment as foreground, some pixels outside of the object as background and the MBD between these two sets of pixels and all other pixels are computed to deduce the boundary of the object. In [5], the MBD is computed on grayscale images, and its extended color version is presented in [20]. These articles show that this process is robust to noise, blurring and seed point position. Therefore, the MBD seems to have the potential for real interactive segmentation, where a user can manually add/remove seed-points to affect the result.

Many applications take advantage of the relevance of the saliency map computed by the MBD. The classical usage of this saliency map is object segmentation. A saliency map is computed by the MBD and object are segmented according to the saliency map. For example, in [192], an affinity model based on the MBD is used for object segmentation. An example is illustrated in Fig. 2.29(a). Object segmentation is a starting point for multiple other applications. For example, in [16], object detection is extended to tracking (see Fig. 2.28(a)). Another example, exposed in [189], relies on object segmentation to perform a refocusing (see Fig. 2.28(b)).

The MBD has also been used in object proposal generation as presented in [190]. The authors propose a method for locating object proposal based on the MBD, called *MBDSal Box*. This method takes into account the connectivity between the boundary of the sliding window and other regions to distinguish the window that contains



(a) Object localization and tracking.

(b) Refocusing application

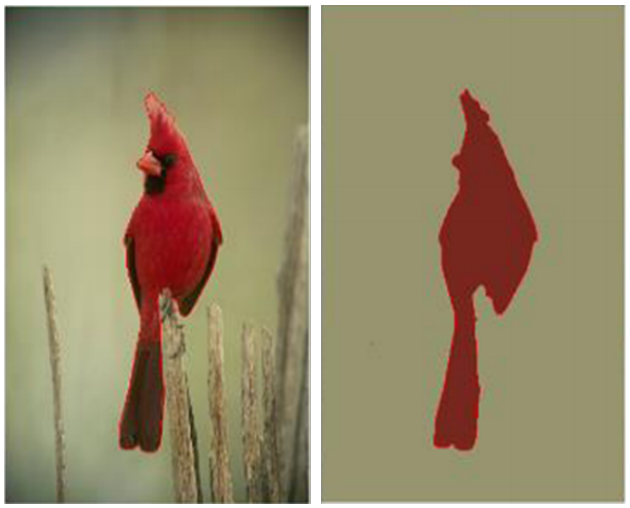
FIGURE 2.28: The MB-based distances are used in object localization (see [16]) and refocusing application (see [189]).

complete objects and incomplete objects. Then, a refinement module is provided to improve the MBD saliency map where the boundary of the window is considered to be background. Finally, a scoring module is used to compute the final objectness in each bounding box area.

Another application is the computation of superpixels. An example is exposed in [188]. The authors propose a method for superpixel segmentation relying on the MBD. k initial centers (seeds) are sampled uniformly and superpixels are determined around them according to "compact-aware MBD ", which is a combination of the original MBD and the (spatial) Euclidean distance. A compactness factor is introduced to control the weight of the spatial distance compared to the barrier distance along the path. An example is given in Fig. 2.29(b).

The MBD is very powerful, however it is difficult to compute the MBD efficiently on an image of a reasonable size. Because computing the exact MBD is too much time-consuming, some approximate methods have been proposed to minimize this drawback and accelerate the algorithm (see [9, 10, 8]).

In [9], the authors proposed a Fast-MBD with a raster scan algorithm, which provides a good approximation of the MBD in several milliseconds on a 3.2 GHz CPU. This method visits each pixel in the direct and inverse raster scan orders, and



(a) Object segmentation.

(b) Superpixels segmentation.

FIGURE 2.29: The MB-based distances are used in object segmentation (see [192]) and superpixels segmentation (see [188]).

then updates the MBD for its neighbors. This salient object detection method runs at about 80 FPS (when single-threaded) and achieves competitive performance with the state-of-the-art saliency detection methods. Despite the fact that it provides good results, the raster scan method has issues when the exact path between two pixels is in a direction between the bottom left and the top right of the image (see [8] for details). Besides, the Fast-MBD only works with seed pixels which are put on the border of the image. Therefore, the Fast-MBD algorithm is not interesting for all usage where the seed points are located into the image as for interactive segmentation, in which we put the seed inside the objects in the image.

Another approximation of the MBD is proposed in [10]; it uses the minimum spanning tree (MST). Firstly, the input image is represented by a tree; paths between pixels correspond to paths between the nodes of the tree. The MST highly reduces the size of the space we look for to find the shortest path between two pixels of the image. Obviously, the MBD computed on the MST needs additional time for the MST construction, but a single tree can be used to compute multiple paths. However, the "simple" structure property of MST can lead to some approximation errors,

especially when noise appears in the image.

Recently, a new algorithm to approximate the MBD has been proposed in [8], which is inspired from the natural phenomena of water flow. The seed pixels which are usually put on the boundary of the image, are assumed to be sources of water. Then, the water is propagated from sources to the neighboring pixels (with different flow costs) until all the pixels are flooded. The MBD can be computed during the flooding process by considering the priority of each pixel during the propagation, which is based on the MBD value from this pixel to the source pixels. The Waterflow-MBD computation achieves a high-speed performance and has comparable results to the other methods. However, it has difficulties when it must compute multiple distances between multiple pixels in the image.

These methods based on the MBD achieve state-of-the-art results with other bottom-up methods on saliency map computation. They can also process an image in real-time, which is relevant for applications with speed requirements. Despite the success of salient object detection, the previous MBD is not well managed on color images (see [9, 10]). Specifically, the MBD is computed separately on each channel and takes the average value. This approach has a limitation because the optimal path on each channel is different. For this reason, Géraud et al. propose a new version of the MBD, named Dahu pseudo-distance, which is also based on the notion of barrier (Eq. (2.32)). We talk about this new distance in Section 2.6.6.

2.6.6 The Dahu pseudo-distance

A new discrete version of the MBD, named the Dahu pseudo-distance is defined in [18] and considers an image (see Fig. 2.25(a)) to be a continous surface in the set-valued sense (see Fig. 2.25(d)) on a discrete topological domain called the Khalimsky grids. Details about set-valued continuity and about Khalimsky grids can be found in [193] and in [194] respectively. The optimal blue path between the two red points is depicted in the image, and has a distance equal to one. It is slightly different from the original MBD. Let us briefly present this Dahu pseudo-distance.

A gray-level image can be seen as a function u: $\mathbb{Z}^2 \to \mathbb{N}$. When we represent an image using a surface, we cannot use scalar functions; we have to use set-valued functions. More exactly, in [19], the authors proposed to replace the domain $\mathbb{Z}^2$ by the topological discrete space $\mathbb{H}^2$ of 2D Khalimsky grids (also known as cubical complexes), and the co-domain $\mathbb{N}$ by the set $\mathbb{I}_{\mathbb{N}}$ of intervals of natural numbers. The 2D cubical complex, which is illustrated in Fig. 2.25(e) is a set of 2D, 1D, and 0D elements, in which 2D elements are the original pixels, 1D and 0D are the inter-pixels which take the interval-valued from its 2D neighbors. For example, the 1D yellow element in Fig. 2.25(e), which is bounded by a purple border, corresponds to the vertical purple part in Fig. 2.25(c). The inter-pixel is actually a transition step between two pixels, which is a way to get a discrete topology and to represent what lies between the pixels. From a scalar image u, we construct an interval-valued image $\widetilde{u}$, which really represents the surface corresponding to u.

The inclusion realtionship between a scalar image and an interval-valued image is denoted by $\Subset$. The Fig. 2.25(f) depicts a scalar image $\overline{u}$ which is "included" in the interval-valued image $\widetilde{u}$ depicted in Fig. 2.25(e); then we can write $\overline{u} \Subset \widetilde{u}$. The adaptation of the MBD on the interval-valued image, called the Dahu pseudo-distance (see [19]), is noted d^{DAHU}. Then the Dahu pseudo-distance between two pixels x and x' on the original image u is defined as:

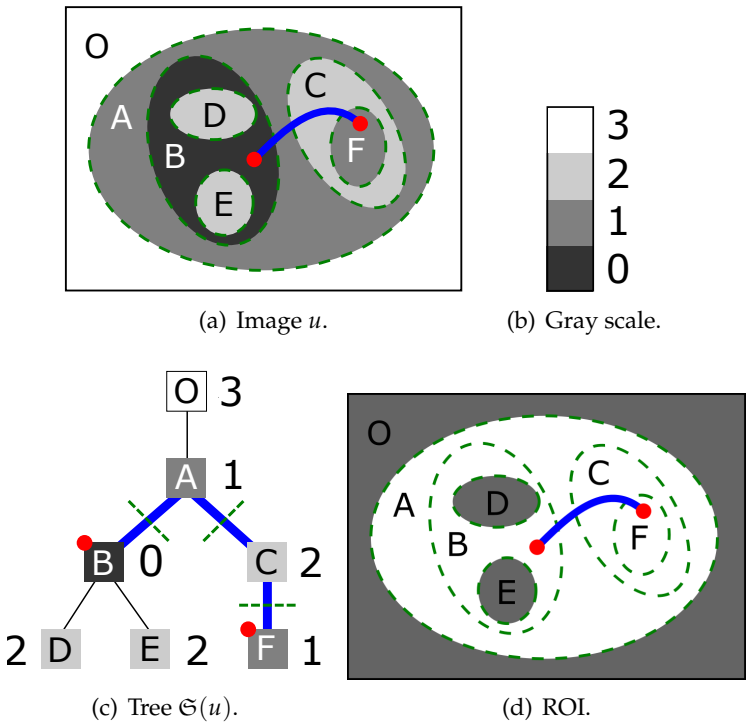
$$d_u^{\mathrm{DAHU}}(x, x') = \min_{\overline{u} \preccurlyeq \widetilde{u}} d_{\overline{u}}^{\mathrm{MB}}(h_x, h_{x'}) \tag{2.35}$$

$$= \min_{\overline{u} \preccurlyeq \widetilde{u}} \min_{\pi \in \Pi(h_x, h_{x'})} \tau_{\overline{u}}(\pi), \tag{2.36}$$

where h_x and $h_{x'}$ are the 2D elements of the cubical complex corresponding to x and x' respectively. It means that we look for a minimal path in the cubical complex, with the classical definition of the MBD, and consider all the possible scalar functions $\overline{u}$ that are "included" in the interval-valued map $\widetilde{u}$. Returning to the earlier example (Section 2.6.3, Fig. 2.25(b)), the shortest path between the two red points in Fig. 2.25(c), depicted as a blue path in Fig. 2.25(f) (image $\overline{u}$ included in the interval-valued image $\widetilde{u}$ that provides the minimal path), has a length of one. The result provided by the MBD (the value 2 Section 2.6.3), is different from the result provided by the Dahu pseudo-distance (the value 1). The Dahu pseudo-distance is thus of combinatorial complexity when we count all the possible scalar images $\overline{u}$ "included" in $\widetilde{u}$. The Dahu pseudo-distance can be interpreted as the *best minimum barrier distance that we can have considering that the input function is continuous in the set valued sense* (see [94]).

Note that, as the MBD, the Dahu pseudo-distance is not a distance, because it can exist some x, y such that $x \neq y$ and $d_u^{\mathrm{DAHU}}(x, y) = 0$.

2.6.7 Efficient Dahu pseudo-distance computation using the tree of shapes



(a) Image u. (b) Gray scale.

(c) Tree $\mathfrak{S}(u)$. (d) ROI.

FIGURE 2.30: The tree of shapes of an image allows to easily express and compute the Dahu pseudo-distance and distance maps (see [19]).

The Dahu pseudo-distance can be computed easily and efficiently thanks to the tree-based representation of the given image. A tree of shapes (see [56, 123]) is a

morphological self-dual representation of an image. This tree is a decomposition of a gray-level image into connected components, called *shapes*, which can be arranged into a tree encoding an inclusion relationship. A shape is a filled-in connected component without hole inside (its boundary is then an iso-level line). Two iso-level lines (at different levels or not) cannot cross each other (under some particular constraints). A very strong consequence is that shapes are either disjoint or nested, which explains that the tree of shapes is a tree and not a graph with cycles. In [18], the authors compute the ToS with assuming that the image has its domain on a cubical grid that allows continuous properties while staying on a discrete space. Therefore, the ToS representation is able to deal with the Dahu pseudo-distance properties.

The tree of shapes is a good representation that facilitates the computation of the Dahu pseudo-distance. The minimal path between two points in the image corresponds to a path between two nodes on the tree. On Fig. 2.30(a), the path between two points (x, x') indicated by red bullets in u is depicted by a blue line, which starts from region B, then goes through A and C, and finally ends in region F. Such a path is minimal because every path in $\Pi(x, x')$ should at least cross this same set of level lines to go from x to x'; thus the Dahu pseudo-distance corresponds to the level dynamics of this set of lines. Actually, this path in the image space is exactly *the* (shortest in number of nodes) path in the tree of shapes between the nodes t_x and $t_{x'}$:

$$\dot{\pi}(t_x, t_{x'}) := \langle t_x, \ldots, \mathrm{lca}(t_x, t_{x'}), \ldots, t_{x'} \rangle, \tag{2.37}$$

where $\mathrm{lca}(t_x, t_{x'})$ is the lowest common ancestor of the pair $(t_x, t_{x'})$ (see the blue path on the tree depicted in Fig. 2.30(c)). Note that a path in a tree is denoted by $\dot{\pi}$ to distinguish it from paths in the image space.

The Dahu pseudo-distance in the image space between two points x and x' can be written as the minimum barrier distance between the two nodes t_x and $t_{x'}$ representing the components in the tree of shape containing respectively x and x':

$$\begin{aligned} d_u^{\mathrm{DAHU}}(x, x') &= d^{\mathrm{MB}}_{\mathfrak{S}(u)}(t_x, t_{x'}) && (2.38)\\ &= \max_{t \in \dot{\pi}(t_x, t_{x'})} \mu_u(t) - \min_{t \in \dot{\pi}(t_x, t_{x'})} \mu_u(t), && (2.39) \end{aligned}$$

where $\mu_u(t)$ denotes the gray-level associated with the node t of the tree of shapes $\mathfrak{S}(u)$ of the image u. For instance, in Fig. 2.30(c), the blue path gives the sequence of node values $\langle 0, 1, 2, 1 \rangle$, so the Dahu pseudo-distance is 2. There is *no need* to find the best scalar image $\overline{u} \Subset \widetilde{u}$, nor to find the best path $\pi \in \Pi(x, x')$ in the image space; it thus means that the seminal definition of the Dahu pseudo-distance (see Eq. (2.36)) is the best choice to be fast in time. The new expression of this distance (see Eq. (2.39)) is just a barrier strength computation (such as Eq. (2.32)) on the trivial path $\dot{\pi}(t_x, t_{x'})$ of nodes in the space of the tree of shapes.

2.6.8 Saliency map based on the Dahu pseudo-distance

A distance map function of an image u can be derived from the MBD as we have seen in Eq. (2.34). Indeed, we can define the saliency map based on the Dahu pseudo-distance in the following manner:

$$S_u^{\mathrm{DAHU}}(x, X') := \min_{x' \in X'} d_u^{\mathrm{DAHU}}(x, x'),$$

where X' is some set of points of the domain of the image u.

Now, let us define the corresponding set of nodes on $\mathfrak{S}(u)$ of X':

$$T_{X'} := \{ t_{x'} ; x' \in X' \}. \tag{2.40}$$

Then, we obtain using Eq. 2.38 and then Eq. 2.34 that:

$$S_u^{\text{DAHU}}(x, X') = S_{\mathfrak{S}(u)}^{\text{MBD}}(t_x, T_{X'}), \tag{2.41}$$

which shows how the saliency map induced by the MBD is related to the saliency map induced by the Dahu pseudo-distance.

2.6.9 Conclusion

In this section, we present a review about distance function, which is used widely in image processing in general and mathematical morphology in particular. The goal of the distance function is to measure the dissimilarity between pixels and regions in the image, thereby providing additional information for the following steps. We present several well-known discrete distances such as city-block, chessboard, Euclidean and geodesic distance. We also introduce the MBD and its continuous version called the Dahu pseudo-distance. The MB-based distances show that they are robust to noise, blur and seed point positions in the image. However, these distances are not well-handled for color images. In Chapter 3, we propose a method to extend this distance to multivariate images.

2.7 Visual saliency detection

"Everyone knows what attention is..."
William James, 1890

With the development of camera devices and social networks, a massive amount of images are appeared in our daily life activities. It leads to the need for a powerful technique that automatically gets the most useful messages from these images and filters out unnecessary information in a short time for further processing. This problem brings us to the idea of visual saliency detection, which is a technique to simulate the human perception of an image. Human eyes are capable to focus on specific objects in the image with different priorities than others even at first glance. Research on human visual saliency is a good way to understand the scene and to collect more information for object detection and recognition process.

In computer vision, modeling visual saliency on images is referred to as saliency detection or salient image regions detection. The result of the visual saliency detection, simply called saliency map, is an intensity image where regions with high-intensity value indicate the most important objects in the image. The visual saliency detection is categorized into two classes: eye fixation modeling and salient object detection. Early approaches in visual saliency detection belong to the former [195–197]. Although eye fixation modeling has gained a lot of progress since then, this approach also has a drawback. In the case of the large object, it usually produces a sparse map, since it highlights the edges and corners of the object instead of the whole. The latter one, salient object detection, is employed in many applications in computer vision. In contrast to predicting eye fixations, the goal of salient region detection is to detect and segment entire salient objects in a scene. In other words,

(a) Image. (b) Ground truth of eye fixation. (c) Ground truth of saliency detection.

FIGURE 2.31: An example of visual comparison between eye fixation modeling and saliency detection.

salient object detection is equivalent to the foreground/background segmentation problem. An example of the difference between eye fixation modeling and saliency object detection is illustrated in Fig. 2.31.

Visual saliency detection methods are exploited in various computer vision applications, such as object detection and recognition [198–200, 196], image and video compression [197], content-based image editing [201, 202] and image retrieval [203, 204]. Moreover, there are several well-known datasets, which are used for saliency detection applications. Input images with the annotations are shown in Fig. 2.32.

The salient object detection methods are categorized into two types based on their different strategies [205], namely bottom-up and top-down. The former methods are relied on several assumptions about the objects and background in the image, without taking into account the knowledge of the image. On the other hand, the latter methods require the prior information and the class of object in the image, consequently, need high computational costs.

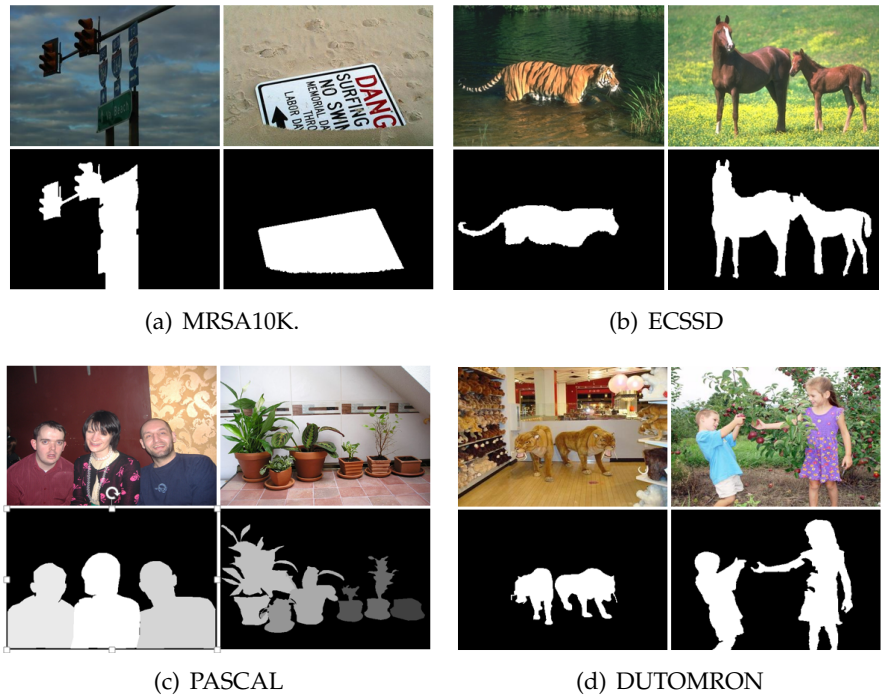
The salient object detection in this thesis is based on bottom-up approaches because of its efficiency in both run-time and quality.

2.7.1 Bottom-up methods

The bottom-up methods are studied for non-specific saliency detections task. The result is computed based on the input image itself without using any knowledge about the kind of objects and backgrounds in the image. This method usually uses some assumptions about the different contrast between the objects and the background, or the position of the objects. The very first bottom-up saliency model is proposed in [195]. This method is inspired by the concept of feature integration theory (FIT) [206] and visual attention [207]. Following the Itti's paper, many bottom-up methods have been proposed. In the following, we present several assumptions that are used in the bottom up saliency detection methods.

2.7.1.1 Contrast prior

Contrast is used as an important feature in many visual saliency models. This feature relies on the assumption that the background is homogeneous, and there is a high contrast between objects and the background. In [202], the authors proposed a method (context-aware saliency) to detect salient objects in the image. The idea



(a) MRSA10K. (b) ECSSD

(c) PASCAL (d) DUTOMRON

FIGURE 2.32: Images and pixel-level annotations from four salient object datasets.

is that salient objects are distinctive with respect to both their local and global surroundings. Therefore, they computed local and global saliency by considering the contrast of each patch with the K most similar patches in the image concerning their position distance. The multi-scale model is used to incorporate the local and global saliency, which leads to a decrease in the saliency of background pixels.

In [208], the authors proposed a frequency tuned method to compute the saliency map by considering the difference between objects and the average image color of all pixels. It also eliminates fine texture and noise in the image. The boundaries of the object are highlighted by keeping more frequency content from the original image.

The global contrast differences and spatial information are exploited in [209] to compute the saliency map. Their method begins with partitioning image into regions using a graph-based image segmentation method [31]. The color histogram of each region is computed to analyse the color statistic in the image. Then the contrast saliency of a region is computed by considering the distance between its color to all regions in the image. The spatial information is involved to increase the effects of closer regions.

Since calculating the contrast and spatial saliency in a pixel-wise image is costly, superpixel segmentation methods are adopted. The SLIC [25] and FH [31] methods are used widely in many visual saliency applications [209–211]. In [210], after segmenting the image into multiple partitions, their method (saliency filter) derives a saliency map from two contrast measures based on the uniqueness and spatial distribution of the superpixels. Then these two measures are combined and normalized to get the final result.

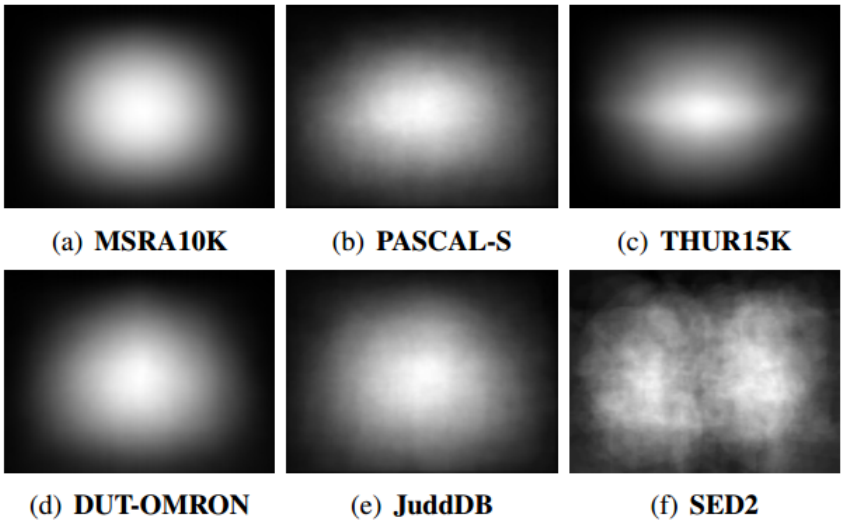


FIGURE 2.33: Average annotation maps of six datasets used in benchmarking. Images taken from [212].

2.7.1.2 Center Prior

Center prior is firstly introduced in [213]. This research has been shown that the human eye is always biased to the center of the image. In other words, objects in the center of the image usually get more attention than other objects. In general, when people take pictures, the objects are often placed near the center of the image. In [214], an average of human saliency map is calculated from 1003 images. The result shows that 40% of saliency maps are appeared in 11% near the center of the image, 70% of saliency maps appear in 25% near the center of the image. Also, in this paper, the authors demonstrate that a Gaussian center is a good choice to simulate the center prior for visual saliency map. Fig. 2.33 shows the distribution of the object positions in the images. This figure of center bias is studied in a survey of Borji et al [212].

The center prior can be used directly to compute the saliency map. An example is the uniqueness term in [210]. It is also used in many saliency detection methods as a post-processing method to refine the saliency map [9, 8, 202].

2.7.1.3 Boundary and Connectivity Prior

The center prior is a good way to compute the saliency map, however this prior fails when an object is placed not in the center but at the corner or near the border of the image. Therefore, another approach that has been proposed in visual saliency detection is boundary and connectivity prior as we already talk about it. Different from previous prior, which focuses on the object position, this prior concentrates the background of the image. The boundary prior comes from a rule in photographic that objects do not cut off the image borders. Even the object touches the border, the border pixels are still mostly background. This prior is similar to the bounding box prior that we usually use in interactive segmentation application [215]. The second prior comes from the fact that the background image is connected with the border of the image. Usually, background regions large and homogeneous, consequently they are easy to connect with each other. The first paper that applies this prior is [22]. The authors consider pixels in the border of the image, as the background. Then they compute the distance between every pixel with the border of the image. The

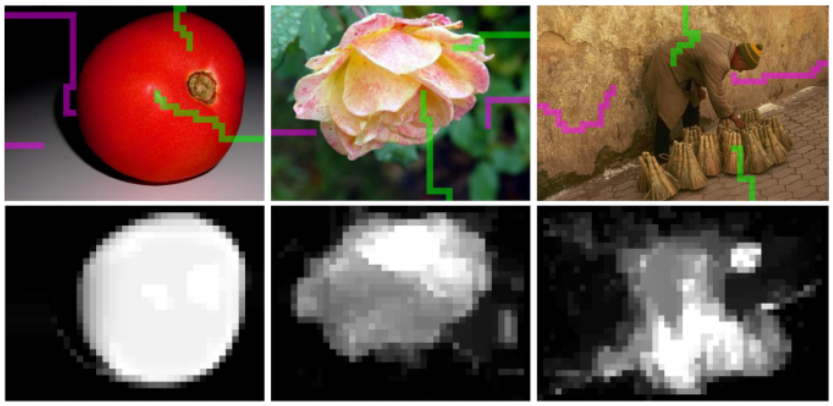


FIGURE 2.34: Examples from [22] showing the paths of background (in magenta) and foreground (in green) from the boundary in the top row. Bottom row shows saliency maps retrieved by their algorithm.

saliency map is computed by using the path-wise distance, namely geodesic distance. This method resorts to searching the shortest path between each region to the image border. The background prior is also used in [141] to calculate background connectivity.

These methods work well in case of objects do not touch the border of the image. However, this prior has a problem when objects partially touch the image border. To deal with this problem, several methods have been proposed. In [140], the authors sequentially compute the saliency map between every superpixel with the top, bottom, left and right border and then combine it to get the final saliency. Another approach is to calculate the boundary edge weight before computing the geodesic distance in [22]. In [8], the authors proposed a method to modify the image border. Fig. 2.34 shows the saliency maps, which are computed with taking into account the boundary image. Also, this prior can be combined with other contrast-based methods to deal with this problem. The details of this method can be found in [9, 10].

2.7.1.4 Graph-based Approach

As presented in the previous section, image elements are considered to be nodes in the graph. The graph-based approach is also used in visual saliency detection. In this section, we present three principle approaches, namely random walk [216, 217], manifold ranking [140], and combinational optimization framework [141].

In [216, 217], the authors used the random walk method to solve the problem of salient object detection. The seed nodes which belong to the salient object and background regions are obtained by using the global and local properties of salient regions. The "pop-out" graph model and the seed nodes are used to learn the salient object in the image.

Manifold ranking method [140] ranks each of the superpixels based on the similarity between superpixel with image background and foreground. The top, bottom, left and right border are sequentially assigned as queries, then the optimal ranking of queries are computed. The saliency of the image regions is defined based on their relevances to the given seeds or queries.

Another graph-based method to compute the saliency map is combinational optimization approach. This method generates an optimized function from different saliency cue maps [140, 218, 219, 11, 220, 187]. Different cue that is used in these

models can be contrasted prior, spatial prior, boundary prior, object prior or multi-scale segmentation. The optimize formulation usually has two terms: the first term is the fitting constraint maintaining the initial query assignment, and the second term is the smoothness constraint aiming at preserving the similarity between adjacent regions [220].

The bottom-up methods are efficient and work without any knowledge about the object and the background. In this thesis, we only focus on this approach.

2.7.2 Top-down methods

The bottom-up models are efficient, but they have limited use because they do not consider image semantic. The top-down approaches, on the other hand, are goal-oriented, which depend on the application. These methods require prior knowledge of the scene context and also the information about the class of object in the figure. They are task-driven, and involve a complete understanding of the image. These methods usually divide the saliency detection task into multiple operations such as object detection, object segmentation and classification. Therefore, these methods need high computational costs. However, with the development of the processing unit (GPU), these difficulties can be dealt. Nowadays, top-down methods or supervised learning approaches are widely used in studying the presence of salient object from images. The concept of "learning to detection" is presented firstly in [214, 221]. The knowledge comes from memories, which are collected from the training data.

In [214], they create a dataset of eye-tracking data from 15 users across 1003 images. They propose a combination of different feature levels from low to high to model the salient objects. An SVM learning model is used to train this saliency dataset. The SVM approach is also used in [222] to detect objects of interest. Tong et al. [223] propose a bootstrap learning model, which are combined of weak and strong saliency map to detection salient objects in the image. Firstly, weak saliency maps are computed using contrast and center-bias prior to generate a training dataset for a strong classifier. Then a multiple kernels SVMs are used to measure saliency features. Then the weak and strong saliency maps are weighted to generate the final saliency map.

In [221], they proposed a saliency detection method by using Conditional Random Field (CRF) model. The idea behind this method is considering salient object detection as an image segmentation, where we segment foreground object out of background regions. A set of features are used including multiscale contrast, color histogram, and spatial information. The Conditional Random Field is also used in [224] to aggregate individual saliency from different methods. Weights for aggregation are learned in a data-driven way from most similar images retrieved from a pre-defined dataset. In the work of Yang et al. [198], a top-down saliency model based on image patches is computed by jointing Conditional Random Field and visual dictionary.

Alex et al. [225] propose a method to score objectness measure inside sample windows. This method combines several image cues from multi-scale features. In [226], a method which is implemented on the superpixel images is proposed to detect saliency based on a dataset with manually marked salient objects. The initial segmentation after using classifier models is refined by a graph-cut optimization.

Recently, along with the success of deep learning and other high level features extraction, saliency detection has been achieved significantly good performances.

In [227], Wang et al. compute a saliency map based on local estimation and global search. A deep neural network (DNN-L) is used to estimate the local saliency map of pixels. The local map with the global contrast is used to predict the saliency score of each region in the image by using another deep neural network (DNN-G). Zhao et al. [228] propose a multi-context Convolutional neural network for saliency detection. The global and local context are integrated into this deep learning model. The input of this network is the superpixel image, which is partitioned by the SLIC algorithm. Chen et al. [229] propose deep learning saliency computing framework (Disc) to compute the saliency map. Their model is built upon two stacked CNNs. The first CNN produces coarse saliency map based on the global context. The second one focus on the detail of the object.

The top-down methods involve the knowledge about the object and the image. Therefore their performances are much better than the bottom up methods in case of low contrast and complex background image.

2.8 Document detection

In nowadays world, the demand for using digital document is increasing because of its convenience in searching, storing, retrieving, etc. A traditional way to digitize paper is using a scanner machine, which is heavy, costly, and usually not portable. With the development of smartphone cameras, many people use them to acquire documents. Digitizing papers in images or videos captured by smartphones is not the same procedure as scanning: images captured by smartphones do contain a background. Therefore, the first step of the digitization process is the extraction of the document region from the scene. In this thesis, our goal is to segment automatically documents in an acceptable run time.

Images taken by smartphones can pose many challenges to the digitization process. The scene contexts are unknown, the lighting conditions are variable, and the illumination is not homogeneous. Images can be noisy. Moreover, the camera is handheld, this can lead to out-of-focus or motion blur.

Document detection in images captured by smartphones is important for later steps. That is the reason why challenge 1 of the ICDAR 2015 Smartdoc competition [30] focuses on the evaluation of the document detection and segmentation algorithms. Eight submissions were made and these eight methods can be classified into two categories according to the used strategy: The most common strategy is to rely on lines detection and the other is the hierarchical tree-based representation of the image. Seven over the eight methods extract lines in the image as candidates for document segmentation. The Canny edge detector [230], Hough transform [231], and LSD algorithm [232] are adopted to detect the lines in the image. Although it is the most common strategy, these methods cannot work well if the document is curled.

Among them, two methods outperform others. The ISPL-CVML method uses the LSD algorithm to get vertical and horizontal segments on the down-sampled image. Then the color and edge features are exploited to select document boundaries. SmartEngines method [27] uses several algorithms to detect segments in the image, then builds a graph of these segments. A quadrangle of a possible document is constructed from this graph while considering the weights and angles of edges. The final quadrangle is obtained after applying a Kalman filter based on some local descriptors.

The hierarchical tree-based representation method, LRDE method [26], gets the highest score. They compute the energy of each node on the tree, which consists of two terms measuring how the shape fits the quadrilateral form and how “noisy" the object is (text lines and figures, etc) and then select the best candidate. Nevertheless, this method is slow.

Besides these methods, in the literature, there are also many proposed methods. Smart IDReader [233] method combines a series of the algorithm depending on the class of documents. A Viola-Jones method is applied as a decision tree of strong classifiers for document detection [234].

Geodesic Object proposal [28] method starts with using six seeds to cover all of the objects in the image. The sign geodesic distance transform computed from each seed which is specified with an image region, is then evaluated for being the best document candidate.

Recently, a novel CNN-based method [235] has been proposed, which considers the document localization problem as finding four corners of the document. The AlexNet architecture is used to predict four corners of the document, then refine each prediction by using a shallow convolutional neural network. However, it has difficulties dealing with occlusion or when the document touches the image boundary.

As presented above, the classical approaches have some limitations in dealing with challenging images, such as, blurred images, non-straight boundaries document, or partially occluded document. To overcome these problems, we present in this thesis a region-based approach. A key feature of our method is that it relies on visual saliency, using a recent distance existing in mathematical morphology (the Dahu pseudo-distance).

2.9 Conclusions

In this chapter, we have reviewed several fundamental concepts that relate to the context of our works. Specifically, we present two classes of hierarchical image representation, which are employed to solve many problems in image processing and computer vision. Moreover, we introduce the idea of using the distance function, explicitly, the Dahu pseudo-distance on the tree-based representation. This new distance is proved to be powerful against the pixel fluctuation. It is also the cornerstone of our thesis. Chapter 3, Chapter 4, and Chapter 5 are devoted to giving us a new point of view about this distance, analyzing the properties of this distance and using it in several proposed frameworks. In addition, we propose to extend this distance on multivariate images (color or multimodal/multispectral images). Finally, we use this new distance in a specific application: document detection. Our proposed method is able to achieve high performance compared to many state-of-the-art methods.

Chapter 3

Dahu pseudo-distance improvements and applications

In previous sections, we have introduced the MBD and the Dahu pseudo-distance. There exist many useful applications of the MB-based distance. However, they have a lot of limitations, and especially color images (or more generally multivariate images) are not well handled (or not handled at all). Therefore, we introduce in this chapter an efficient method for computing the Dahu pseudo-distance on multivariate images. Besides, we also propose several frameworks based on the Dahu pseudo-distance for image processing applications.

This chapter is the fundamental contribution of our work. It is divided into two sections: **Dahu pseudo-distance's improvements** and **applications**.

- **Improvements of the Dahu pseudo-distance**: Section 3.1 presents the improvement of the Dahu pseudo-distance in speed performance and an extension of the Dahu pseudo-distance to multivariate images. Explicitly, we provide a method to efficiently compute the Dahu saliency map while constructing the tree of shapes. In addition, we propose an efficient extended version of the Dahu pseudo-distance to color and multivariate images.
- **Applications based on the Dahu pseudo-distance**: We propose several frameworks based on the Dahu pseudo-distance in Section 3.2. Initially, the shortest path application based on the Dahu pseudo-distance is presented by using a two-steps procedure that takes into account at the same time the domains of the tree of shape and of the initial image. This measure is related to the topographical representation of the image. Moreover, we employ the Dahu pseudo-distance on several applications, for example, salient object detection, shortest path finding or object segmentation. Finally, we apply the Dahu pseudo-distance for a particular case of object detection: document detection.

3.1 Dahu pseudo-distance improvements

The Dahu pseudo-distance, which inherits the properties from the Tree of Shapes (ToS) (see [123]), has been shown to be robust to noise and blur effects in the image. This section is dedicated to ameliorate speed performance of the Dahu pseudo-distance and to extend the Dahu pseudo-distance distance on color images.

3.1.1 Improvement of speed performance: simultaneous computations of the Dahu pseudo-distance and the tree of shapes

To obtain the Dahu distance map w.r.t the set of seed points, we have to construct the ToS and compute the map by using Eq. (2.41). However, in the special case of salient

Algorithm 6: Modification of the sorting procedure of the tree of shapes to compute the Dahu pseudo-distance.

Data: Image U, Image domain D, Point p_∞
Result: Dahu pseudo-distance

```
1  begin
2      for all h do
3          deja_vu(h) ⟵ false
4      i ⟵ 0;
5      PUSH(q[l_∞], p_∞) ;
6      deja_vu(p_∞) ⟵ true;
7      l ⟵ l_∞;
8      Image2d min_im, max_im, Dahu;
9      min_im (p_∞) ⟵ l , max_im (p_∞) ⟵ l , Dahu (p_∞, p_∞) ⟵ 0 ;
11     while q is not empty do
12         p ⟵ PRIORITY_POP (q, l) ;
13         u^♭(p) ⟵ l;
14         R[i] ⟵ p ;
15         for all n ∈ N(p) such as deja_vu(n)== false do
16             l' ⟵ PRIORITY_PUSH(q, n, U, l) ;
17             deja_vu(n) ⟵ true ;
18             min_im (n) ⟵ min_im (p) , max_im (n) ⟵ max_im (p) ;
19             if l' < min_im (n) then
20                 min_im (n) ⟵ l'
21             if l' > max_im (n) then
22                 max_im (n) ⟵ l'
23         i ⟵ i + 1 ;
24     for all p ∈ D do
25         Dahu (p_∞, p) ⟵ max_im (p) - min_im (p) ;
26     return(R, u^♭, Dahu)
```

object detection, which is based on the "boundary and connectivity" prior, the Dahu distance map can be computed while constructing the tree of shape. As a result, we can improve the execution time of distance map computation. The boundary prior is introduced in [22], which assumes that the border of the image is mostly background. Similar to previous works (see [9, 10, 8]), we compute the distance map, which is the Dahu pseudo-distance of every pixel in the image according to the border of the image.

The construction of the tree of shapes of the gray-level image is mentioned in [18]. Our algorithm (see Algo. 6) is a modification of the sorting procedure used to compute the tree of shapes: we add some operations (see the blue lines) to the pixel-sorting procedure (of $u^\flat$) during the tree construction (the green lines are used to compute the data exclusively needed to the computation of the tree of shapes). Classically, the root node of the tree represents the whole image, and a saturation operator, sometimes called *cavity fill-in operator*, is implemented to compute later the *shapes*. We add an artificial border surrounding the image domain, in which we set the point p_∞. Only one step remains to be able to proceed to the front propagation: we must input the set-valued map U computed thanks to a span-based interpolation

on the image u. Then, we call the sorting procedure described in [18]. This procedure is based on the handling of a hierarchical queue, denoted by q; the current level is denoted by l. The Dahu pseudo-distance of the starting point is set at the value $l_\infty = 0$. Since we use interval-valued maps, we have to decide at which level to enqueue those elements. The face h is enqueued at the value of the interval $U(h)$ which is the closest to l, which is denoted l' (see the produce PRIORITY_PUSH). The value l' is compared with the minimum and maximum values of its neighbors to update the Dahu pseudo-distance. When the queue $q(l)$ at the current level is empty, the procedure PRIORITY_POP decides whether the next level to be processed is less or greater than l. This loops continues until all of the pixels have been visited. The resulting pseudo-distance is then obtained. More information about the PRIORITY_PUSH and PRIORITY_POP procedures can be found in [18]. Note also that to finally obtain the tree of shapes, three procedures must be executed (see Algo. 3 in [18]), but we will not go into details.

When the seed pixels are not placed in the outer boundary of the image (for example, if they are placed at the center of the image), we need to build the tree of shapes first, and then we can compute the Dahu pseudo-distance. The major difference with a classical saliency map, defined in the image space (such as the one of Eq. (2.34)), is that the tree structure is one-dimensional. Since the Dahu pseudo-distance on the tree (given by Eq. (2.39)) has the form of a barrier "max - min", the saliency map $S^{\mathrm{MBD}}_{\mathfrak{S}(u)}$ at each node t_x can be easily computed by a propagation method on the tree using a priority queue. Besides, this saliency map can also be computed by a two-steps procedure (here, downwards and then upwards) like the classical computation of a *Chamfer distance map* (see [236]). Afterwards, getting the 2D saliency map S^{DAHU}_u means reading for each x the value of $S^{\mathrm{MBD}}_{\mathfrak{S}(u)}$ at t_x. Eventually, once we have computed the tree of shapes $\mathfrak{S}(u)$, the computation of a saliency map $x \mapsto S^{\mathrm{DAHU}}_u(x,\, X')$ is immediate (whatever the set X').

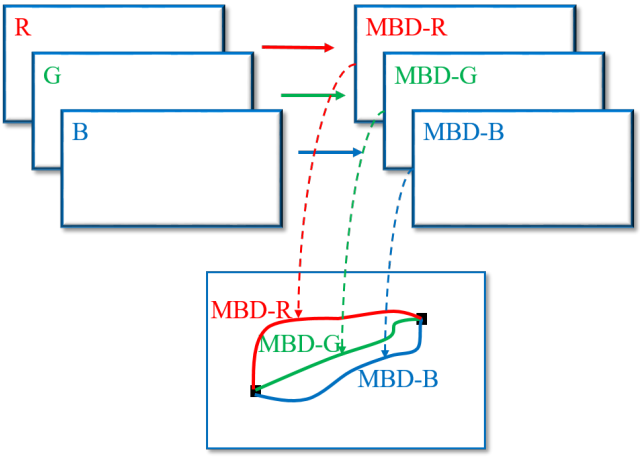
Last, let us mention that the representation of an image into a tree of connected components is easy to handle (see [114]). Furthermore, the tree of shapes of an image can be computed in quasi-linear time w.r.t. the number of pixels (see [18]), and can be parallelized (see [125]).

3.1.2 Extending the Dahu pseudo-distance to multivariate images

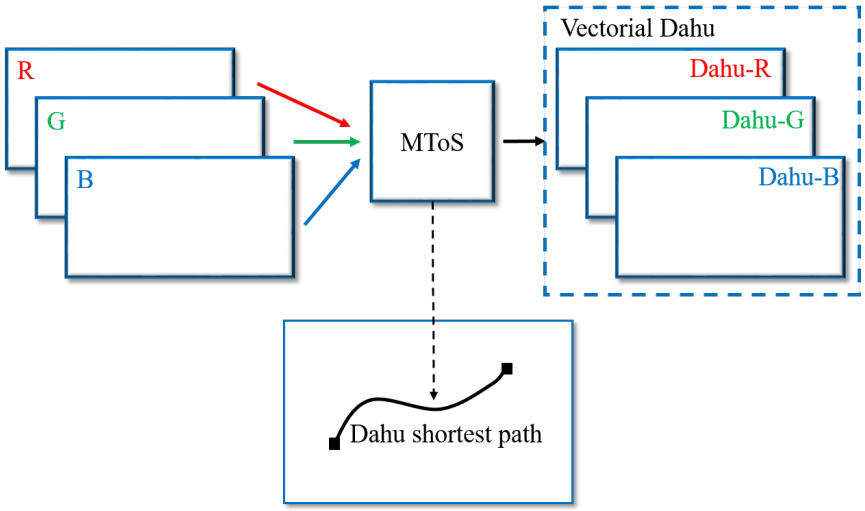
As mentioned before, the previous MBD methods (see [9, 10]) are only defined on grayscale images or on separate channels of color images. In this last case, they compute the mean or the maximal value of the distances obtained on each separate channel, see [10] for details. This approach is not satisfying for image segmentation purpose: we generally obtain different paths for each color, and then computing the mean or the max value of the distances has no sense and cannot be used for image segmentation. An example of the computation of the MBD is illustrated in Fig. 3.1(a).

In [20], a vectorial minimum barrier distance (VMBD) is proposed to compute the MBD on a multivariate image. This distance is a volume of the minimal bounding box which contains all pixels color on the path between two pixels. However, this VMBD is not easy to compute directly on the image. Moreover, the VMBD is not effective when computing multiple distances between multiple points in images. To solve this problem, in this section, we present a Dahu pseudo-distance extended to multivariate images based on the tree space.

The tree of shapes, primarily defined on gray-level images, has been recently extended to multivariate data (see [17]); this extension is called the *Multivariate Tree*



(a) A procedure to compute the MBD and their shortest paths in the color image when processing separately each channel.



(b) A procedure able to compute the vectorial Dahu pseudo-distance. Even with color images, our method is able to obtain a coherent shortest path between two pixels in the image.

FIGURE 3.1: The computation of the MBD and the vectorial Dahu pseudo-distance in a color image.

of Shapes (MToS). It yields a tree mapping the inclusion relationship of shapes in the image. Such a representation is of prime importance for computer vision (see [237]) because it satisfies strong invariance properties featured by natural images, such as self-duality and local contrast changes (see [238]).

However, the definition of the Dahu pseudo-distance on the tree of shapes (see Eq. (2.39)) cannot be used without modification/improvement. Let us now consider that $\boldsymbol{u}$ is a multivariate image, t is a node of the MToS of $\boldsymbol{u}$, and $\mu_{\boldsymbol{u}}(t)$ is the vector value associated with the node t. The superscript i indicates which one of the N components of the vector is taken into account. We can then extend the Dahu pseudo-distance like this:

$$d_{\boldsymbol{u}}^{\text{DAHU}}(x, x') := \sum_{i\in\{1..N\}} \alpha_i \, \tau_{\boldsymbol{u}}^{(i)}\big(\dot{\pi}(t_x, t_{x'})\big). \tag{3.1}$$

with:

$$\tau_u^{(i)}(\dot{\pi}) := \max_{t \in \dot{\pi}} \mu_u^{(i)}(t) - \min_{t \in \dot{\pi}} \mu_u^{(i)}(t), \tag{3.2}$$

where α_i is the coefficient weighting each channel. Note that $\dot{\pi}$ denotes a path between two nodes on a tree, which is expressed in Eq. (2.37).

Algorithm 7: Computation the Dahu pseudo-distance between two pixels in the image.

Data: Image U, Image domain D, Point x, x'
Result: Dahu pseudo-distance

Compute($MToS(u)$);
Compute($t_x, t_{x'}$);
Compute($\text{lca}(t_x, t_{x'})$);
Compute($\dot{\pi}(t_x, t_{x'})$);
for $i \in [1, N]$ **do**
  Compute($\min_{t \in \dot{\pi}} \mu_u^{(i)}(t)$);
  Compute($\max_{t \in \dot{\pi}} \mu_u^{(i)}(t)$);
  Compute($\tau_u^{(i)}(\dot{\pi}(t_x, t_{x'}))$);
end
Compute($d_u^{\text{DAHU}}(x, x')$);
return($d_u^{\text{DAHU}}(x, x')$)

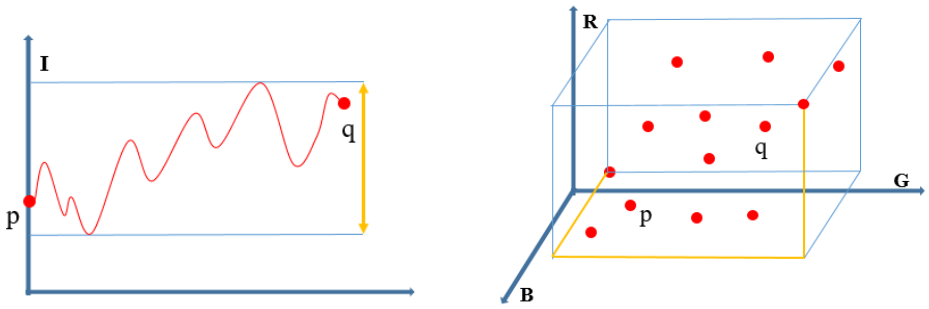
In other words, the vectorial Dahu pseudo-distance between two points x and x' in the domain of the image u can be computed using Algo. 7. The algorithm begins with the computation of the $MToS$ of the image u. Then the node t_x and $t_{x'}$, which correspond to the pixel x and x', are found in the $MToS(u)$. In the next step, the algorithm searches for $\text{lca}(t_x, t_{x'})$, the lowest common ancestor of the nodes t_x and $t_{x'}$, thereby generating the sequence of nodes $\dot{\pi}(t_x, t_{x'})$ that connects these two pixels. The maximum and minimum along the path are updated to get the barrier strength on each channel. Finally, the $d_u^{\text{DAHU}}(x, x')$ is computed with respect to Eq. (3.1). This algorithm is simple and easy to understand.

In [20], several path cost functions are presented, such as the maximum diameter, the city-block diameter, and the volume of the bounding-box methods. The city-block diameter function gives the best performance in several experiments. Since they consider that the importance of each channel is equivalent, we propose to proceed like them and to fix:

$$\alpha_i = 1/N. \tag{3.3}$$

Then, for RGB-color images, our equation becomes:

$$d_u^{\text{DAHU}}(x, x') = \frac{1}{3} \sum_{i \in \{R,G,B\}} \tau_u^{(i)}(\dot{\pi}(t_x, t_{x'})). \tag{3.4}$$



(a) Dahu in the grayscale image. (b) Dahu in the color image.

FIGURE 3.2: The Dahu pseudo-distance in the grayscale image and in the color image.

Please note that, although Eq. (3.1) looks simple, we have here a strong result. To be able to compute saliency maps (based on the MBD) in an efficient way while taking into account colors, we need to compute a particular distance between two points. This distance is one of the *optimal* paths between two points in the image space; this path is such that the set of colors on the path has the smallest bounding box in the color space. On the contrary to the Dahu pseudo-distance in the grayscale image, which is illustrated in Fig. 3.2(a), the vectorial Dahu pseudo-distance is illustrated in Fig. 3.2(b) as the length of the yellow line.

More precisely, the distance between the two points is the city-block diameter (with the L^1 norm) of this 3D bounding box. This is a highly combinatorial problem, far to be trivial, and which cannot be solved efficiently in the image space. Our contribution here is to turn this problem into an efficient and straightforward computation in tree space. Therefore, we can easily compute the Dahu saliency map by using the propagation method proposed in [8] to our MToS structure.

The MToS is computed from the ToS of each image channel by merging some marginal shapes. Due to the MToS properties, it is not a complete representation of an image. The node of the final tree is associated with multiple values of the image. Therefore, a node is assigned to a single value computed from the set of values it contains. In our case, we set each node in the MToS using the median value of its pixels. The vectorial Dahu pseudo-distance computed on the color image is not the exact distance between two points in the image. Nevertheless, this approximation is still useful for some applications in computer vision and image processing. Some results will be shown in the next section. The whole process to compute the vectorial Dahu distance is illustrated in Fig. 3.1(b). This way, we obtain a "coherent" shortest path between two pixels in the image (see Fig. 3.1(b)).

We have illustrated our vectorial Dahu pseudo-distance on classical $R.G.B.$ color images. However our vectorial Dahu pseudo-distance is not restricted to 3 channels and is fully usable on any kind of multi-channels images like multi/hyper-spectral images thanks to the coefficient in Eq. 3.3. We will illustrate this point further on satellite multi-spectral images and even on medical multimodal images.

3.2 Dahu pseudo-distance applications

In this section, several applications based on the Dahu pseudo-distance are presented. Firstly, we combine the Dahu pseudo-distance with spatial information to

search for the shortest path between pixels in the image. Secondly, an efficient propagation method in the MToS, which is inspired from the water flowing method [8], is presented to compute efficiently the Dahu saliency map in color images. Thirdly, we propose an interactive segmentation method based on the Dahu pseudo-distance by using a statistical approach to study the background and foreground information from the scribbles. Fourthly, the Dahu pseudo-distance is employed for image segmentation. Our approach belongs to the hierarchy image segmentation class that we have presented in Section 2.4.2. Finally, our new distance is applied to a document detection application in videos captured by smartphones.

3.2.1 Shortest path finding based on the Dahu pseudo-distance

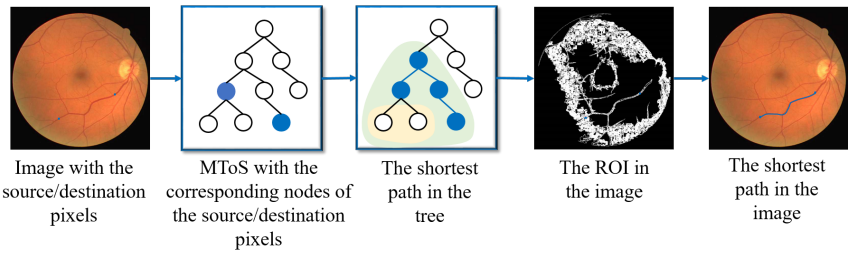


FIGURE 3.3: A scheme for shortest path finding application.

In this section, we present an improvement of the Dahu pseudo-distance by involving the spatial information between two pixels in the image. This improvement is actually a "two-steps" procedure, which is illustrated in Fig. 3.3.

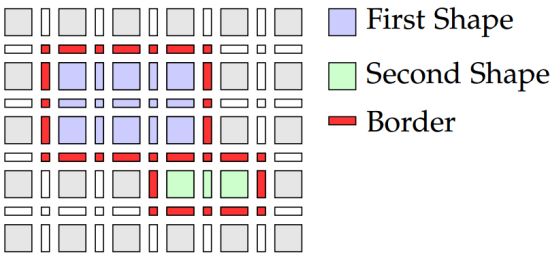


FIGURE 3.4: Shapes on the cubical complex. Image is taken from [130].

First, let us recall the topological consideration of the ToS. In [18], the ToS is computed from the 2D cubical complex that represents the continuous properties. An example of the shapes on the cubical complex is illustrated in Fig. 3.4. As we can see, the shape A on the tree is an open set which are composed of 0D, 1D, and 2D elements. We define $\delta(A) = A \cup \partial A$ as the closure operator of the shape A, where ∂A is the boundary of the shape A, which is depicted as the set of red 1D and 0D elements in Fig. 3.4.

Considering two given pixels x and x', in the first step, we look for the shortest path in the sense of the Dahu pseudo-distance in the tree space between two nodes t_x and $t_{x'}$, which correspond to these two given pixels (see the blue nodes on the tree depicted in Fig. 3.3). We denote $par(t_x)$ as the parent node of node t_x in the tree, and $lca(t_x, t_{x'})$ as the lowest common ancestor of the nodes t_x and $t_{x'}$. The shortest path $\dot{\pi}(t_x, t_{x'})$ between two nodes t_x and $t_{x'}$ is the sequence of nodes that begins from

node t_x, goes through the lowest common ancestor $lca(t_x, t_{x'})$, and ends at the node $t_{x'}$. When we have $t_x \neq t_{x'}$, the shortest path $\dot{\pi}(t_x, t_{x'})$ can be formulated as follow:

$$\langle t_x, par(t_x), \ldots, lca(t_x, t_{x'}), ..., par(t_{x'}), t_{x'}, \rangle \tag{3.5}$$

otherwise it is the trivial path $\langle t_x \rangle$.

Note that each node t_x on the tree represents a connected component $CC(t_x)$ in the image domain. We denote $\Re^*(t_x)$ as the region which is the union of connected components that correspond to descendants of the node t_x, and $\Re(t_x)$ is the union of $\Re^*(t_x)$ and the connected component $CC(t_x)$ of the node t_x itself. After computing the shortest path $\dot{\pi}(t_x, t_{x'})$, we need to find a region in the image space that connects two pixels x and x', which is deduced from the shortest path $\dot{\pi}(t_x, t_{x'})$ in the tree space. We call this region as $ROI(t_x, t_{x'})$. The method that computes $ROI(t_x, t_{x'})$ directly from the union of connected components, which correspond to a set of nodes that belongs to the shortest path $\dot{\pi}(t_x, t_{x'})$ in the tree may create disconnected regions. It can be explained in the sense of topology that a shape in the tree of shapes is an open set (see [130]). Therefore, a sub tree in the tree of shapes may also be an open set. To deal with this problem, we rely on the subtraction the region $\Re^*(\dot{\pi}(t_x, t_{x'}))$, from $\delta(\Re(lca(t_x, t_{x'})))$, which is expressed as:

$$ROI(t_x, t_{x'}) = \delta(\Re(lca(t_x, t_{x'}))) - \Re^*(\dot{\pi}(t_x, t_{x'})) \tag{3.6}$$

where $\Re^*(\dot{\pi}(t_x, t_{x'}))$ which is the region under the shortest path in the tree of shapes can be defined as:

$$\Re^*(\dot{\pi}(t_x, t_{x'})) = \Re(lca(t_x, t_{x'})) - \bigcup_{t \in \dot{\pi}(t_x, t_{x'})} CC(t) \tag{3.7}$$

$\delta(A)$ is the closure operator of a shape A. The basic idea of this closure operator is to enclose the open set on the tree, thereby ensuring to generate the path that connect two pixels x and x'. Region $\Re(lca(t_x, t_{x'}))$ and $\Re^*(\dot{\pi}(t_x, t_{x'}))$ are respectively illustrated in the third figure of Fig. 3.3 as green and yellow regions.

This ROI is actually the set of all the possible paths between the two given points in the image space minimizing the Dahu pseudo-distance. In this region, we are able to obtain a coherent path between the two given pixels in the multivariate image. Therefore, this extended Dahu pseudo-distance solves the problem that we presented at the beginning of Section 3.1.2, in which the MBD is computed separately on each channel.

In the second step, we want to find a path between the two given pixels x and x', which belongs to the $ROI(t_x, t_{x'})$, so that it has the shortest length in the image space. This optimal path has different meanings. This path is not only the shortest path in the "color space" but also the shortest path in the image space. An example of the optimal path is depicted in Fig. 3.3 depicting a human retinal). The blue path in the figure is actually the optimal path between two given points x and x'. The shortest path is found in this region by using the heuristic A^* algorithm (see [21]). As we can see, the shortest path between two given points run along the blood vein. This property of the Dahu pseudo-distance can be applied to the path routing application. Several experiments of this optimal path will be illustrated in the next chapter.

3.2.2 Salient object detection based on the Dahu pseudo-distance

FIGURE 3.5: Boundary and connectivity priors [22].

To use the Dahu pseudo-distance in visual saliency detection, we adopt two priors about the background in natural images, namely *boundary* and *connectivity priors*, which are proposed in [22]. The first prior states the fact that most photographers restrict to cut off salient objects. In other words, the border of the domain of the image is mostly background. Concerning the second prior, the authors assume that the background regions are large and homogeneous, and background elements tend to connect with the border of the image. These two priors are illustrated in Fig. 3.5.

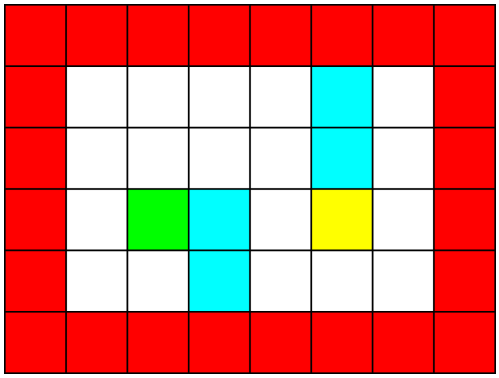


FIGURE 3.6: Finding the shortest path from every pixel in the image to the seed set.

Relying on these two priors, we consider pixels along the border of the domain of the image as seed nodes to compute the visual saliency map. This map is generated by finding the shortest path from every node on the MToS to the seed nodes and obtaining its Dahu pseudo-distance. An example is depicted in Fig. 3.6, where two target pixels are put inside the image, and all image boundary is considered to be the seed set. The optimal paths between the target pixels and the image boundary are shown in cyan color.

Inspired from the water flowing algorithm [8], which computes the MB-based saliency map directly on the image, we compute the Dahu saliency map based on the Dahu pseudo-distance by using the propagation approach. The process is flowed from the source node to the other nodes on the tree with different cost values, which are measured by the Dahu pseudo-distance. Each node on the tree may have three possible labels 0, 1 and 2, which correspond respectively to three states: droughty (the node is not touched yet), waiting (the node is pushed into the priority queue)

and flooded (the node is popped out of the priority queue). Initially, all the source nodes are flooded, and other nodes on the tree are droughty. The process is propagated from flooded nodes to their neighboring droughty nodes (child or parent), and then changes the state of the droughty nodes to flooded. This process continues until every node on the tree is flooded.

The computation of the Dahu saliency map from a set of seed pixels is illustrated in Algo. 8. Two auxiliary matrices L_u and U_u are used to represent the maximal and minimal values on the current path for each node. In the beginning, a corresponding set of nodes $T_{X'}$ on $MToS(u)$ of set of a seed pixels X' is found. Then the maximal and minimal value of each node $t_x \in T_{X'}$ are initially assigned to their color as follows:

$$\begin{cases} L_u^{(i)}(t_x) = \mu_u^{(i)}(t_x) \\ U_u^{(i)}(t_x) = \mu_u^{(i)}(t_x) \\ Dahu^{(i)}(T_{X'}, t_x) = 0 \end{cases} \tag{3.8}$$

where μ_u is the color of each node on the tree, and i indicates the index of color channel.

Every node in a set $T_{X'}$ is sequentially pushed to the priority queue Q. In this algorithm, each node has three different states: 0, 1 and 2. Initially, every node is assigned to a state 0, except the set of seed nodes $T_{X'}$, which is assigned to a state 2. Then, a node t_j is popped out of the queue. Its parent and children nodes, called t_k, are analysed. The updated procedure between two nodes that have parental relationship, is implemented as Eq. (3.9) on all color components.

$$\begin{cases} L_u^{(i)}(t_k) = min(\mu_u^{(i)}(t_k), L_u^{(i)}(t_j)) \\ U_u^{(i)}(t_k) = max(\mu_u^{(i)}(t_k), U_u^{(i)}(t_j)) \\ D^{(i)}(t_k) = U_u^{(i)}(t_k) - L_u^{(i)}(t_k) \\ Dahu^{(i)}(T_{X'}, t_k) = min(Dahu^{(i)}(T_{X'}, t_k), D^{(i)}(t_k)) \end{cases} \tag{3.9}$$

If the node t_k has not touched yet ($state(t_k) = 0$), $Dahu(T_{X'}, t_k)$ is computed and the node t_k is pushed into the priority queue Q depending on its distance value. In the case of $state(t_k)$ = 1, if the distance Dahu pseudo-distance of node t_k is higher than the t_j one, we update the maximal and minimal values of node t_k as Eq. (3.9). Then this new value ($D(t_k)$) is kept unless it is lower than its old distance ($Dahu(T_{X'}, t_k)$). The node t_k is then pushed into the queue Q. The process is iterated until the queue is empty. The saliency map of each pixel $S_u(X', x)$ is simply reading the distance at its corresponding node $Dahu(T_{X'}, t_x)$. Note that, Algo. 7 and Algo. 8 can be used in both color and gray-scale image.

As presented in Eq. (3.4), the input of the process is a multivariate image when the output is a (scalar) distance. However, in Fig. 3.1(b), the output of the process can also be a multivariate image (one distance map per channel). In the experimental section, we will show some examples of what we call abusively "vectorial distance maps". Note that we do not use the vectorial distance map for an evaluation purpose but for visualization only. It is actually a multivariate image, which is computed from a multivariate input based on the vectorial Dahu pseudo-distance.

To avoid ambiguities, we will refer in the sequel to *viso* for *vectorial-input-scalar-output*, to *vivo* for *vectorial-input-vectorial-output*, and to *siso* for *scalar-input-scalar-output* Dahu pseudo-distances.

Algorithm 8: Computation of the Dahu saliency map from a set of seed pixels.

Data: $MToS(u)$ of the image u , set of seed pixels X'
Result: The Dahu saliency map $S_{\boldsymbol{u}}(x,\, X')$

Find corresponding set of nodes $T_{X'}$ on $MToS(u)$ of X' as defined in Eq. (2.40);
Initiate the Dahu pseudo-distance of each node $t_x \in T_{X'}$ as Eq. (3.8) ;
Assign $state(t_x \in T_{X'}) = 2$, $state(t_x \notin T_{X'}) = 0$;
Push all of $t_x \in T_{X'}$ into a priority queue Q;
while Q *is not empty* **do**
    A node t_j is popped ;
    if $state(t_j)$ ==2 **then**
        skip ;
    end
    $state(t_j)$ =2 ;
    for *all children and parent nodes* t_k *of* t_j **do**
        if $state(t_k) == 0$ **then**
            Update the Dahu pseudo-distance of node t_k as Eq. (3.9);
            Push t_k and $Dahu(T_{X'}, t_k)$ into Q ;
            $state(t_k) = 1$;
        else
            if $state(t_k) == 1$ *and* $Dahu(T_{X'}, t_k) > Dahu(T_{X'}, t_j)$ **then**
                Update the Dahu pseudo-distance of node t_k ($D(T_{X'}, t_k)$) as Eq. (3.9);
                if $Dahu(T_{X'}, t_k) > D(T_{X'}, t_k)$ **then**
                    $Dahu(T_{X'}, t_k) = D(T_{X'}, t_k)$;
                    Push t_k and $Dahu(T_{X'}, t_k)$ into Q ;
                end
            end
        end
    end
end
Get the Dahu saliency map $S_{\boldsymbol{u}}(x,\, X') = Dahu(T_{X'}, t_x)$;
return$(S_{\boldsymbol{u}}(x,\, X'))$

3.2.3 Interactive segmentation based on the Dahu pseudo-distance

In this section, we use the Dahu pseudo-distance for interactive segmentation. The robustness of this distance w.r.t pixel fluctuation is promising for segmentation application. We show here two interactive segmentation versions based on the Dahu pseudo-distance. The first one segments the object directly on the MToS, while the second one borrows the probability distributions of the foreground and background regions from the scribbles.

3.2.3.1 A simple version for interactive segmentation based on the Dahu pseudo-distance

Interactive segmentation can be considered to be the binary classification where two sets of points F and B representing the user's input information. Each pixel in the image is classified based on the distance between the pixel itself and the set of seeds

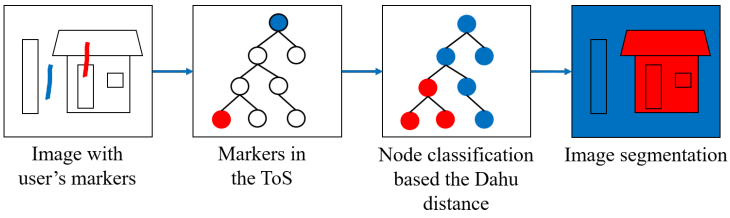


FIGURE 3.7: Interactive segmentation scheme on the MToS.

F and B. Differently to other methods which process directly in the image, our method segments the object region on the tree space. Our simple scheme for image segmentation using the Dahu pseudo-distance is illustrated in Fig. 3.7, which can be summarized as:

- We construct an MToS that represents the original color image.
- We label nodes in the MToS which correspond to the user's input marker as background/foreground classes. The markers are exploited as prior information about the background (B) and the foreground (F) in the image.
- Then the Dahu distance map of the non-labeled node S ($d_F(S)$ and $d_B(S)$) is computed using the marker F and B, respectively.
- The node S gets a label depending on the nearest class in the tree:
  $\arg\min_{C\in\{F,B\}} d_C(S)$.
- The segmentation image is reconstructed from the label of all of nodes in the tree.

This method is quite simple and easy to understand. However, there is still one challenge in interactive segmentation that we need to solve. The problem happens when the object region traverses several level-lines that belong to the background (the level-line goes from inside to outside of the object). As a result, a node may get a confused label from background and foreground. This is due to the fact that, in the simplification step to obtain the MToS from a GoS, two different color regions in the original image can be merged in the final tree. To address this issue, we present an extended method for interactive segmentation in the next section.

3.2.3.2 An extended version for interactive segmentation based on the Dahu pseudo-distance

In this section, an improving model for interactive segmentation is proposed using the Dahu pseudo-distance. Our method aims to solve the problem presented in the last section. We also consider an investigation of the prior information of the foreground and background regions from the scribbles. In the last version, the saliency maps is computed directly from scribbles in the original image. In this version, a statistical approach is applied to obtain more information about the image. The scheme of our method is presented in Fig. 3.8.

In the first step, two Gaussian mixture models (GMM) are used to perform the probability distribution of the foreground and background regions. A Gaussian mixture model is a probabilistic model that assumes all the data points are generated

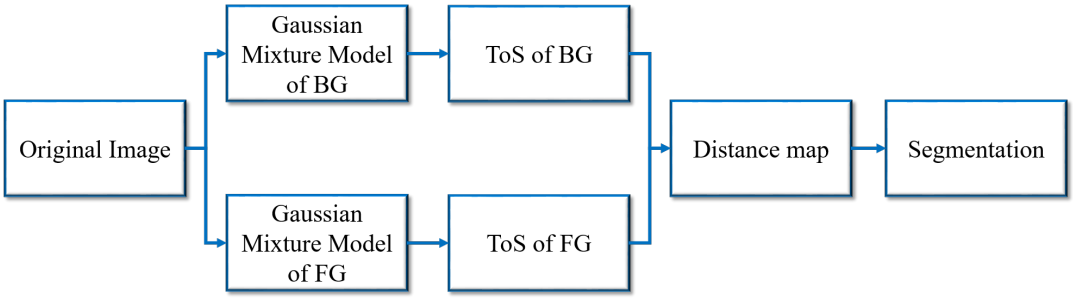


FIGURE 3.8: GMM model for interactive segmentation based on the Dahu pseudo-distance.

from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians. A Gaussian mixture model, which is composed of several Gaussian functions, is expressed as:

$$p(x) = \sum_{i=1}^{K} \Phi_i N(\mu_i, \sum_i) \tag{3.10}$$

where x is the pixel in the image, Φ_i defines how big or small the Gaussian function will be, a mean μ_i that defines its center, and a covariance $\sum_i$ that defines its width. The parameter K here denotes the number of clusters of our dataset. The mixing coefficient is the probability of each Gaussian function so that it satisfies this condition:

$$\sum_{i=1}^{K} \Phi_i = 1 \tag{3.11}$$

In our method, we use $K = 3$ to fit the two GMMs for foreground and background regions from the prior scribbles. After fitting these models, we estimate a probability of every pixel w.r.t the foreground scribbles in the image as follows:

$$P_F(x) = \frac{p(x|F)}{p(x|F) + p(x|B)} \tag{3.12}$$

A pixel with high probability indicates its color value that closes to the scribbles. We implement similarly for the probability of each pixel w.r.t the background scribbles $P_B(x)$.

In the next step, we construct two ToSs to represent these two probability maps. We label the node of the tree that corresponds with the markers. Then the Dahu pseudo-distance is used to compute the saliency map from the marked nodes. These two distance maps are compared to each other to determine the label of the pixel in the image. Then the image with the labels is reconstructed. The results of this method are presented in Chapter 5.

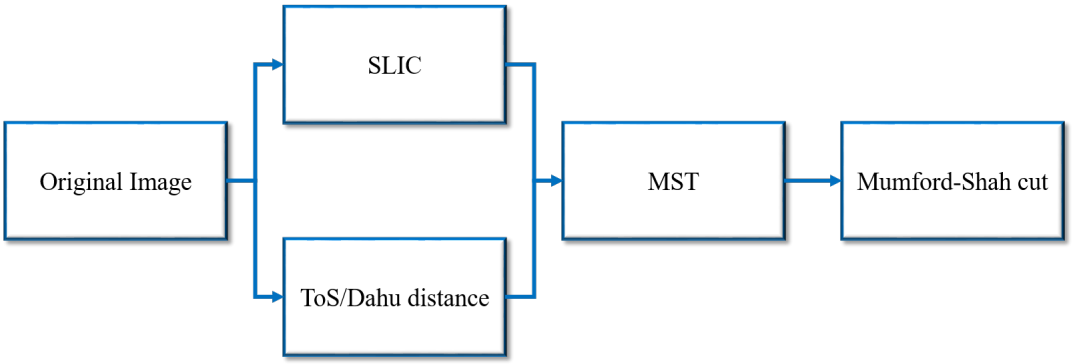


FIGURE 3.9: Image segmentation based on the Dahu pseudo-distance.

3.2.4 Image segmentation based on the Dahu pseudo-distance

Here we present another application of the Dahu pseudo-distance: image simplification and segmentation. Image segmentation has been widely used as an intermediary step for many tasks in image processing and computer vision, such as object detection and segmentation [34, 40, 52, 26].

The scheme for image segmentation of our method is illustrated in Fig. 3.9. Our proposed approach is a hierarchical segmentation method. It starts with the SLIC algorithm [25] to partition an image into several small regions called super-pixels. The regions contain more information about the object in the image than the pixels. Moreover, processing on image regions is faster than the original image because of significantly reducing the number of elements. The SLIC algorithm allows us to obtain a set of regions in several milliseconds. It is an advantage for many applications in image processing. The simplified image now can be seen as a graph of super-pixels. This first step drastically reduces the number of image elements to process in the next steps. Let $G = (V, E)$ denote an undirected graph consisted of vertices $v \in V$ and edges $e \in E$. Each edge $e_{ij} = (v_i, v_j)$ is assigned to a weight that measures the dissimilarity between the two vertices. This is the finest segmentation of the hierarchy.

A ToS of the image is constructed in parallel with the SLIC algorithm. The purpose of the ToS is to compute the Dahu pseudo-distance between the centers of each pair of neighboring superpixel, thereby analyzing the similarity between superpixels. Depending on the distance value, we can merge these superpixels to obtain a hierarchical segmentation.

The merging process is implemented by using the minimum spanning tree (MST) approach based on Kruskal's algorithm [55]. At the beginning of MST construction, we sort all the edges in non-decreasing order of their weights. The two super-pixels which have similar appearance tend to be connected in the MST. On the contrary, the edges with large weights are removed. We pick the smallest edge, which connects two nearest super-pixels, and join them together. The algorithm is repeated until one tree has remained. To quickly construct a hierarchical, we only used the low-level features for computing the similarity score. Let denote R_i and R_j two neighboring super-pixels, the distance $D(R_i, R_j)$ between R_i and R_j is used as an edge weight on the MST which is expressed as:

$$D(R_i, R_j) = \alpha \times d_u^{\text{DAHU}} + \beta \times d_c. \tag{3.13}$$

where d_u^{DAHU} is the Dahu distance between the center marker of two neighbor super-pixels R_i and R_j; d_c is a measure of the difference between the color histogram of two neighbors. Let C_i denote the center marker (3×3 pixels) of the super-pixels R_i. The Dahu distance between two center markers C_i and C_j of super-pixels R_i and R_j is computed as:

$$d_u^{\text{DAHU}}(C_i, C_j) = \min_{x_i \in C_i} \min_{x_j \in C_j} d_u^{\text{DAHU}}(x_i, x_j). \tag{3.14}$$

The Dahu distance between two center markers C_i and C_j is the minimum of the Dahu distance between all of pixels x_i and x_j inside the markers. This step is computed efficiently thanks to the advantage of the tree of shape.

Another distance, which is adopted in this scheme, is the dissimilarity of the color histogram between two super-pixels. For each region R_i, the histogram H_i is calculated from the quantized colors of all pixels in the region R_i, then normalized so that $\sum_{k=1}^{m} H_i(k) = 1$. The Chi-square distance between color histograms H_i and H_j is computed to express the color similarity between R_i and R_j.

$$d_c(R_i, R_j) = \exp\left(-\frac{1}{2}\sum_{k=1}^{m}\frac{[H_i(k) - H_j(k)]^2}{H_i(k) + H_j(k)}\right) \tag{3.15}$$

Each time we merge two regions R_i and R_j, the image segmentation $S^{(l)}$ arrives at a coarser level $S^{(l+1)}$, where (l) is the level of the image segmentation. Note that, K is the number of super-pixels, so that we have $K-1$ levels of image segmentation. After constructing a MST of super-pixels, our mission is producing a meaningful segmentation on this tree. There exist several works of hierarchical image segmentation based on energy minimization [48, 40]. Here, we adopt the Mumford-Shah functional proposed in [46] as an energy functional. A general energy functional has a form as : $E_{\lambda_s} = \lambda_s E_{re} + E_{fi}$, in which E_{re} is a regularization term and E_{fi} is a data fidelity term and λ_s is a parameter, which is able to control the simplification or segmentation degree of the algorithm. The higher value of λ_s is, the coarser segmentation degree is. Here, we aim to find an optimal image segmentation of a color image u. The data fidelity term E_{fi} is computed from the scalar luminance value with $l = (r+g+b)/3$. It is actually the variance of the luminance of each node on the MST. The regularization term E_{re} is equal to contour length $|\partial\{R\}|$ of each node R. The total energy of node R is expressed as:

$$E(R) = \sum_{x \in R} \left\| l(x) - \bar{l}(R) \right\|^2 + \lambda_s(|\partial R|) \tag{3.16}$$

With a fixed value of λ_s, the optimal cut is chosen from additive laws of composition. The energy on the parent node R is compared with the sum of the energy of all the children nodes T_i^R. The parent node is kept if it satisfies this condition : $E(R) \leq \sum E(T_i^R)$. After producing a cut to every node in the image, we reconstruct the simplified image from the remaining nodes in the tree, where each node corresponds with a region in image. We assign a median color from every pixels in the region to perform a unique color for this region. Several segmenting results of our method are illustrated in Section 5.4. In fact, the scheme, that we present in this section, is proved to be useful for a particular object detection: document detection, which is introduced in Section 3.2.5.2.

3.2.5 Document detection based on the Dahu pseudo-distance

Smartphones are now widely used to digitize paper documents. Document detection is the first important step of the digitization process. Whereas many methods extract lines from contours as candidates for the document boundary, we present in this section a region-based approach. A key feature of our method is that it relies on visual saliency (VS), which pertain to computer vision, have not been considered yet for this particular task. Here we compare different VS methods, and we propose a new VS scheme, based on the Dahu pseudo-distance. We show that our resulting saliency maps are competitive with state-of-the-art visual saliency methods, and that such approaches are very promising for use in document detection and segmentation, even without taking into account any prior knowledge about document contents.

In the first section, we present a simple method to detect the document, which is very fast and efficient. However, this method has a limitation on choosing the best threshold in the segmentation step. To overcome this problem, we propose an extended version which is based on the hierarchical image segmentation. This method is able to accurately segment the document region with a fast runtime speed.

3.2.5.1 A simple version for document detection based on the Dahu pseudo-distance

We present here a simple method for document detection. The scheme of our proposed method is illustrated in Fig. 3.10.

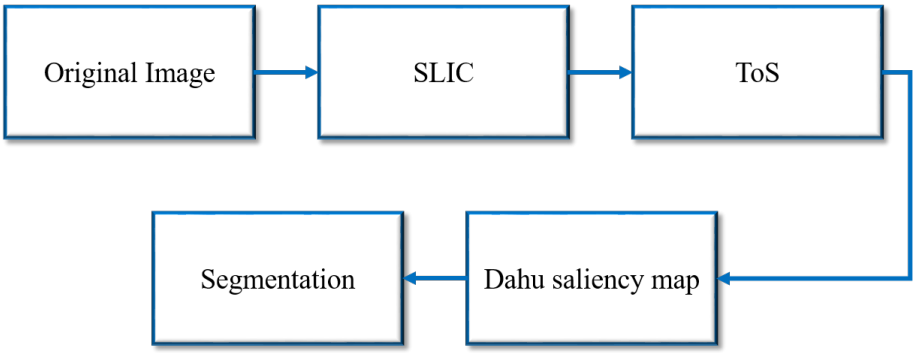


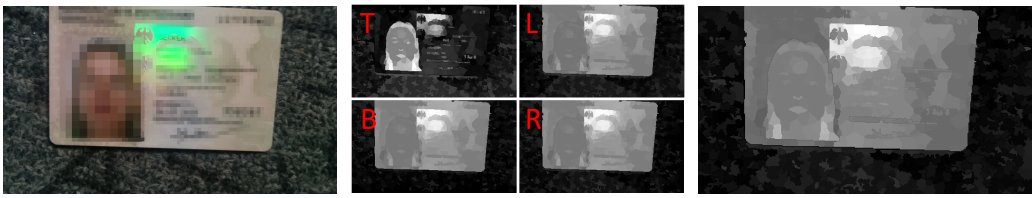
FIGURE 3.10: Simple scheme for document detection.

The method we propose is composed of four steps. In the first step, we rely on the SLIC algorithm [25] to simplify the image into superpixels (clusters of pixels, *i.e.*, very tiny regions). This step is interesting because it removes unnecessary image details, and the image can now be seen as a *graph of superpixels*, which has a reasonable size (instead of a huge matrix of pixels). That drastically reduces the number of elements to deal with the next steps. Then in the next step, we assign to each superpixel its average color, and a tree of shapes is computed from this graph. This tree of shapes is a good way to represent an image. From the tree of shapes, we then produce a saliency map from this structure using the Dahu pseudo-distance, and we normalize this map (defined later) in the third step. Finally, we apply a simple detection step by choosing a threshold value to obtain the resulting detection.

We assume that the four sides of the image boundary are mostly composed of the scene background (*i.e.*, the document does not predominantly touch the image boundary). Hence, from each boundary side of the image, we compute a saliency

map; for instance, with X_{top} being the set of pixels of the image top row, we have the saliency map $S_u^{\text{DAHU}}(x,\, X_{\text{top}})$. We end up with 4 saliency maps, depicted in Fig. 3.11(b), that we combine in a pixel-wise way using:

$$S_u^{\text{DAHU}}(x) \;=\; \sum_{i\in\{\text{top, left, right, bottom}\}} S_u^{\text{DAHU}}(x,\, X_i) \;/\; 4. \tag{3.17}$$



(a) Input (b) 4 saliency maps (c) Fusion

FIGURE 3.11: Effect of fusing four side-specific maps using Eq. (3.17).

An example is given in Eq. (3.17). As we can see in Fig. 3.11(a), the fact that the document touches the top row gives an irrelevant saliency map $S_u^{\text{DAHU}}(x,\, X_{\text{top}})$, marked T in Fig. 3.11(b). However, after the fusion of the 4 maps, we obtain a satisfying result, which is depicted in Fig. 3.11(c).

Similarly to some previous works [239, 6, 7], we normalize the saliency map by using "a - b" normalization (with $a = 0.1$ and $b = 0.8$), followed by an adaptive contrast enhancement with a sigmoid mapping. The saliency map in Fig. 3.11(c) is depicted *after* normalization in the 2nd row of Fig. 5.12(f).

The final detection step is deducing a binary image from the saliency map obtained by Eq. (3.17). In this section, our detection step is still experimental (briefly put, we only search for a threshold so that the result looks like a quadrilateral), since we focus on comparing gray-level saliency maps w.r.t. all possible thresholds.

3.2.5.2 An extended version for document detection based on the Dahu pseudo-distance

In the previous section, we present a simple method for document detection. The Dahu distance is used to compute the saliency map, then a simple threshold is applied to segment the document. However, a thresholding method is not strong enough to extract the whole document, and also the method has difficulty in choosing the best threshold for all images in the dataset. To deal with this problem, a scheme of our proposed method is given in Fig. 3.12.

Our extended approach is a saliency-based method which is composed of four main steps. In the beginning, we compute the Dahu saliency map similarly with the previous section by using Eq. (3.17). This method allows us to know the document position in the image. To successfully segment the document region, we propose to use a hierarchical image segmentation in parallel with computing saliency map. Hierarchical image segmentation partitions an image into several meaningful regions, hence reduces the number of image elements or reduce the search space in other words. The hierarchical image segmentation is implemented by using the Dahu pseudo-distance, which is introduced in Section 3.2.4. Then, we combine the result of the Dahu saliency map and hierarchical image segmentation to achieve a final saliency map. In this final saliency map, the pixels of the document region are brighter than other pixels. Otherwise speaking, the document region is highlighted

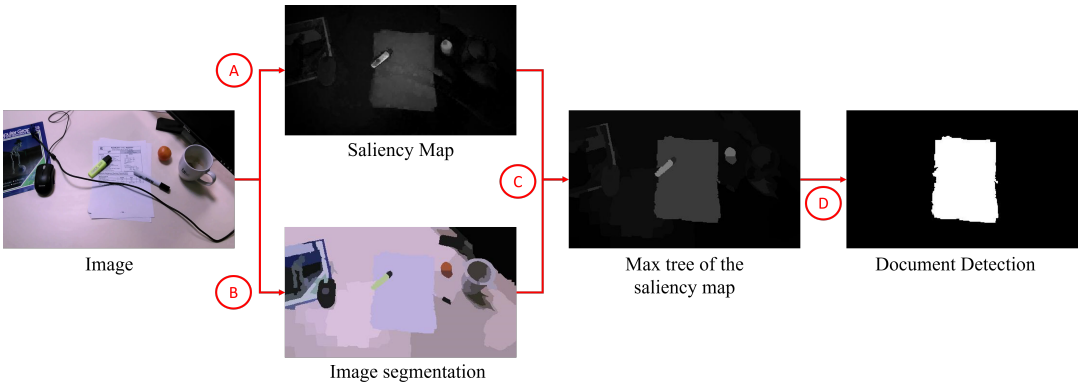


FIGURE 3.12: An extended scheme for document detection based on the Dahu pseudo-distance.

out of the image. Therefore, we construct a max-tree (which is presented in Section 2.3.1) of this saliency map. The document features are computed at each node of the tree. By computing several document features from each node in the tree, we are able to correctly segment the document region. The idea is to consider the local maxima of the energy map as candidates for document detection. To enhance the document detection during a video stream, a simple tracking method compares the positions of the shapes in consecutive frames.

Saliency based on the Dahu distance

We assume that we have a high contrast between the document and the background, and the border of the image is mostly background. Thus, we consider pixels along the border of the image as seed nodes to compute the visual saliency map $S_u^{\text{DAHU}}(x, X')$ similarly to Section 3.2.5.1. The saliency map $S_u^{\text{DAHU}}(x, X')$ is computed by using the propagation approach that we present in Section 3.2.2. In fact, this procedure is computed instantly on the ToS $\mathfrak{S}(u)$, whatever the set X'. Note that the ToS can be computed in quasi-linear time w.r.t. the number of pixels [18, 240] in the image, and can be parallelized [125].

Image simplification and segmentation

Different from the method developed at LRDE [26], which looks for a document among hundreds of thousands of nodes in the ToS of the original image, we use an image simplification and segmentation method based on the Dahu distance to reduce the number of image elements into tens of nodes for max-tree construction in the next step. This step is implemented in parallel with the computation of the saliency map. The details of the algorithm are presented in Section 3.2.4.

Max tree of a visual saliency map

Because the Dahu saliency map S_u^{DAHU} and the hierarchical image segment method are presented in previous sections, here we introduce the construction of the max-tree. In this step, we combine the Dahu saliency map with the result of the image segmentation method. The final saliency value of each region R_i is the average of the saliency map of every pixel in the region.

$$S_u^{\text{DAHU}}(R_i) = \frac{\sum_{x \in R_i} S_u^{\text{DAHU}}(x)}{|R_i|} \tag{3.18}$$

After combining the saliency map with the image segmentation, we get the final saliency map which is a set of connected components or a graph of regions. In the

final saliency map in Fig. 3.12, the document candidate is brighter than the background. We construct a max-tree representation directly on the graph of regions. After segmenting, the number of image elements is trivial, so that this max-tree is built immediately. This max-tree reduces significantly the search space of the document and provides an efficient way for document segmentation. The next step is to find the document candidate from all the nodes on the max-tree.

Document detection

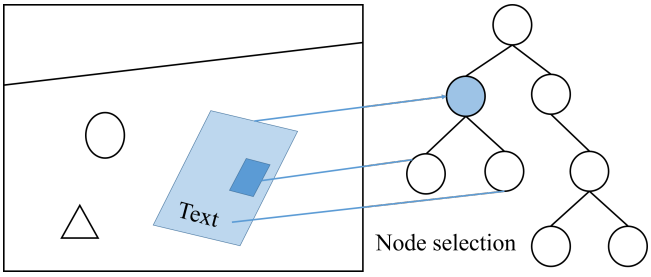


FIGURE 3.13: Document detection from the max-tree. A document candidate tends to have a quadrilateral shape, also the top line is parallel with the bottom line (respectively with the left line and the right line). On the other hand, the document region is brighter in the saliency map.

Assuming that the candidate document is represented in the max-tree, then the document segmentation problem resorts to finding the document in the tree space. To do that, we assign an attribute to each region that corresponds to a node on the max-tree. Here, we borrow one prior knowledge from the document information that is the document has a quadrilateral shape. We compute sequentially the attribute on every node of the tree and we observe how much these attributes fit with the document criteria. Our criteria are the followings:

1. A ratio that measures how much a shape boundary of a node A is close to the best fitting quadrilateral $Quad(A)$:

$$E_f(A) = \frac{|A \cap Quad(A)|}{|A \cup Quad(A)|} \tag{3.19}$$

2. The angles between the top (resp. the bottom) lines, denoted by TL (resp. BL), and between the left (resp. the right) lines, denoted by LL (resp. RL):

$$E_a(A) = \frac{\cos(TL, BL) + \cos(LL, RL)}{2} \tag{3.20}$$

3. The saliency map value of each node of the tree:

$$E_s(A) = S_u^{\text{DAHU}}(A) \tag{3.21}$$

The final attribute is computed by this equation:

$$E(A) = E_f(A) \times E_a(A) \times E_s(A) \tag{3.22}$$

Once the attribute $E(A)$ is available, we can look for the "most likely" node on the tree maximizing this attribute function. Fig. 3.13 shows the node selection procedure.

Tracking document between frames

To implement document detection in video streams, a tracking method is used to compare the document position between the previous frame and the current frame. Based on the node attributes computed in the previous section, we select the best three nodes in the tree as candidate documents, and then we look for this document position in the previous frame. The current detected shape A_t^* is the one that minimizes the distance to the shape A_{t-1}^* in the previous frame.

$$A_t^* = A_t^k : k = \min\{i | 1 \leq i \leq 3 : d(A_{t-1}^*, A_t^i)\} \tag{3.23}$$

where $d(X, Y)$ is the Jaccard index.

Chapter 4

Validation of the Dahu pseudo-distance

In this section, we validate the properties of the vectorial Dahu pseudo-distance via some experiments.

- Section 4.1 demonstrates the robustness of the vectorial Dahu pseudo-distance on salient object detection. We also compare the Dahu pseudo-distance with state-of-the-art MB-based distances.
- In Section 4.2, we analyze the efficiency of the Dahu pseudo-distance w.r.t noise and contrast between objects and background based on the ratio between inter- and intra-distances. We recall that the inter-distance is the distance from a marker outside the object to a marker inside the object. On the other hand, the intra-distance is the distance between two markers in the same object.
- In Section 4.3, we provide a comparison between the complexities (in time) of the Dahu pseudo-distance vs. some other MB-based distances.

4.1 Visual saliency detection

To show the robustness of the vectorial Dahu pseudo-distance, we start with visual saliency detection applications (see [8–10]). It should be reminded that visual saliency detection has been widely used in computer vision to obtain visual attention areas in the image. It is considered as a useful intermediary step for object detection and recognition.

First, we compare the vectorial Dahu pseudo-distance with the Dahu pseudo-distance on separate channels. Then in the following section, we compare the vectorial Dahu pseudo-distance with state-of-the-art MB-based distances.

As presented in Section 3.2.2, our visual saliency method is based on two priors about the background in natural images, namely *boundary* and *connectivity priors*, which are proposed in [22].

Datasets. To perform this evaluation, we use four large benchmark datasets:

- The first dataset is MSRA-10K dataset (see [209]), which contains 10000 images with pixel accurate salient object labeling for each image. It is widely used in salient object detection and segmentation community.
- The second DUTOMRON dataset (see [241]) consists of 5166 challenging images, each of which has one or more salient objects and complex background,

with pixel-wise ground truth annotated by five users. Therefore, images in the DUTOMRON dataset are more difficult and challenging for salient object detection.

- The third dataset ECSSD (see [242]) contains 1000 images along with pixel-wise ground truth masks, which includes more salient objects under complex scenes.
- The final PASCAL-S dataset (see [243]) contains 850 images and 1296 object instances from the validation set of PASCAL VOC 2010. Each image consists of multiple complex objects and clustered backgrounds, which is manually segmented for salient object annotation. The PASCAL-S dataset is designed to eliminate the center bias and color contrast bias.

Among these datasets, the PASCAL-S and DUTOMRON datasets are the most challenging.

Evaluation metrics. We use the following measures:

- The Precision-Recall (PR) curve, to evaluate the overall performance of a method concerning its trade-off between the precision and recall rates. For a saliency map, we generate a set of binary images by thresholding at values in the range of $[\![0, 255]\!]$ with a sample step as 1, and compute the precision and recall rates for each binary image. On a dataset, an average PR curve is computed by averaging the precision and recall rates for different images at each threshold value.
- The Mean absolute error (MAE), which is the average difference between a saliency map S (gray-level image) and a ground-truth image GT (binary image):

$$\text{MAE} := \frac{\sum_{x \in \mathcal{D}} |GT(x) - S(x)|}{Card(\mathcal{D})}, \tag{4.1}$$

  with $\mathcal{D}$ the domain of the initial image and $Card$ the cardinal operator.
- An F_β-measure defined by:

$$F_\beta := (1 + \beta^2) \times P \times R \,/\, (\beta^2 \times P + R), \tag{4.2}$$

  where P and R are respectively the precision and the recall which we mentioned above. We will set $\beta^2 = 0.3$ (because it is the classical setting in the visual saliency community).
- The percentage curve, which shows how many images in the dataset having a F_β score over a specific value. To compute it, we threshold the saliency map at each value being between 0 and 255 (we compute upper threshold sets), and we chose the "best" threshold set, that is, the one who gives the highest F_β score (we call this score ${F_\beta}^{max}$). After its computation for each image in the dataset, we compute the corresponding histogram (we chose a number of bins equal to 10), and we finally obtain the percentage curve.

- A score (briefly called EMD) inspired from [244] relying on the Earth Mover's Distance, which is the cross-bin distance function. It is used as a measure to estimate the dissimilarity between two signatures. In our case, the EMD is computed as the cost between the histogram of F_β score and the histogram of the ground truth image, which is equivalent to one bin at the value $F_\beta = 1$. Note that, the lower EMD values are, the better the method is.

4.1.1 Comparison of saliency maps obtained by the usual Dahu pseudo-distance on separate channels and by our vectorial Dahu pseudo-distance

(a) ECSSD

Method	MAE	$F_\beta{}^{max}$	EMD
Color	**0.21**	**0.69**	**0.29**
Gray	0.22	0.6	0.33
R	0.22	0.62	0.34
G	0.22	0.6	0.33
B	0.23	0.62	0.35
Combination	0.22	0.62	0.33

(b) DUTOMRON

Method	MAE	$F_\beta{}^{max}$	EMD
Color	**0.17**	**0.57**	**0.41**
Gray	0.18	0.50	0.43
R	0.18	0.52	0.45
G	0.18	0.50	0.43
B	0.19	0.52	0.45
Combination	0.18	0.52	0.43

(c) PASCAL

Method	MAE	$F_\beta{}^{max}$	EMD
Color	**0.22**	**0.69**	**0.28**
Gray	0.24	0.63	0.3
R	0.23	0.65	0.31
G	0.23	0.64	0.3
B	0.24	0.65	0.31
Combination	0.23	0.65	0.3

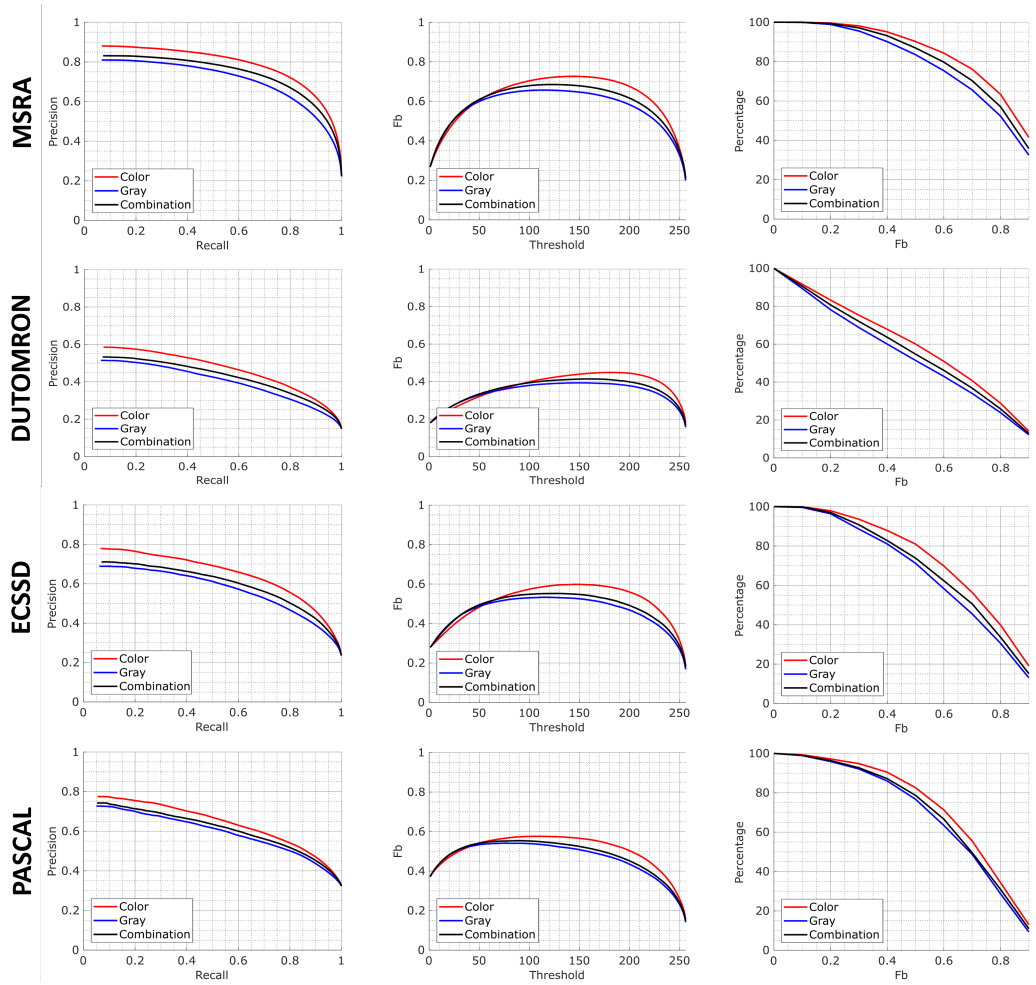
(d) MSRA

Method	MAE	$F_\beta{}^{max}$	EMD
Color	**0.16**	**0.79**	**0.17**
Gray	0.19	0.72	0.21
R	0.18	0.75	0.22
G	0.18	0.73	0.21
B	0.18	0.74	0.21
Combination	0.18	0.75	0.23

TABLE 4.1: Comparison between saliency maps obtained using the vectorial Dahu pseudo-distance and using the Dahu pseudo-distance on separate channels using $F_\beta{}^{max}$ measure and EMD score. "Color" is the *color* saliency map computed using our vectorial Dahu pseudo-distance applied directly on color image, "Gray" is the saliency map deduced from the Dahu pseudo-distance computed on the grayscale image, R, G and B are the saliency maps deduced from the Dahu pseudo-distance computed on each channel separately and "Combination" is the saliency map obtained by averaging the three saliency maps R, G and B. The best result is in bold form and the worst is in underlined. The three different measures show that our vectorial Dahu pseudo-distance leads to a much better saliency map.

Experimental setting. We compare our vectorial Dahu pseudo-distance (the extension of the Dahu pseudo-distance on the color images which are mentioned on Section 3.1.2) with the Dahu pseudo-distance computed on separate channels. To do so, we compare a ("*color*") saliency map computed using our vectorial Dahu pseudo-distance and saliency map computed by the Dahu pseudo-distance computed on separate channels (gray, red, green, blue). We also compare against a saliency map obtained by a simple combination of saliency maps computed on each three color channels (pixel-wise average of the three channels).

As mentioned previously, for our ("*color*") saliency map, relying on our vectorial Dahu pseudo-distance, we adopt the MToS and compute the saliency map following Eq. (3.1). For the saliency map computed on each separate color channel, the tree of shapes (ToS) is constructed to represent each image channel; then the saliency map of the Dahu pseudo-distance is computed as detailed in Eq. (2.41).

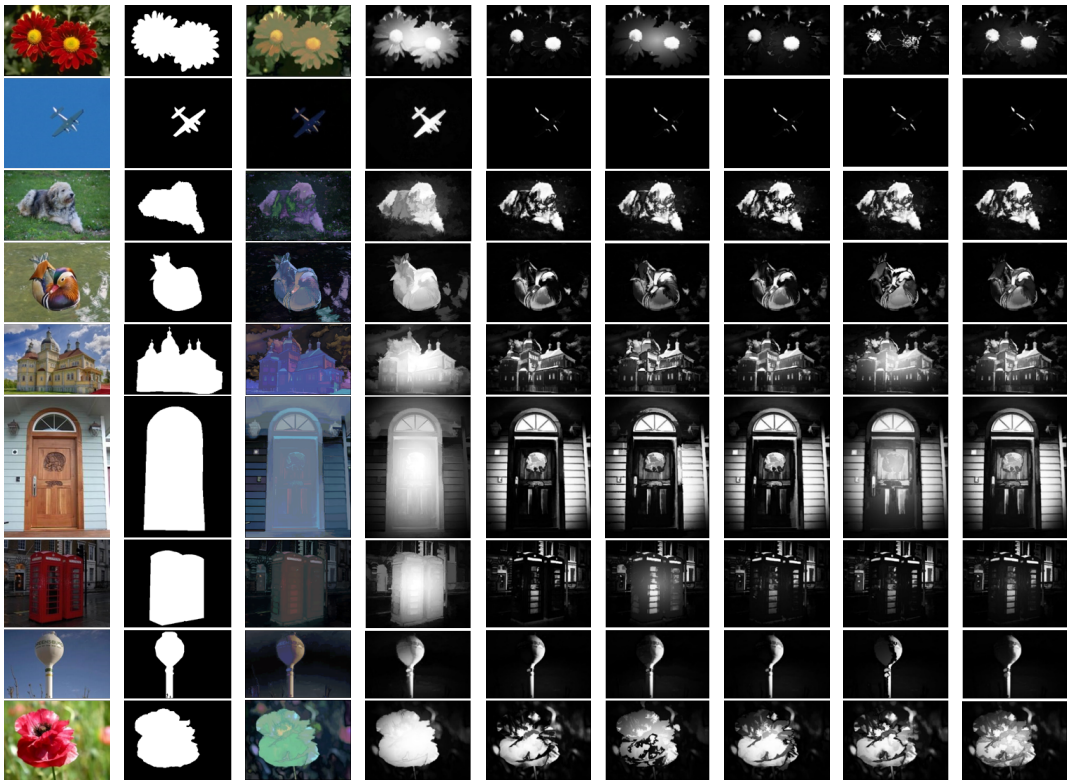


(a) PR (b) F_β-measure (c) Percentage

FIGURE 4.1: Comparison between saliency maps obtained using the vectorial Dahu pseudo-distance and using the Dahu pseudo-distance on separate channels. From top to down are four datasets: MSRA-10K, DUTOMRON, ECSSD, PASCAL-S. From left to right are three evaluation metrics: (a) Precision-recall curves, (b) F_β-measure, (c) Percentage curves. "Color" is the *color* saliency map computed using our vectorial Dahu pseudo-distance applied directly on color image, "Gray" is the saliency map obtained using the Dahu pseudo-distance computed on the grayscale image and "Combination" is the saliency map obtained by averaging saliency maps computed on separate red, green and blue channels. The three different measures show that our vectorial Dahu pseudo-distance leads to a much better saliency map.

In our implementation, input images are resized proportionally so that the maximum dimension (width or height) is 300 pixels. In the tree of shapes computation step, a border with the median value of all of the pixels on the boundary of the image is added to the image. We consider all the pixel in the added border of the image as seed pixels when we compute the saliency map. For the post-processing step, we used the same method as presented in [9] to "normalize" the resulting saliency maps.

Evaluation using PR curves. In a dataset, an average PR curve is computed by averaging the precision and recall rates for different images at each threshold value.

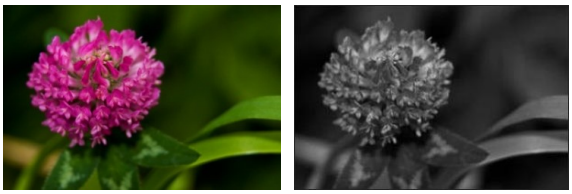


(a) Input (b) GT (c) *vivo* (d) *viso* (e) Gray (f) Red (g) Green (h) Blue (i) Combination

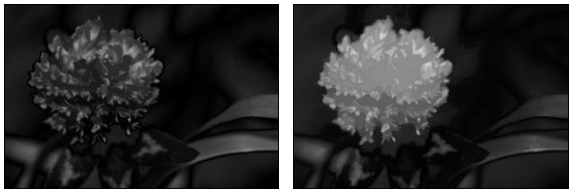
FIGURE 4.2: Several saliency maps of the vectorial Dahu pseudo-distance on color images and the Dahu pseudo-distance on separate channels. Note that image (c) and (d) are respectively the *vivo* and *viso* Dahu pseudo-distances on the color image. The Dahu pseudo-distance on the color image highlights the object over the background, whereas, when only one channel is used, the saliency map only spots a part of the object.

In Fig. 4.1, we show the PR curves for the saliency maps: directly computed on color images thanks to our vectorial Dahu pseudo-distance, computed on grayscale images thanks to the Dahu pseudo-distance and, a pixel-wise combination saliency map of the three saliency maps computed separately on the red, green, and blue channels (as presented in [9]). These saliency maps have been computed on four datasets: MRSA-10K, DUTOMRON, ECSSD, and PASCAL-S. The vectorial Dahu pseudo-distance outperforms the Dahu pseudo-distances on grayscale images and the combination of three channels in all datasets. On the most challenging dataset DUTOMRON, the performance of the distance maps deduced from Dahu pseudo-distance are lower than the performance of the one on others datasets. Note that in this dataset, there are multiple objects in images, the backgrounds are complex and the color contrasts between the foreground and the background are low.

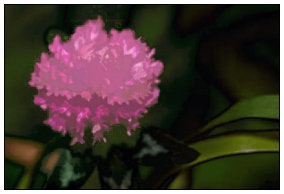
Evaluation using MAE. The MAE scores of compared methods are shown in Table 4.1. Note that the lower the MAE is, the better performance of the method is. The comparison of the saliency maps shows that the Dahu pseudo-distance does not give a better score on the grayscale images compared to the separate channels (R/G/B) however the combination of the saliency maps computed separately from each color channels improve the quality of the saliency map. This comparison shows also that the vectorial Dahu pseudo-distance achieves better scores than all other methods.



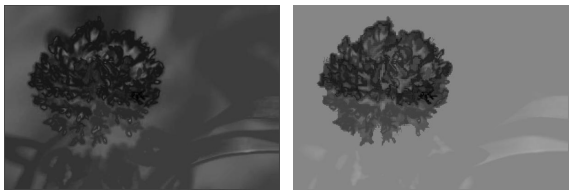
(a) Input (b) Gray



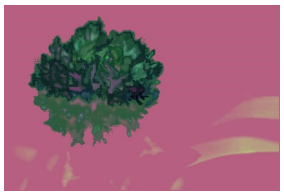
(c) *siso* border (d) *viso* border



(e) *vivo* border



(f) *siso* center (g) *viso* center



(h) *vivo* center

FIGURE 4.3: Different versions of saliency map deduced from Dahu pseudo-distances computed on the original color image (a) or the corresponding grayscale image (b); The saliency map when the seeds are the border pixels deduced from: the Dahu pseudo-distance (c) computed on the grayscale image and the vectorial Dahu pseudo-distance (d) computed directly on color image (with *vivo* (e) an additional color visualization of this latter); The saliency map when the seed is the center pixel deduced from: the Dahu pseudo-distance (f) computed on the grayscale image and the vectorial Dahu pseudo-distance (g) computed directly on color image (with *vivo* (h) an additional color visualization of this latter).

Evaluation using F_β-measure. We adopt the F_β-measure proposed in [245] to evaluate saliency maps. The F_β-measure scores are shown in Fig. 4.1. The vectorial Dahu pseudo-distance, (used to compute the "*color*" saliency map) significantly achieves

better scores than the Dahu pseudo-distance on grayscale images and than the combination across all datasets (whatever the threshold). We also notice that the F_β-measure curves of the Dahu pseudo-distance (whatever the strategy we use) have stable and flat curves, which is an advantage because the "best" threshold remains unknown and can vary a lot from an image to another. This stability makes the algorithm to search this threshold more robust. The "global" F_β-measure, maximized for all thresholds and all images, is denoted by ${F_\beta}^{max}$. In Table 4.1, ${F_\beta}^{max}$ of our vectorial Dahu pseudo-distance gives the best performance on all datasets.

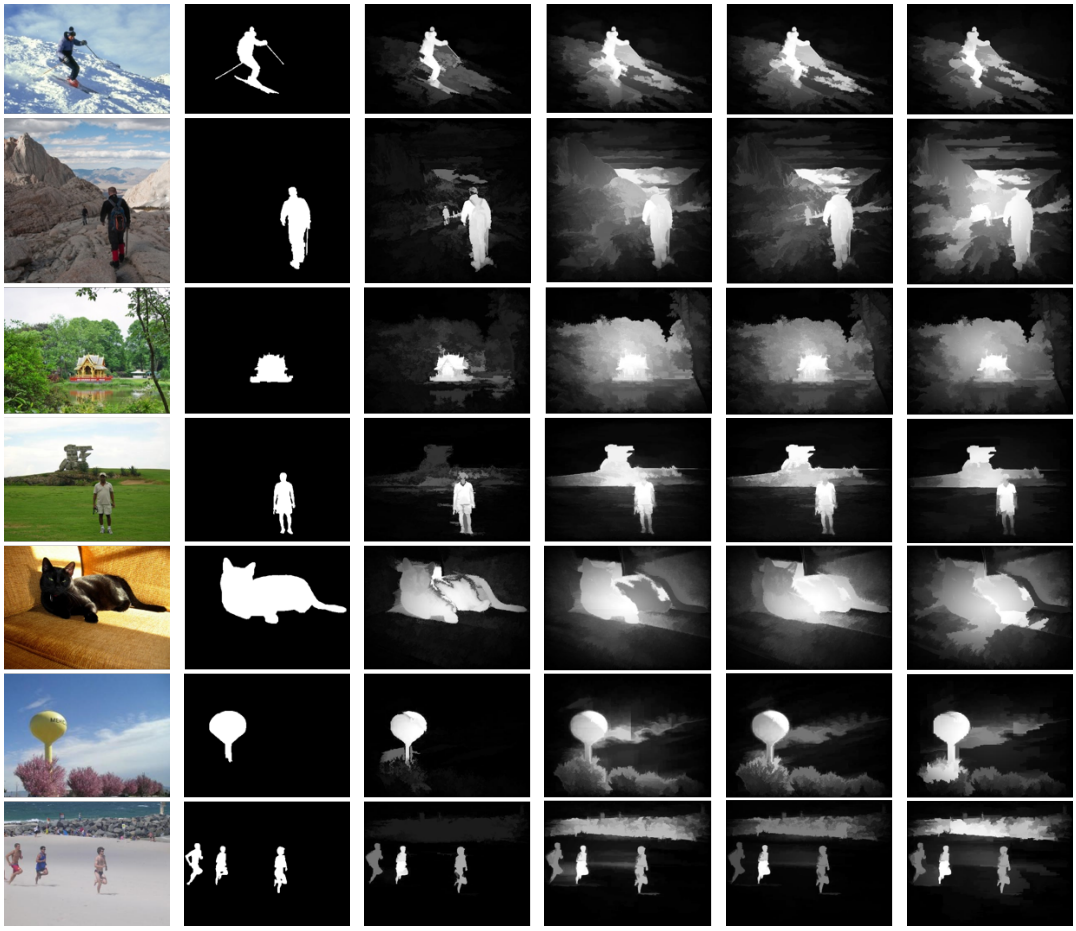
Evaluation using percentage curves and EMD. We do not only apply the value of ${F_\beta}^{max}$ to evaluate the performance of the Dahu pseudo-distance, but also use the percentage curves and the EMD. By computing the histogram of the best cut of F_β-measure, we are able to calculate the EMD and the percentage curve. In Fig. 4.1, the vectorial Dahu pseudo-distance not only gives better results of F_β-measure than the others, but also provides better saliency maps for the images of the dataset. Notably, Table 4.1 shows that in the MSRA-10K and ECSSD dataset, the number of good saliency map (F_β-measure > 0.8) of the vectorial Dahu pseudo-distance is higher around 7% than the Dahu pseudo-distance on separate channels. In the case of MSRA dataset, the vectorial Dahu pseudo-distance has more than 60% good saliency maps with the only assumption that the boundary is mostly background. In Table 4.1, the EMD results of the vectorial Dahu pseudo-distance is lower than the Dahu pseudo-distance on the separate channel which proves that our vectorial Dahu pseudo-distance improves saliency map computation. Furthermore the EMD score is quite low on MSRA, which means that the histogram of vectorial Dahu pseudo-distance is close to the histogram of the ground truth. It proves that the vectorial Dahu pseudo-distance is robust for saliency detection.

Some examples of saliency maps induced by the Dahu pseudo-distance are presented in Fig. 4.2. The saliency map induced by the vectorial Dahu pseudo-distance is shown in Fig. 4.2(d) (the "*viso*") and a color representation of the saliency map is given in Fig. 4.2(c) (the "*vivo*"). The "optimal" visual quality is reached for the vectorial Dahu pseudo-distance (compared to the Dahu pseudo-distances on separate channels or on the grayscale image). Indeed, the main barrier is clearly visible around the objects. The robustness of the vectorial Dahu pseudo-distance is easy to explain: the tree of shapes on the color image contains more information and is more structured than the tree of shape computed on separate channels.

In another example (see Fig. 4.3), we compare visually the vectorial Dahu pseudo-distance and the Dahu pseudo-distance by comparing deduced saliency map computed on a color image and computed on the corresponding grayscale image while using different sets of seed points. In the first case, all border pixels of the image are set to seed points, and in the second one, a single seed point is placed in the center of the image. The impact of seed positions and color usage is illustrated. When the seeds are the border pixels, the vectorial Dahu pseudo-distance on color image performs better than the Dahu pseudo-distance on the grayscale image. With the vectorial Dahu pseudo-distance, the flower in the image is spotted in the image. Whereas, the Dahu pseudo-distance computed on the grayscale image does not well distinguish the background and the flower (the contrast is low). Similarly, when the only seed is the center point, the vectorial Dahu pseudo-distance on the color image is better than the Dahu pseudo-distance on the grayscale image. The flower zone in the image is darker than the background. Besides, the background zone in the case of the vectorial Dahu pseudo-distance is more homogeneous than the Dahu pseudo-distance on the grayscale image. Similar intensities are obtained on most of

the background regions in the distance map. Normally, the more homogeneous of the distance map in the background is, the less seed points that we need to segment the image. This is an advantage of the vectorial Dahu pseudo-distance to reduce the number of seed points for object segmentation.

4.1.2 Comparison of saliency maps of the vectorial Dahu pseudo-distance with state-of-the-art methods



(a) Input (b) GT (c) *viso* Dahu (d) Fast-MBD (e) Waterflow-MBD (f) MST-MBD

FIGURE 4.4: Comparison on color images of saliency maps deduced from our vectorial Dahu pseudo-distance on color images with saliency maps deduced from state-of-the-art methods.

Experimental setting: In this section, the saliency map computed by the vectorial Dahu pseudo-distance is compared with some saliency maps deduced from multiple MB-based methods: Fast-MBD (see [9]), MST-MBD (see [10]), Waterflow-MBD (see [8]). To compare these methods, we modify them, as [8] do, by adding color and computing a color MBD by summing MBD on each channel. For the MST-MBD method, we construct a MST from the color image, then we compute the MBD in the same way as we did for MST-MBD on the grayscale image. In order to fairly evaluate the performance of these methods, we add an outer border to the image and consider all pixels on the boundary image as the background. Note that, in this experiment, we just want to compare the Dahu pseudo-distance with the MB-based distance, we do not try to achieve the best results of the saliency maps. The same

post-processing to normalize the saliency map, as in the previous section, is applied here.

Evaluation using MAE: Our method gives better MAE scores than other MB-based methods across all datasets. It can be explained by the fact that the vectorial Dahu pseudo-distance tends to give distance values lower than other MB-based distances, especially in the background regions, which constitute the largest part of an image. However, even if the vectorial Dahu pseudo-distance provides systematically better results, the difference is very low.

(a) ECSSD

Method	MAE	$F_\beta{}^{max}$	EMD
Dahu	**0.21**	0.73	0.228
Fast-MBD	0.22	**0.74**	0.21
MST-MBD	0.22	0.73	0.227
Waterflow	0.22	**0.74**	**0.205**

(b) DUTOMRON

Method	MAE	$F_\beta{}^{max}$	EMD
Dahu	**0.17**	**0.634**	**0.316**
Fast-MBD	0.21	0.626	0.324
MST-MBD	0.21	0.606	0.344
Waterflow	0.21	**0.634**	**0.316**

(c) PASCAL

Method	MAE	$F_\beta{}^{max}$	EMD
Dahu	**0.22**	0.72	0.23
Fast-MBD	0.24	**0.73**	**0.22**
MST-MBD	0.24	0.72	0.23
Waterflow	0.24	**0.73**	**0.22**

(d) MSRA

Method	MAE	$F_\beta{}^{max}$	EMD
Dahu	**0.17**	0.815	0.14
Fast-MBD	0.18	0.821	0.135
MST-MBD	0.18	0.812	0.143
Waterflow	0.18	**0.824**	**0.132**

TABLE 4.2: Numerical comparison of saliency maps deduced from the vectorial Dahu pseudo-distance applied on color images and different MB-based distances adapted to manage color images. The comparison is performed using $\overline{F_\beta}$ measure and EMD score. Best scores are in bold. Results of all methods are comparable and variations among them are negligible.

Evaluations using the F_β-measure: The F_β-measure is illustrated in Table 4.2. At a glance, the vectorial Dahu pseudo-distance shows equivalent results to the MST-MBD method and lower results than the Fast-MBD and Waterflow-MBD methods. However, the differences between these methods are minimal. In the DUTOMRON dataset, the Dahu pseudo-distance achieves better F_β-measure than other methods. Especially, in the MSRA dataset, the Dahu pseudo-distance and MB-based methods can achieve a high value of 0.82.

Evaluation Using EM distance. For the EMD, the Fast-MBD and the Waterflow-MBD methods achieve similar results in all datasets. Whereas, the Dahu pseudo-distance gives comparable results with the MST-MBD method, and slightly lower results than the Fast-MBD and the Waterflow-MBD methods but here again, the difference is rather low.

Some example images are given in Fig. 4.4. In these images, the backgrounds are not homogeneous like in the scene of the sky, the field of grass or even the sofa image. The Dahu pseudo-distance seems to deal better in these cases and achieves better performance than the MB-based distances. The tree of shapes properties and the insertion of the inter-pixels between the neighbor pixels allow the Dahu pseudo-distance to get the lower value compared to the MB-based distances. Also, each node on the tree of shapes is set at the median value of all the pixels in the node; this reduces the impact of noise in the color images. The Dahu pseudo-distance is shown to be robust to noise in the image. In the next section, we will explore this problem in details.

4.2 Efficiency and robustness of the algorithm

In this section, we investigate the efficiency of the algorithm and its robustness against noise. To do so, the ratio between inter-distances (the distance from a marker outside the object to a marker inside the object) and intra-distances (the distance between two markers in the same object). The higher this ratio is, this more discriminant the distance is. To analyze the noise stability of the vectorial Dahu pseudo-distance we will analyze the evolution of this ratio when noise in the image increases.

4.2.1 Ability to distinguish object and background

In this section, we analyze the ability to separate the object from the background. To do so, we measure the difference between the Dahu pseudo-distance and the MB-based distances by using the ratio between the inter-distance (the distance from a marker outside the object to a marker inside the object) and the intra-distance (the distance from two markers inside the object).

Experimental setting: This experiment is implemented on four benchmark datasets: ECSSD [242], PASCALS [243], DUTOMRON [241] and MRSA [209]. With two random markers in the image, each method finds the optimal path between them, which has the minimum bounding box on the color space. We compare the distance between two markers using the following "distances": the MST-MBD, the Waterflow-MBD and the vectorial Dahu pseudo-distance. We can not include Fast-MBD in this comparison because the Fast-MBD (see [9]) method works only when all the seed pixels are in the boundary of the image.

We create randomly 100 markers in the image and sequentially compute the distance between two markers. The Dahu pseudo-distance between two markers X and X' is computed this way:

$$d_u^{\text{DAHU}}(X, X') := \min_{x' \in X'} \min_{x \in X} d_u^{\text{DAHU}}(x, x'). \tag{4.3}$$

Evaluation metric: Using the binary ground truth, the inter- and intra-distances are well defined. The contrast metric is denoted by the ratio between the average of the inter-distances and the average of the intra-distances:

$$R = \frac{\frac{1}{N_1} \sum_{N_1} d_{\text{inter}}}{\frac{1}{N_2} \sum_{N_2} d_{\text{intra}}} \tag{4.4}$$

in which N_1 and N_2 are respectively the numbers of inter- and intra-distances.

TABLE 4.3: A comparison of ratio of inter- and intra-distances between the Dahu pseudo-distance and other MB-based methods.

Dataset	MST-MBD	Waterflow-MBD	Dahu
ECSSD	1.28	1.36	**1.404**
PASCALS	1.324	1.398	**1.448**
DUTOMRON	1.341	1.432	**1.483**
MRSA	1.784	**1.997**	**1.992**

Ratio of inter- and intra-distances evaluation: The ratio between the inter-distance and the intra-distance are presented in Table 4.3: the ratio of the Dahu pseudo-distance is higher than the one of the MB-based distances in all of the datasets. It means that the Dahu pseudo-distance is more contrasted than the MB-based distances. We can give an intuition of this result: the Dahu pseudo-distance is computed on the tree of shape, which considers an image to be a surface and a scalar value to be replaced by an interval. So, during the front propagation procedure, the pixel can pass through the inter-pixels to decrease the Dahu pseudo-distance between points on a same background as an illustration in Fig. 2.25(f). It leads to an increase of the ratio of the inter- and intra-distances of the Dahu pseudo-distance.

4.2.2 Robustness against noise

This section shows the impact of the noise on Dahu pseudo-distance and MB-based distance.

Experimental setting: An example image is chosen in Fig. 4.5 where two markers p_1 and p_2 (5×5 pixels) are set in the background and another marker p_3 is set inside the object. We observe the inter-distance $d(p_1, p_3)$ and intra-distance $d(p_1, p_2)$ during the test, with d a pseudo-distance among the Dahu or the MB-based one. During the noising experiment, a zero mean Gaussian noise is added to the image with the respective variance values: 0.0001, 0.001, 0.01, 0.1 and 0.5. One hundred noisy images are generated for each value of variance. The three markers are fixed for the entire experiment.

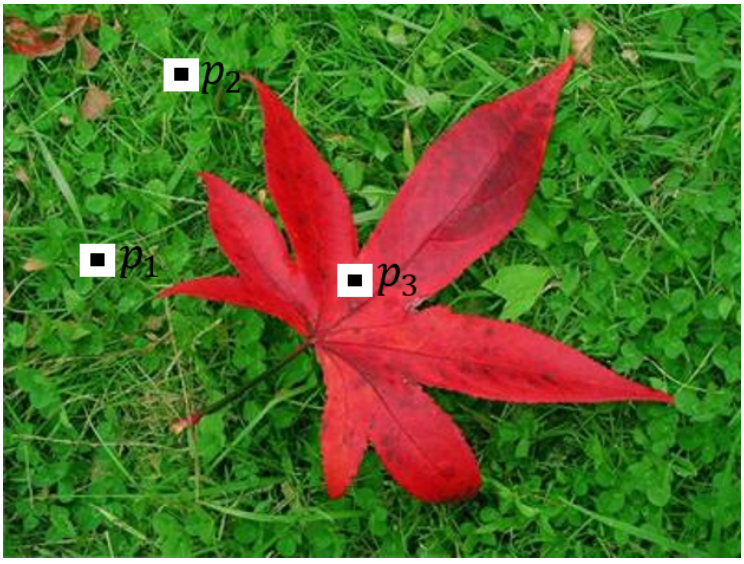
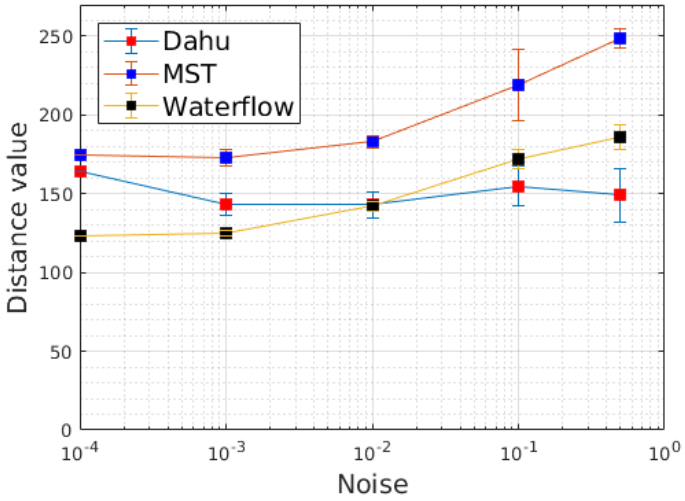
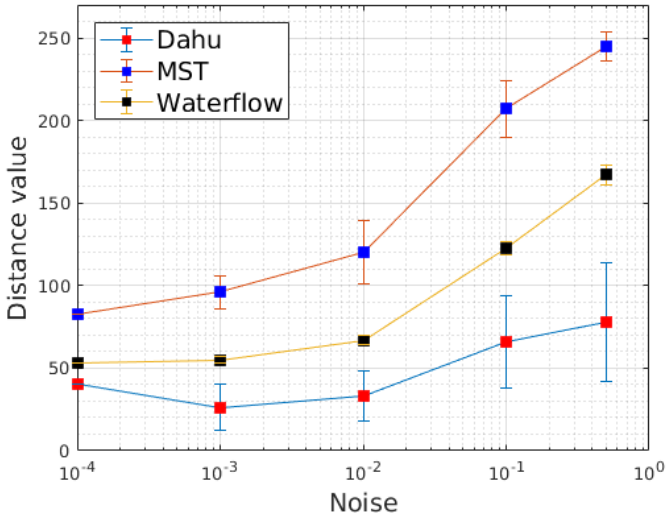


FIGURE 4.5: An example image to investigate noise stability of the Dahu pseudo-distance and MB-based distance. The points p_1 and p_2 belong to the background, when p_3 is inside the object (this picture comes from the MSRA dataset (see [209])).

Evaluation: The results of the experiments are presented, respectively, in Fig. 4.6 with the mean value as well as the associated confident interval. In both Fig. 4.6(a) and Fig. 4.6(b), we can see the evolution of the Dahu pseudo-distance and other MB-based distances. The MST-MBD and Waterflow-MBD both increase when the



(a) Inter-distance



(b) Intra-distance

FIGURE 4.6: Stability of the inter- and intra-distances using the vectorial Dahu pseudo-distance or other MB-based methods against Gaussian noise.

variance of the noise increases. Also, the Dahu pseudo-distance is much more robust against noise than the MB-based distances. Especially when the noise variance is high, the difference between inter- and intra-distances of MST-MBD and Waterflow-MBD is minimal. Whereas, the ratio of inter- and intra-distances of the Dahu pseudo-distance remains more stable. This experiment shows that the vectorial Dahu pseudo-distance is stable against noise variations. This property is important for many real-world applications.

4.3 Speed performance

In this section, we introduce empirical evaluation on the speed performance of our Dahu pseudo-distance against several MB-based distances.

Experimental setting: We want to measure the time necessary to compute numerous distances between two points using the Dahu pseudo-distance and other MB-based distances. We analyze the runtime of the algorithm to compute the distance between 100, 1000, 10000 and 100000 pairs of pixels on 20 tested images. Our method is implemented in C++. The evaluation is conducted on a machine with a 4-cores processor Intel i7 at 2.6GHz with 8GB of RAM (but we use always only one core). The size of test image is the same as used in previous experiment (the maximum dimension is 300 pixels) for all evaluated methods.

Speed performance evaluation: The construction of the tree of shapes is based on the Union-Find algorithm (see [18]). In [240], the authors propose to construct the tree of shapes based on a linear Max-tree algorithm on the depth map image, which is the depth of nodes that contain pixels in the image. The whole process is linear on average (and quasi-linear at worst). The computation of the ToS runs at about 20 FPS when used on grayscale images. Whereas, it takes about 1 second to construct the MToS of the color image. Although the computation of the MToS is longer than the ToS, but the Dahu distance maps deduced from MToS achieve better performances as we presented in Section 4.1.1. Depending on the application, we can choose either the ToS or MToS to compute the Dahu pseudo-distance. On the other hand, the contruction of the MST is fast (30 FPS) and easy to implement. However, this method is sensitive to the impact of noise and usually does not provide good results in this case.

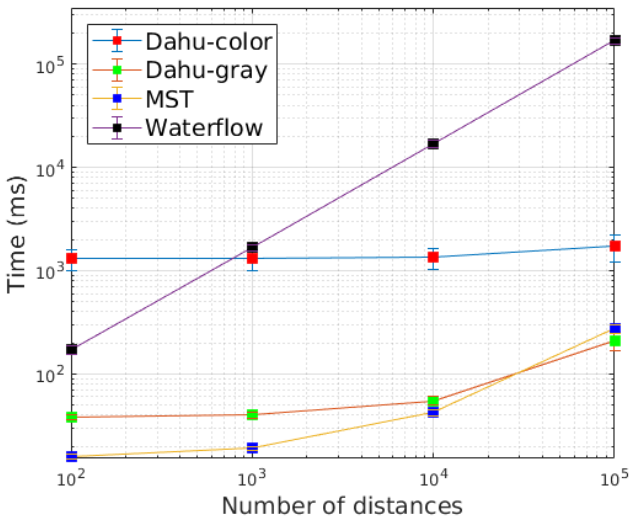


FIGURE 4.7: Execution time (in milliseconds) to compute numerous distances between two points using the (pseudo-)distances presented in this thesis.

There is another convenient point of the Dahu pseudo-distance (based on the tree of shapes): once the tree is computed, the Dahu pseudo-distance between any two points in the image is computed instantly. The execution time is illustrated in Fig. 4.7 with mean and confident values. As we can see in this figure, for a small number of distances, the Waterflow-MBD has an advantage compared to the vectorial Dahu pseudo-distance. However, when the number of distances increases, the Dahu pseudo-distance and the MST-MBD are much faster than the Waterflow-MBD. It can be explained by the fact that the Dahu pseudo-distance and the MST-MBD take a fixed time while to construct the tree, but when the tree is computed, the time to compute the distances in the tree is extremely fast thanks to the fast search of the

nodes corresponding to the points in this tree. This is (like the robustness against noise) a huge advantage for some real-time applications.

Chapter 5

Applications and Evaluations

The major application of the Dahu pseudo-distance is visual salient object detection, which is used in various other applications such as object detection and localization, object segmentation and tracking, or image refocusing. The visual salient object detection is carefully investigated in the previous section. Therefore, in this section, we demonstrate the ability of the Dahu pseudo-distance in other applications.

- The first application that we present in Section 5.1 is the shortest path finding between two points. It is actually an extension of the Dahu pseudo-distance that takes into account the spatial information in the image, which is introduced in Section 3.2.1.
- In Section 5.2, the Dahu pseudo-distance is applied in multimodal medical images with experiments to demonstrate the usability of the Dahu pseudo-distance on multimodal medical images, usage also made possible by the multivariate extension of the Dahu pseudo-distance. The next investigated application is to validate the ability of the Dahu pseudo-distance on multivariate images is multi-spectral imaging. We exploit the Dahu pseudo-distance to segment objects in the satellite images.
- In Section 5.3, we apply the Dahu pseudo-distance for interactive segmentation in both synthetic and natural images. We also propose an extended version for interactive segmentation using the Dahu pseudo-distance.
- Besides the interactive segmentation, we exploit the Dahu pseudo-distance for automatic image segmentation in Section 5.4. This approach can be used to provide the intermediary results for the next steps in object detection.
- In Section 5.5, a full framework based on the Dahu pseudo-distance is used to segment the document in the image. Our method is fast, easy to understand and able to achieve state-of-the-art results.

5.1 Shortest path in images

In this experiment, we apply our distance to the shortest path finding algorithm. The scheme of this algorithm is presented in Section 3.2.1, it is an extension of the Dahu pseudo-distance that takes into account the spatial information of the image.

Experimental setting: Let us assume we have two markers. We compare the resulting shortest paths found by the Dahu pseudo-distance on one side and by the other MB-based pseudo-distances on a second side. Some images, which are extracted

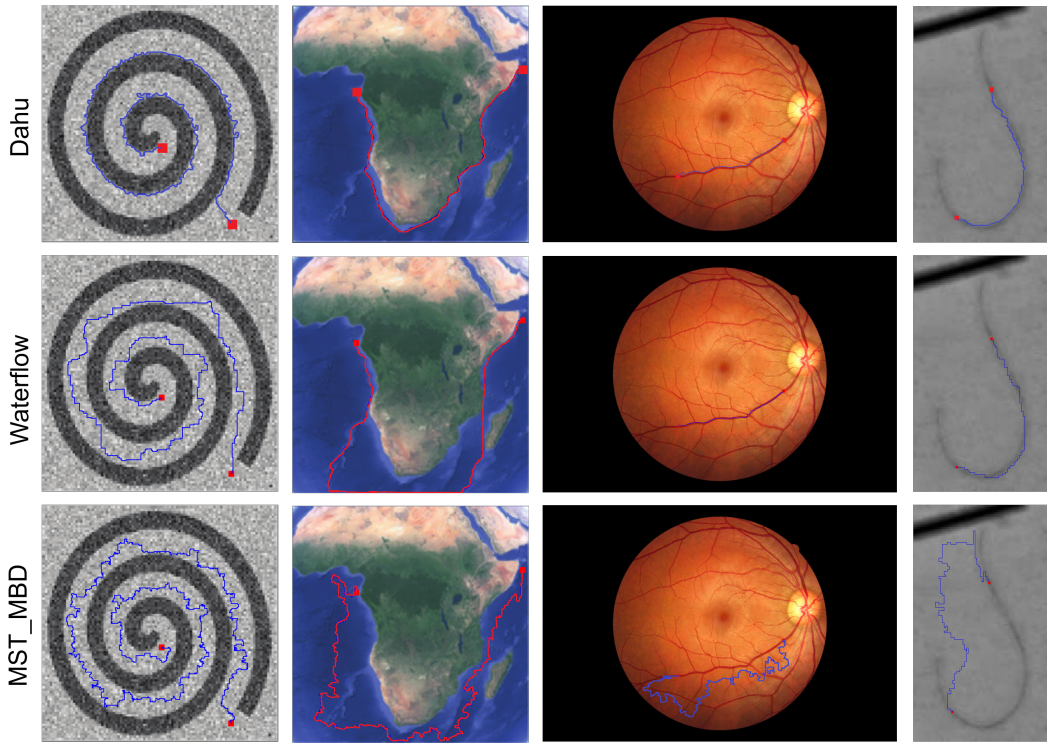


FIGURE 5.1: Shortest path finding in images. The input images and the end points (depicted in red) of the path we want to find are shown on each picture. Result are given for Dahu pseudo-distance, Waterflow-MBD and MST-MBD. Images are extracted from [246] and from [247].

from [246] and from [247] such as a noisy synthetic image, a map image, a retinal photography and a thin glass fiber are illustrated in Fig. 5.1.

Results: In the synthetic spiral image (see Fig. 5.1, column 1), there are two parts: the spiral and background. We can see at a glance that the shortest path provided by the Dahu pseudo-distance is "shorter" than the ones provided by the other MB-based pseudo-distances. The two chosen markers are in the background, and the shortest path between them based on the Dahu pseudo-distance, runs follow the shape of the spiral as we expected.

Similarly to the map image (Fig. 5.1, column 2), the goal is to find the shortest path connecting two points located on the sea near the coast. The shortest path based on the Dahu pseudo-distance is still better than the ones using other MB-based pseudo-distances.

The retinal image is depicted in Fig. 5.1, column 3. The two chosen markers are placed on a retinal blood vessel. In this image, the Dahu pseudo-distance and Waterflow-MBD give satisfying results when the MST-MBD is sensitive to noise and to blurring (which explains that its shortest path is deviated from the blood vessel).

Similarly, in the last example (see Fig. 5.1, column 4), the markers are placed on the glass fiber. The image is quite blurred, and the intensities of pixels along the fiber are variant, some parts of the fiber are darker than other parts. However, both the Waterflow-MBD and the Dahu pseudo-distance still find the shortest path that follows the fiber.

To conclude, the Dahu pseudo-distance gives better performance than the other MB-based pseudo-distances in this context.

5.2 Dahu pseudo-distance on multimodal images and hyperspectral images

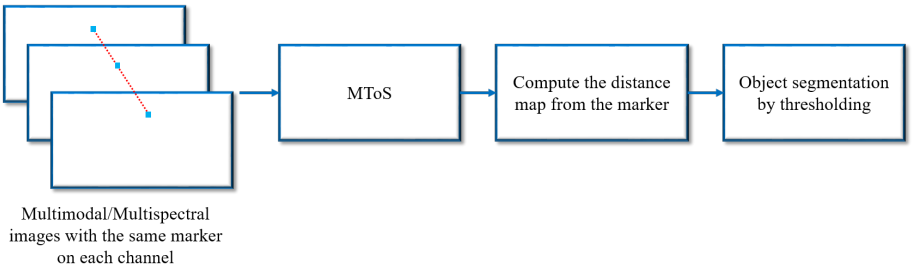


FIGURE 5.2: A scheme for object segmentation on multimodal/multispectral images.

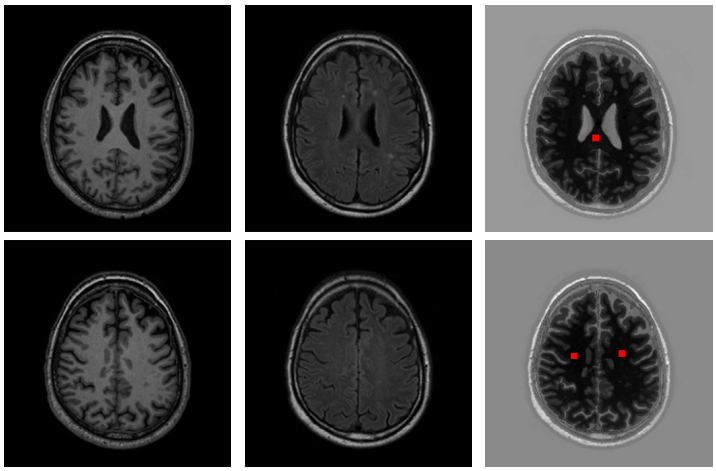
Multivariate imagery is widely used in various applications, ranging from medical imagery to satellite remote sensing. Multivariate can be color, multispectral, multimodal or multi-source imagery which corresponds to a set of one channel images. The data information from each image channel can be combined with other channels, so that we have more information about the image. A color image is just a special case of multivariate image. In this section, we present the application of the vectorial Dahu pseudo-distance in multimodal medical imagery and in multispectral satellite imagery. We use the same strategy to deal with the multimodal and multispectral images, which is illustrated in Fig. 5.2. The method begins with the construction of the MToS. Then we put markers in the image and compute a distance map from these markers based on the Dahu pseudo-distance. Finally, we use simple thresholding to segment the object in the image.

5.2.1 Multimodal images

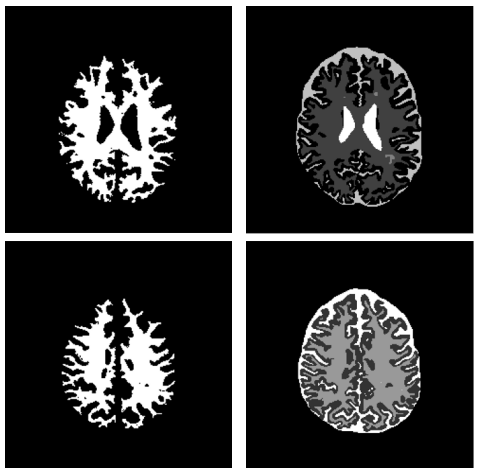
Multimodal imaging, defined as a combination of imaging modalities, which are acquired using different techniques, is becoming increasingly common in diagnosis and treatment planning (see [249]). Several common methods to achieve multimodal imaging are computed tomography (CT), magnetic resonance imaging (MRI), and positron emission tomography (PET). These methods demonstrate their abilities in a specific activity. Therefore, to overcome the limitation of each individual technique, multimodal imaging is proposed to provide a better solution. In this subsection, we applied the vectorial Dahu pseudo-distance to segment the white matter in 3D brain MR images.

We consider two images for each slice of the volume to the segment: the T1 slice (Fig. 5.3(a)), the T2-FLAIR slice (Fig. 5.3(b)). Examples of saliency maps are shown in Fig. 5.3, where images have been acquired from different modalities. The Dahu pseudo-distance is able to segment the white matter region from the combination of these two modalities.

We construct the MToS on these images to get the mutual information from different machines. Then a marker (5 x 5 pixels) is put on the white matter region. The



(a) T1 (b) T2-CLAIR (c) Distance map



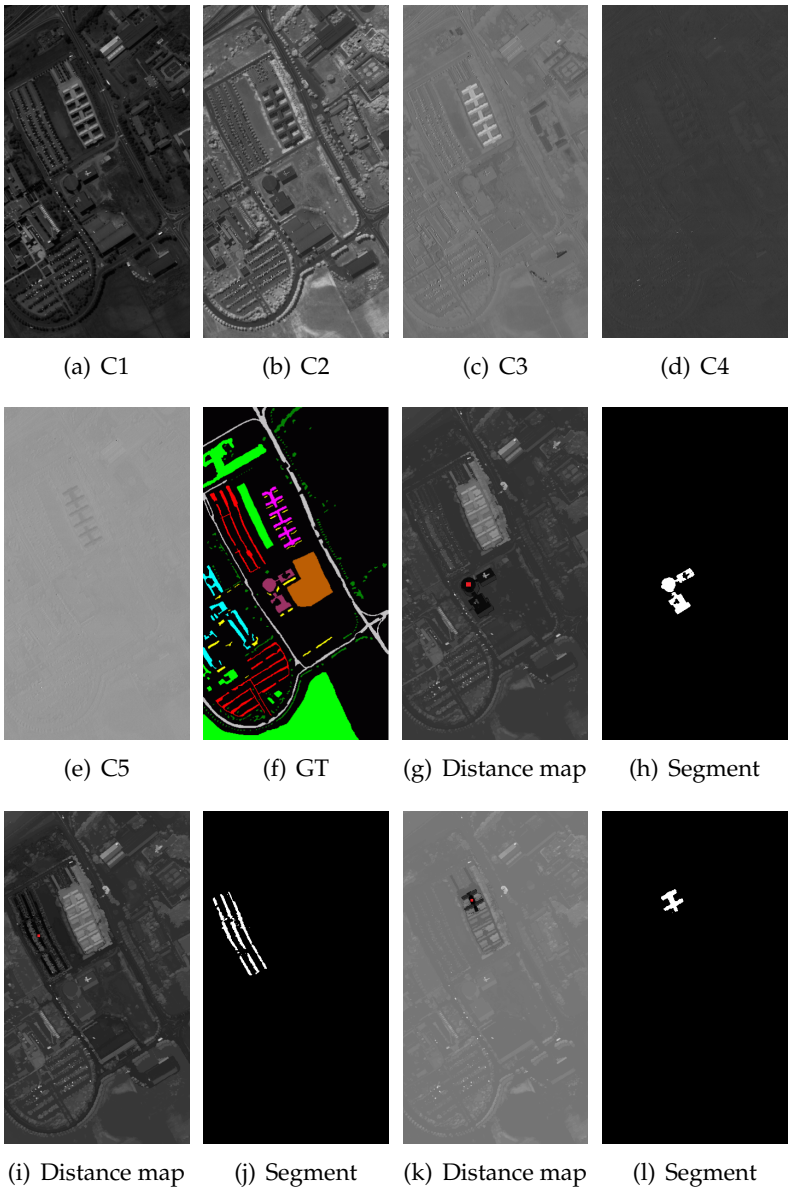
(d) Segmentation (e) GT

FIGURE 5.3: White matter segmentation using the vectorial Dahu pseudo-distance. Images are taken from [248].

Dahu pseudo-distance is used to compute a distance map from this seed. A simple threshold method is used to segment the white matter region in the image. As a first remark, the MToS preserves the geometric information of the two channels and mixes them in a sensible way. Second, as one can see, the distance map gives low values to the white matter region, which will be thresholded to get a final segmentation. As compared to the ground truth image, our method achieves a good segmentation result. The vectorial Dahu pseudo-distance show then to be efficient for this experiment.

5.2.2 Multispectral images

Over the last few years, the use of multispectral imaging has been increasingly investigated in many applications, especially in target detection and recognition (see [250]). Multispectral imaging collects information from hundreds of spectrum bands, thus providing a powerful mean to discriminate different objects in the image. Similarly to the application in the previous section, we focus here on the use of the vectorial Dahu pseudo-distance in multispectral imaging.



(a) C1 (b) C2 (c) C3 (d) C4

(e) C5 (f) GT (g) Distance map (h) Segment

(i) Distance map (j) Segment (k) Distance map (l) Segment

FIGURE 5.4: Hyperspectral images

We apply our method on the Pavia University dataset (see [251]). It consists of 103 images which correspond each to a spectral channel. The dataset has a size of 610*340 pixels, contains nine classes which represent trees, meadows, asphalt, etc. The images are pre-processed with a PCA (see [252]) to reduce number of channels of the images. This pre-processing relies on the fact that neighboring bands of multi-spectral are highly correlated and contain mutual information about the object. The PCA algorithm reduces the correlation among the bands, and selects the best bands for object detection.

In general, most of the information may be contained in the first few bands. In our case, we choose the first 5 channel components. As we can see in Fig. 5.4, some objects clearly appear in some images but not in the others. The MToS is then constructed on these images. We put some seeds in the image to compute the distance map. Then a simple threshold is used to segment the object in the image. As we can see in Fig. 5.4, our proposed method can segment the objects in the image with

high accuracy, for instance, the painted metal sheets, the bitumen, and self-blocking bricks classes.

These results demonstrate the robustness of the vectorial Dahu pseudo-distance in this context.

5.3 Dahu distance in interactive segmentation

In this section, we demonstrate the robustness of the Dahu pseudo-distance with regard to the marker positions for interactive object segmentation. The position of seeds plays a key role in the quality of image segmentation, especially if the segmentation is based on computing the distances from the seeds. We investigate how the Dahu pseudo-distance is stable and how this stability influences the results of distance-based image segmentation. The Dahu pseudo-distance is firstly applied on the synthetic and then natural images. We also proposed a new scheme based on the Dahu pseudo-distance for interactive segmentation.

5.3.1 Dahu pseudo-distance in interactive segmentation on synthetic images with taking into account noise and position of seeds

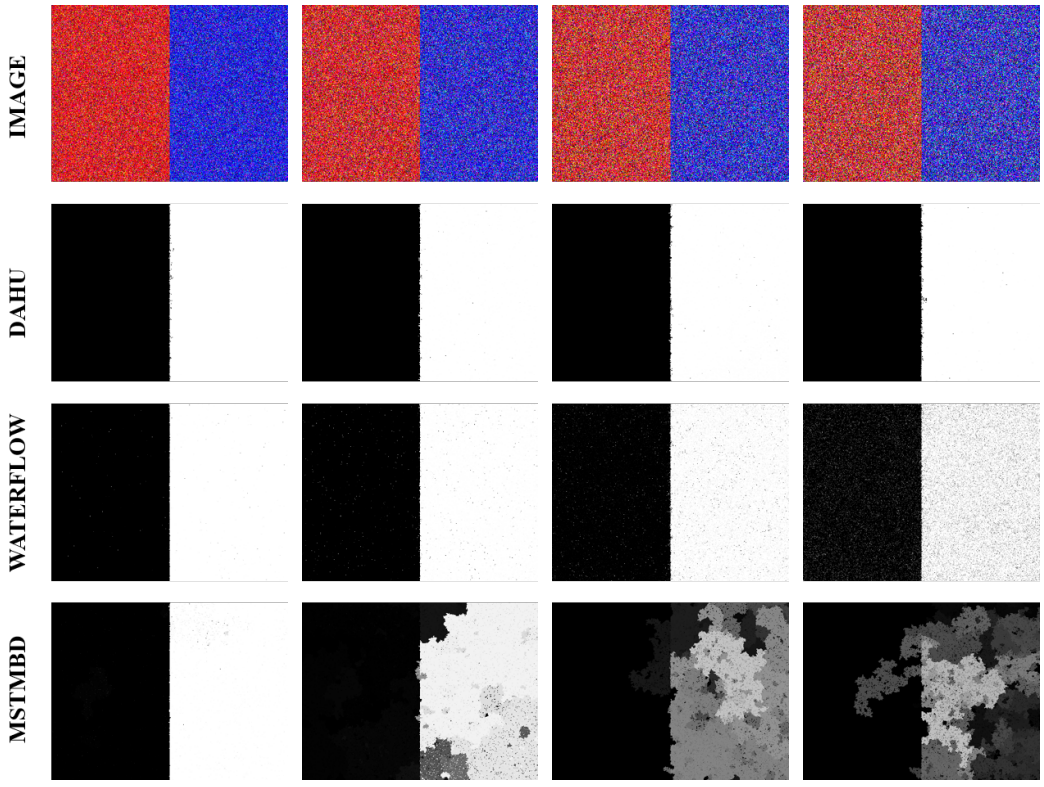
In this experiment, we demonstrate the ability of Dahu pseudo-distance in dealing with noise and position of seeds in interactive segmentation on synthetic images. To do that, we compare the Dahu pseudo-distance with several MB-based distances.

Experimental setting: The interactive segmentation is applied on the synthetic images, which have two separated segments by a boundary in the image center. The example images are illustrated in Fig. 5.5. Four examples are noisy images, which are added a zero- mean Gaussian noise with the variance values σ^2 are respectively 0.2, 0.3, 0.4, and 0.5. Two random markers which sequentially correspond to the two segmented areas (one marker in each segment). The marker is a square 5 x 5 pixels to reduce the local noise influence. We also take into account the impact of the position of markers on the segmentation results by creating 200 samples with various marker positions and compute the standard deviation of the segmentation error.

Evaluation metric: We analyze the result of interactive segmentation of our method using the Dahu pseudo-distance by comparing with method MST-MBD and Waterflow-MBD. To evaluate the segmentation results, a segmentation error is evaluated as a percentage of incorrectly labelled pixels. The results are presented in the form of mean values and standard deviations. For better visualization, we also accumulate the segmentation result of each method across 200 images. The higher contrasted between the two segments are, the better method is.

Evaluation: The segmenting images are illustrated in Fig. 5.5. As a glance, the Dahu pseudo-distance gives better segments than other MB-based methods. The image is well segmented and also the accumulated result is well contrasted. The Waterflow-MBD achieves quite good results, although the accumulated gray-scale image is not well contrasted as good as the result of the Dahu pseudo-distance. On the contrary, the MST-MBD is sensitive to noise. Therefore, it does not get satisfying results.

The segmentation errors table is shown in Table 5.1. In the table, we can see that the Dahu pseudo-distance gives extremely good results on these synthetic images, the error rates are quite low. Especially, the standard deviation result is really low. The small value of the standard deviation expresses the stability to the seed



(a) $\sigma^2 = 0.2$ (b) $\sigma^2 = 0.3$ (c) $\sigma^2 = 0.4$ (d) $\sigma^2 = 0.5$

FIGURE 5.5: Interactive segmentation on synthetic images with taking into account the seed point positions and noise.

TABLE 5.1: The segmentation results of the synthetic images (5.5). The percentage of incorrectly labelled pixels is presented in the form of the mean values and standard deviations. The best scores are in bold.

Method	$\sigma^2 = 0.2$	$\sigma^2 = 0.3$	$\sigma^2 = 0.4$	$\sigma^2 = 0.5$
Dahu	**0.077 (0.003)**	**0.1 (0.04)**	**0.13 (0.06)**	**0.15 (0.009)**
Waterflow	**0.068 (0.03)**	0.3 (0.3)	1.48 (1.57)	5.76 (5.15)
MSTMBD	0.17 (0.5)	9.9 (10.5)	35.27 (6.28)	44.2 (41.66)

positions of the Dahu distance. It proves that the Dahu pseudo-distance is robust to noise and also stable for the position of seed points. This property is very important for choosing a suitable method for interactive segmentation. If the seeds placed in different positions lead to a stable segmentation result, it simplifies user interaction. The Waterflow-MBD gives quite good results, but lower than the Dahu pseudo-distance. With the low value of the variance of the noise, the MST-MBD method gives satisfying results. However, when the variance increases, the quality of MST-MBD decreases dramatically. The Dahu pseudo-distance is strong for this kind of synthetic images. In the next section, we adopt the Dahu pseudo-distance for interactive segmentation natural images.

5.3.2 Dahu pseudo-distance in interactive segmentation concerning the numbers of markers

In this test, we examine the dependence of the number of seeds of the Dahu pseudo-distance for interactive segmentation in natural images. The less the number needed of seed pixels are, the more powerful method is. It helps users not to use many seed points.

Experimental setting: The testing images, which are extracted from the ECSSD dataset [242], are depicted in Fig. 5.6. We suppose that there is only one object in the background. For the object, only one marker is necessary. For the background, we respectively use one, two, three markers and evaluate the segmenting result of each case. The size of the marker is similarly with the previous test (5 x 5 pixels). We take into account how many needed markers to achieve a good segmentation. The method for interactive segmentation is similar to the one that is presented in the previous section.

FIGURE 5.6: On the sensitivity to the number and position of seeds.

Results: On Fig. 5.6, the segmenting results by using one marker for the object and one marker of the background is very good. The segmenting region is closed to the ground truth image. We also notice that the segmenting results by using one, two, or three markers are similar. By using very few seed points, the Dahu pseudo-distance on color image is able to achieve good segmentation. It proves the robustness of the Dahu pseudo-distance to the number of seed points.

5.3.3 A simple interactive segmentation based on the Dahu pseudo-distance on natural images

In this section, we investigate the ability of the Dahu pseudo-distance in interactive segmentation on natural images via a simple method which is presented in Section 3.2.3.1.

Experimental setting: We test the Dahu pseudo-distance in interactive segmentation using the Gulshan dataset [59]. In this dataset, 151 images along with the prior scribbles to define the background and foreground region, are provided. In the beginning, we assign labels F or B to the set of nodes that corresponds with a set of scribbles in the image. In some special cases, a node in the tree may get a common label from background and foreground scribbles. We assign this node to a majority class.

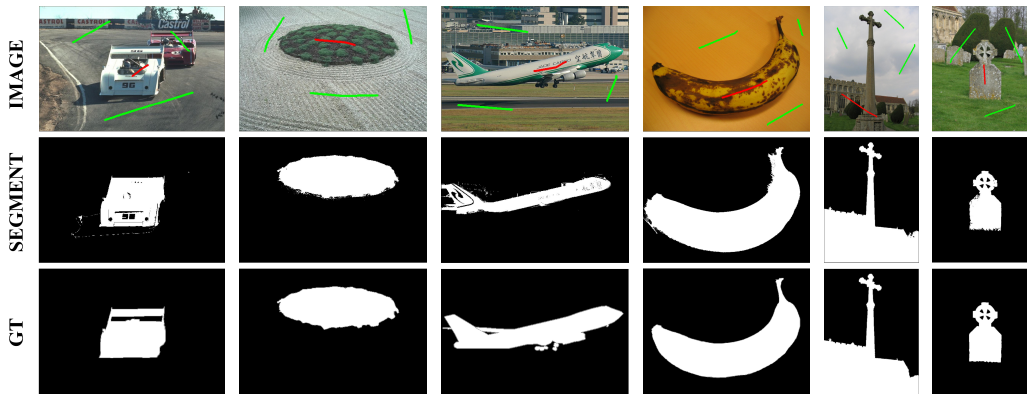
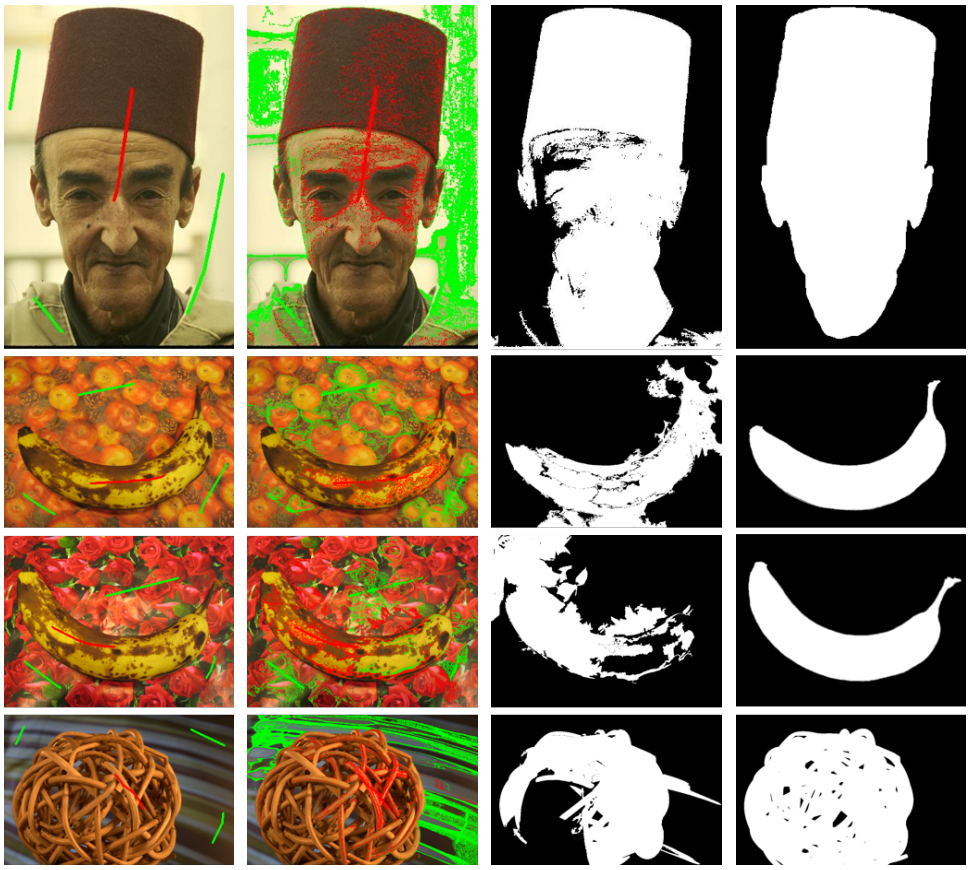


FIGURE 5.7: Interactive segmentation.

The segmenting procedure is implemented similarly with the method that we present in Section 3.2.3.1. We apply the Dahu pseudo-distance to compute two saliency maps from two sets of scribbles for segmenting an image into two parts: object and background. This operation is equivalent to classify every node in the tree into two classes. The obtained label of each node is the nearest label to the set of background and foreground classes.

Qualitative results: Several successful segmenting results are illustrated in Fig. 5.7. The first row shows the original images with their prior scribbles on object and background regions. The segmenting results of our Dahu pseudo-distance are depicted in the second row. Comparing with the ground truth in the last row, our method achieves quite good results. Despite the simplicity of our method, the segmenting results are very close to the ground truth. These results prove that our distance is strong for interactive segmentation.

Limitation: Fig. 5.8 illustrates some failure cases of our algorithm. As we can see, there exist background marker-nodes in the foreground region and vice versa since there are several level-lines that go from inside to outside of the object. In that situation, we can not correctly segment the background and foreground regions, as presented in Fig. 5.8(c). In the next section, we demonstrate an algorithm based on the Dahu pseudo-distance to ameliorate the segmenting results for interactive segmentation.



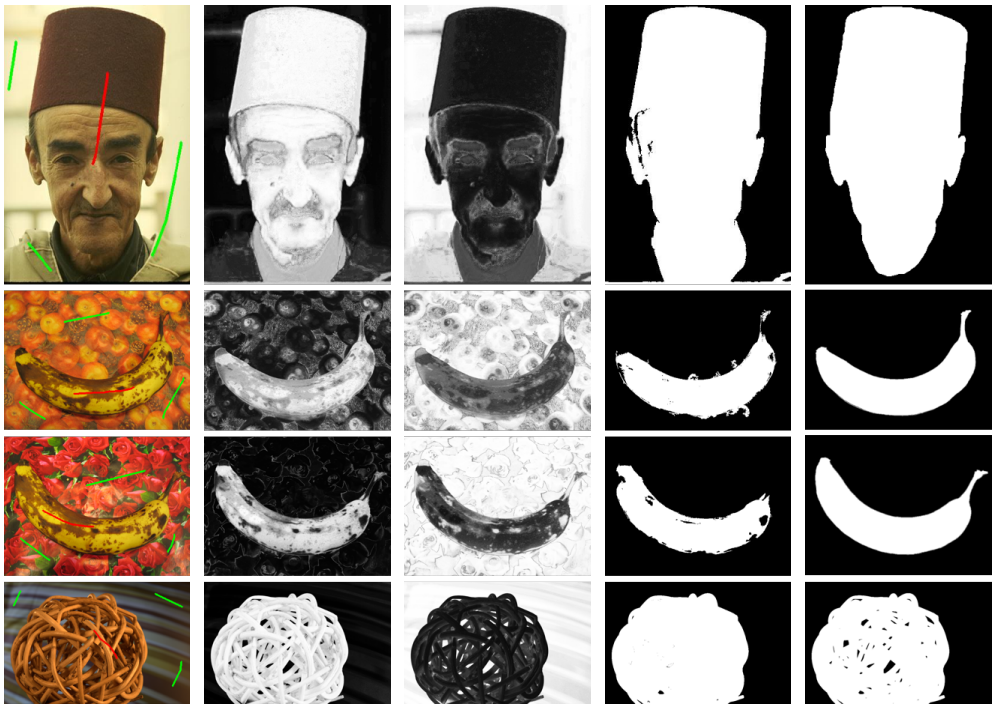
(a) Image (b) Marker-node (c) Segment (d) GT

FIGURE 5.8: Failed Interactive segmentation.

5.3.4 An extension in interactive segmentation based on the Dahu pseudo-distance

Experimental setting: This section aims to validate our proposed method for interactive segmentation that we present in Section 3.2.3.2. Here, we compare our extension method with the simple one that presented in Section 5.3.3. We also compare our proposed method with the grabcut method [23] and the state-of-the-art MB-based distances. Note that, in [24], the authors proposed a distance, named Minimum Spanning Distance (MSD). This distance is computed by counting the number of colors visited along a path. To be specific, they quantize the color space into discrete boxes and count the number of boxes instead of colors.

Qualitative results: Fig. 5.9 presents the results of our method on several examples which are failed in Fig. 5.8. Our method uses the statistic approach to exploit the prior knowledge from the scribbles, thereby demonstrating the probability of every pixel in the image concerning the scribbles. These images are illustrated in Fig. 5.9(b) and Fig. 5.9(c). From these probability images, the Dahu pseudo-distance is used to compute two saliency maps with regard to the two sets of scribble. Fig. 5.9(d) provides the results of our segmentation. Our method performs significantly better than Fig. 5.8, where we use the Dahu pseudo-distance directly on the image. Our results are also close to the ground truth. Our extended algorithm solves the problem that we met in the previous section.



(a) Image (b) GMM-FG (c) GMM-BG (d) Seg (e) GT

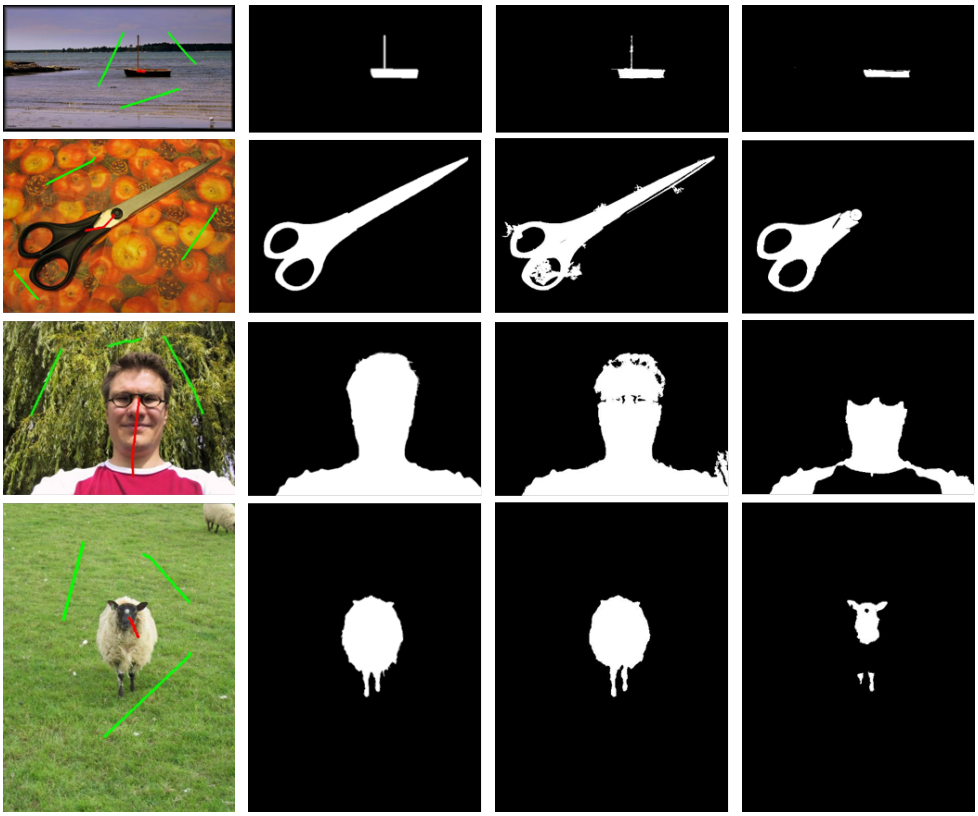
FIGURE 5.9: The qualitative results of our extension method for interactive segmentation. The original images along with the scribbles are presented in column (a); (b) and (c) respectively represent the probability of every pixel in the image with regard to the background and foreground scribbles; the segmentation results are illustrated in column (d) which are close to the ground truth in column (e).

We also present some qualitative results of our method compared to Grabcut method [23]. Both of these two methods exploit the background and foreground information based on the statistic approaches using GMM algorithm. While the Grabcut method segments the object region by using an energy minimisation method, our approach relies on the Dahu pseudo-distance. The results of our proposed and Grabcut method are illustrated respectively in Fig. 5.10(c) and Fig. 5.10(c). In the first row, our method is able to segment a very small detail of the boat while Grabcut method is not. The second row shows the image of a scissor. The propagation of our distance allows to segment correctly the object regions. In the third row and the last row, the Grabcut method requires some additional scribbles to completely segment the object region. On the contrary, our method provides satisfying results with the giving scribbles.

TABLE 5.2: A comparison of interactive segmentation between our proposed method and several state-of-the-art methods.

Distance	Geo [24]	MBD [24]	MSD16 [24]	MSD32 [24]	Grabcut [23]	Our
F-score	0.6469	0.6166	0.6821	0.6807	0.6392	**0.7143**

Evaluation: Table 5.2 presents some quantitative results of our method comparing to state-of-the-art approaches. We compare our method with some other path-wise distance metrics such as geodesic distance, MBD, and minimum spanning distance [24].



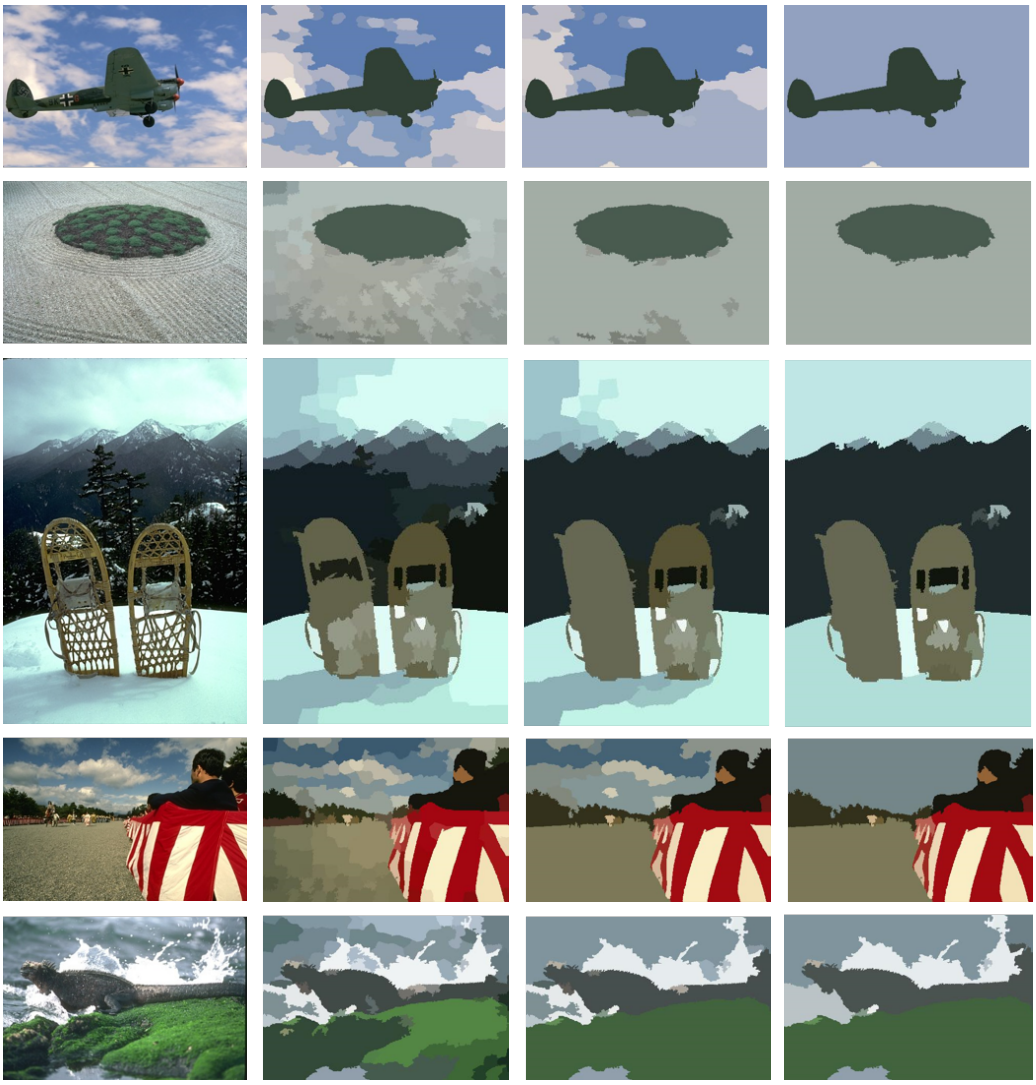
(a) Ori (b) GT (c) GMM-Dahu (d) Grabcut

FIGURE 5.10: Comparison on interactive segmentation between our proposed method and Grabcut method [23].

For color images, Geo and MBD are performed channel by channel and then we average the maps as final distance [24]. For minimum spanning distance [24], their methods use the box size 16 and 32. To evaluate the quality of interactive segmentation methods, we use the weighted F score. This metric can better evaluate the segmentation map since it takes into consideration the location of errors in the predicted maps. The higher weighted F means better segmentation performance. In the table, we can see that our proposed method achieves the best performance. It proves the robustness of our distance. It is able to obtain a good segmentation with very few knowledge about the background and foreground information (scribbles). Note that, the results of geodesic, minimum barrier, and minimum spanning distance are extracted from [24]. Our method achieves better results than Grabcut method, which is usually used in many interactive segmentation applications.

5.4 Image segmentation based on the Dahu pseudo-distance

Image simplification and segmentation is still a difficult challenge, which has been long studied in computer vision. In Section 2.5, we presented several state-of-the-art methods, which are used to segment an image into many meaningful regions. In this section, we demonstrate the proposed fast method for image segmentation based on our new distance which is presented in Section 3.2.4. It proves one more time the robustness and efficiency of the Dahu pseudo-distance in image processing



(a) Image (b) $\lambda_s = 10000$ (c) $\lambda_s = 20000$ (d) $\lambda_s = 30000$

FIGURE 5.11: Image segmentation.

application. The image segmentation is usually used as a preprocessing step for object detection and recognition. Our proposed method here is used for the document detection that we will present in Section 5.5.

Experimental setting: In this section, we present some setting details of the parameter that we use in our scheme. We also illustrate several examples of image segmentation using our method. The algorithm begins by reducing proportionally the size of the image so that the maximum so that the maximal dimension is 300 pixels. Then we convert the color image into gray-scale image. The SLIC algorithm [25] is adopted to segment an image into 300 super-pixels. To compute the distance between two neighbor super-pixels, α and β values are used respectively with $\alpha = 5$ and $\beta = 1$. The λ_s value is chosen to control the coarse level of the image segmentation.

Qualitative results: Our method provides a hierarchical segmentation. We apply it to BSDS500 dataset which is introduced in [52]. Several qualitative results of our

method are illustrated in Fig. 5.11. The higher value of λ_s is, the coarser segmentation is. The segmenting images are obtained by assigning to each region the median color of every pixel inside it. The regions in the segmenting image correspond to the survived nodes after cutting a tree. In the first and second images, there are two parts: the object and the background. Our method is able to segment the foreground region out of the image. Moreover, the contour of the object is very close to the real contour of the object. The three following images contain many object regions in the image. Our method can segment object region in the image. We obtain a homogeneous color to the sky and ground regions.

Our method is very fast. It can achieve 5 frames per second. The results that we present in this section are the primary results of our method. Therefore, we do not provide their evaluation results.

5.5 Document Detection based on the Dahu pseudo-distance

In this section, we evaluate the ability of the Dahu pseudo-distance for document detection. Firstly, we compare our simple method based on the Dahu pseudo-distance with other saliency based methods. Then, in the next section, we evaluate our extended version with several state-of-the-art methods in the end-to-end document detection.

5.5.1 Simple saliency based method for document detection

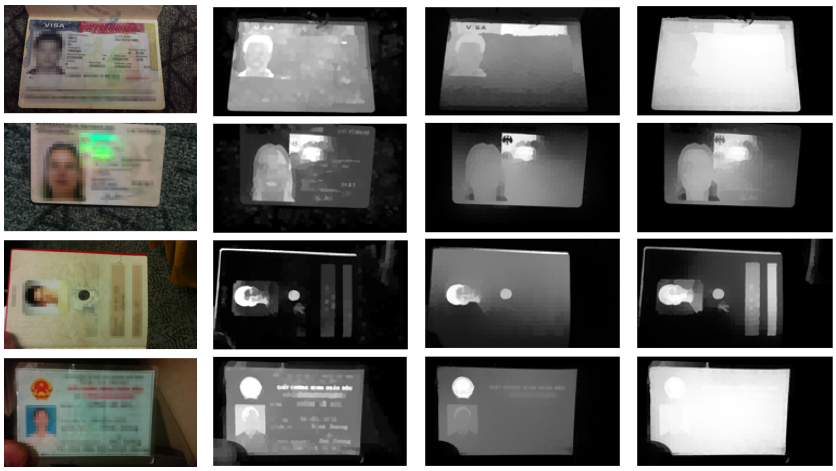
To know how our Dahu-distance-based saliency method performs in the context of identity document segmentation, we are going to compare it with some other similar approaches. This is a simple approach which is presented in Section 3.2.5.1.

5.5.1.1 Experiment setting

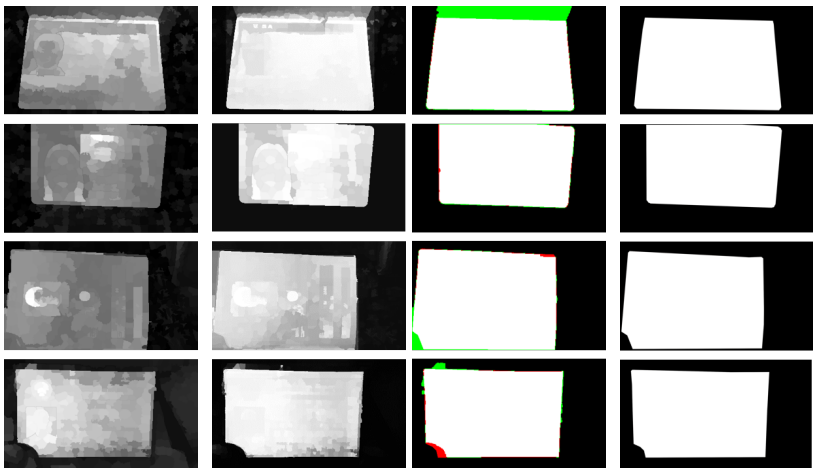
Let us now present three state-of-the-art methods of salient object detection, that we are going to compare our method with. In [22] the saliency detection is based on a geodesic distance (**GS**) which uses background priors. The major assumptions are that the background is usually large, homogeneous, and located near the boundary of the image. In [140] the saliency detection relies on a bottom-up approach to choose some regions by manifold ranking (**MR**) on a graph of superpixels. Such as in Eq. (3.17), the authors compute 4 maps and fuse them. In these maps, the superpixels are ranked w.r.t. the similarity with some seeds located in the image boundaries. In [141], a saliency optimization method (**SO**) is proposed which combines multiple saliency measures, one of them using the notion of "boundary connectivity". Note that all these methods also rely on a post-processing step to "normalize" the resulting saliency maps.

For our experiments, we have built a dataset of identity documents available at *http://publications.lrde.epita.fr/movn.18.das*. We have a dozen of different types of visas and passports from various countries. We recorded over 100 videos under different environment conditions, using several kinds of smartphones. From these videos, we selected 100 frames to create our dataset, so that it presents some realistic difficulties such as out-of-focus and motion blur, inhomogeneous illuminations, etc. Then, we generated manually the corresponding ground-truth images.

We compare our method with the state-of-the-art saliency-based detection methods presented in the previous section. We use two distinct measures:



(a) Input (b) GS [22] (c) MR [140] (d) SO [141]



(e) Dahu-based... (f) ...normalized (g) Detection (h) Ground truth

FIGURE 5.12: Comparison of our saliency maps with other classical or state-of-the-art methods.

1. the Mean Absolute Error (MAE), which is the average difference between a saliency map S (gray-level image) and a ground-truth image GT (binary image): $\text{MAE} = (\sum_x |GT(x) - S(x)|) \,/\, N$, with N being the number of pixels.

2. an F_β-measure defined by: $F_\beta = (1+\beta^2) \times P \times R \,/\, (\beta^2 \times P + R)$, where P and R are respectively the precision and the recall, and with $\beta^2 = 0.3$ (it is the classical setting in the visual saliency community).

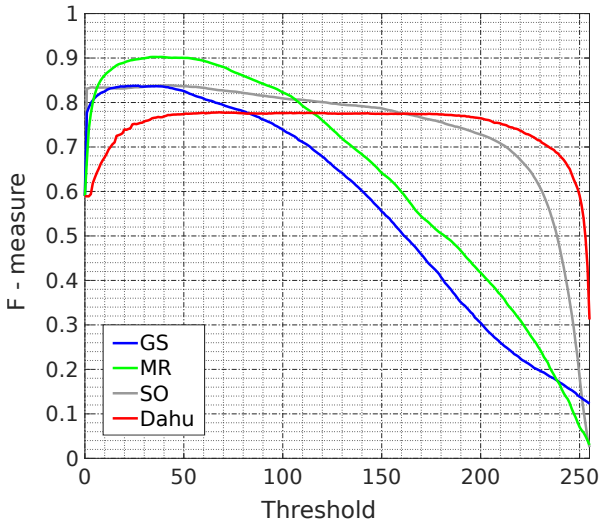
5.5.1.2 Experimental Results

To compute the precision and recall scores, for each image to process, we simply binarize the corresponding gray-level saliency map with a threshold sliding from 0 to 255. Then, for every threshold, we compare the obtained binary map with the ground-truth map. For a given threshold, we depict in Fig. 5.13(b) the average F_β-measure obtained on the dataset of 100 images. The "global" F_β-measure, averaged for all thresholds (and all images), is denoted by $\overline{F_\beta}$.

The values of $\overline{F_\beta}$ and the MAE scores for all the compared methods are depicted in the table in Fig. 5.13(a); note that the better a method is, the lower MAE values

Method	MAE	$\overline{F_\beta}$
GS [22]	0.328	0.573
MR [140]	0.299	0.642
SO [141]	0.265	**0.7461**
Dahu	**0.178**	**0.7465**

(a) MAE and $\overline{F_\beta}$



(b) F_β w.r.t. saliency map thresholding

FIGURE 5.13: Numerical comparison of saliency maps.

are, and the higher $\overline{F_\beta}$ values are. First, we can observe that, over the years, the state-of-the-art methods give better results (first GS, then MR, and last SO). Second, the Dahu-based approach gives the lowest MAE score, and slightly outperforms the SO method for the $\overline{F_\beta}$ criterion.

If we look at the F_β-measure curves for the different thresholds in Fig. 5.13(b), there are two main observations. First, the methods SO (in gray) and Dahu (in red) have stables / flat curves, which is an advantage, because the "best" threshold remains unknown and depends on the image. Conversely, for the GS and MR methods (respectively in blue and green), the curves are not stable, which means that taking a threshold might not be a very robust task. The second observation is that the "best" method with respect to the F_β-measure seems to be the MR method, with a rather low threshold (around 50). Though, the MR method is computationally expensive so it cannot run in real-time on smartphones, whereas the Dahu-based approach can.

Some qualitative illustrations on a few images (Fig. 5.12(a)) are depicted in Fig. 5.12. The prominent observation is that the compared saliency methods, from Fig. 5.12(b) to Fig. 5.12(f), have rather different behaviors. The one based on the Dahu distance, so on the principle of a *barrier* (see Eq. (2.32), Eq. (2.39), and Eq. (3.2)) is effective: the main barrier is visible around the documents, even *before* normalization; see Fig. 5.12(e). Also we can notice that the saliency values *inside* the documents are much more uniform with the Dahu-based method than with the other saliency-based methods.

5.5.1.3 Limitation

The major limitation of saliency-based methods is due to low contrast; some failure cases are depicted in Fig. 5.14. The left image is blurred and the contrast between the document and the background is poor, so the document cannot be detected. In the right image, the identity card has a color similar to the one of the background, so the salient objects are the hand and the portrait. Actually, as perspectives, the method

(a) Image (b) Saliency

(c) Image (d) Saliency

FIGURE 5.14: Some failure cases of the Dahu-based approach.

we present can be improved through taking into account some extra prior information such as "text texture", and can be combined with more classical contour/line-based approaches. An extended version for document detection is presented in Section 5.5.2.

5.5.2 Extended saliency based method for document detection

In this section, we evaluate our completed frame for document detection based on the Dahu pseudo-distance, which is introduced in Section 3.2.5.

5.5.2.1 Dataset and Evaluation

To perform the evaluation, we use the ICDAR 2015 SmartDoc challenge 1 dataset [30]. These videos are taken by a Google Nexus 7 tablet for a total of 25K frames with a resolution of 1920 $\times$ 1080 on six types of document, that are placed over 5 different backgrounds. The document pages are placed inside the image (and never hit the boundary of the image). The dataset is challenging (variable lighting condition, inhomogeneous background, motion blur and out-of-focus blur). Especially, the fifth background is complex with many objects placed near the document or even over it.

To evaluate the performance of the method, the Jaccard index between the detected document A and the ground truth G is used:

$$JI = area(G \cap A)/area(G \cup A) \tag{5.1}$$

5.5.2.2 Experiments and Results

We start with reducing the size of each frame by a factor of 2. We also convert an image to $L^*a^*b^*$ space to mimic the human vision. Then the ToS is built on L^* and b^* channels of each frame (the contrast between the document and the background is not sufficient on the other channel).

Method	Bg 1	Bg 2	Bg 3	Bg 4	Bg 5	Overall	Runtime
A2iA-1	0.972	0.801	0.912	0.635	0.189	0.779	?
A2iA-2	0.960	0.806	0.912	0.826	0.189	0.809	?
ISPL-CVML	0.987	0.965	0.985	0.977	0.856	0.966	?
LRDE [26]	0.987	0.978	0.989	0.984	0.861	0.972	**1min**
NetEase	0.962	0.955	0.962	0.951	0.222	0.882	?
SEECS-NUST	0.888	0.826	0.783	0.781	0.011	0.739	?
RPPDI-UPE	0.827	0.910	0.970	0.365	0.216	0.741	?
SmartEngines [27]	0.989	0.983	0.990	0.979	0.688	0.955	?
L. R. S. Leal [28]	0.961	0.944	0.965	0.930	0.412	0.895	0.43s
LRDE-2 [29]	0.905	0.936	0.859	0.903	?	?	0.04s
Ours	0.985	0.982	0.987	0.980	0.848	0.97	**3.7s**
Smartdoc ave. [30]	0.9465	0.9031	0.9377	0.8122	0.4041	0.8552	?

FIGURE 5.15: Quantitative results on Smartdoc 2015 competitions data. The red (resp. blue) color denotes the best (resp. second) result in each background. Our method gets the second highest overall score. It is competitive with the LRDE method [26], but about 20 times faster than their method.

The SLIC algorithm [25] is adopted to segment an image into 300 super-pixels. The values $\alpha = 5$ and $\beta = 1$ are chosen to emphasize the Dahu distance. Variations on them do not change results so far. The value $\lambda_s = 8000$ is low enough to avoid under-segmentation of the document.

Quantitative results on the Smartdoc 2015 dataset are shown in Fig. 5.15. Our method achieves the second highest overall score over 12 methods. The difference with the first ranked method (LRDE) is negligible (0.972 vs 0.97), but we are about 16 times faster (1 min vs 3.7s). Our method is better than the other methods in the competition (even with SmartEngines method [27] which is ranked first on background 1, 2 and 3). Especially, it fails on the most difficult case: background 5 (shortly Bg. 5). In this evaluation, we do not compare our method with SEECS-NUST-2 [235] method because of the following reasons: they use highly correlated training and testing data. For background 5, they used 50% of each video for training, next 20% for validation, and only 30% for testing. It is not a good strategy because:

- the training and testing dataset are too much similar (the accuracy on Bg. 5 decreased from 0.94 to 0.66 when all samples extracted from Bg. 5 "testing video" were removed from training [235]),
- the testing dataset is different from the other methods.

In Fig. 5.16, we show the results of our method on some challenging images. Our method is well handled with blurred, illumination variation cases. Even in some tedious cases such as the superposition of documents, non-straight boundaries document, partially occluded documents or the document slightly hits the boundary of the image, our method succeeds.

Concerning the tests, we used an Intel i7 2.6 GHz CPU with 8 GB of RAM. The speed can be improved as we use a naïve implementation of the method. The total time (excluding I/O time) of our method depends on the size of the image and the number of super-pixels. Fig. 5.17 demonstrates the compromise between the executed time of the process and the overall score. If we increase the scaling parameter and decrease the number of super-pixels, the executed time is much shortened, while the accuracy remains acceptable. Our method achieves an overall score of

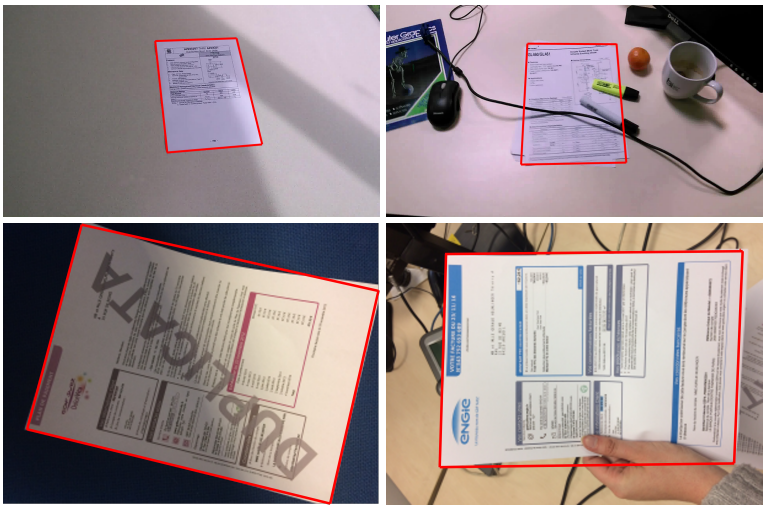


FIGURE 5.16: Some qualitative results of our method. These images show the robustness of our method to illumination, blur and curled document.

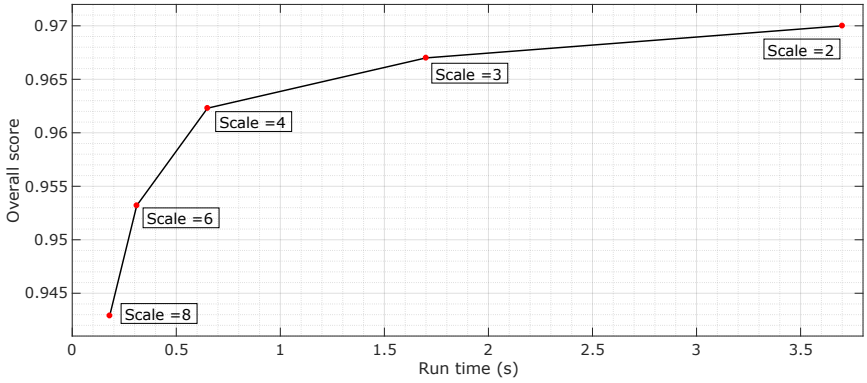


FIGURE 5.17: The compromise between the executed time (image resolution i.e. image scale) and the overall accuracy. Even at low resolution, our method achieves an overall score of 0.962 for a run time equal to 0.65s.

0.962 at run time equal to 0.65 second, which is almost 100 times faster than the method of the LRDE [26].

In this section, we have proposed a framework to detect documents in photos or videos captured by smartphones based on saliency maps, with very few prior knowledges about the documents and the images. We only take into account that the document looks like a quadrilateral and does not mostly touch the image boundary. Our main conclusion (and contribution) is that visual saliency approaches are relevant to document detection. Moreover, while remaining efficient, which is critical in embedded software, we have the potential to offer better results than the one presented here, using some extra knowledge.

Chapter 6

Conclusion

In this thesis, we have studied the Dahu pseudo-distance, which can be considered a continuous version of the Minimum Barrier Distance (MBD), and we have proposed several improvements to it.

The MB-based distances have been proved to be robust for pixel fluctuation. Since these distances were initially developed for grayscale images, they have many limitations, and there is still room for improvement, especially for color images, which are not well handled. Therefore, in our work, we have introduced a vectorial extension of the Dahu pseudo-distance, which is able to deal with multichannels images. Our proposed method is implemented by using the multivariate tree of shapes, which is a version of the tree of shapes extended to multivariate images. Obviously, this vectorial Dahu pseudo-distance can manage color images well which already exposes a great improvement. Nevertheless, our distance is not restricted to three channels images, but it is also applied on multispectral/multimodal images. Through several experiments, the vectorial Dahu pseudo-distance significantly achieves better performance than the Dahu pseudo-distances on grayscale images and separate channels. The robustness of the vectorial Dahu pseudo-distance can be explained as follows: the tree of shapes on the color image contains more information and is well structured. Hence, this extended Dahu pseudo-distance is promising for many image processing applications.

We have improved the Dahu pseudo-distance by combining the pseudo-distance with information on the spatial domain of the images. The optimal path found by our proposed method has an interesting meaning. This path is not only the shortest path in the "color space" (tree space) but also the shortest path in the image space. Thus, our approach can be efficiently used for shortest path finding applications. Typically, we can use this distance to segment a blood vessel.

We have compared our new distance to MB-based distances in some applications. We have shown that the vectorial Dahu pseudo-distance is less affected by noise in the image than other MB-based distances. It can be explained that each node on the tree of shapes is associated with the median value of all pixels in the node, thereby reducing the impact of noise in the image. Additionally, our proposed distance is more contrasted than MB-based distances. It is because the Dahu pseudo-distance is computed on tree of shapes, which considers an image to be a surface and scalar value to be replaced by an interval. Therefore, the Dahu pseudo-distance tends to decrease its path cost between pixels in the same background while retaining the contrast between objects and background.

We also have demonstrated an improvement of the vectorial Dahu pseudo-distance in dealing with multimodal and multispectral images through experiments on multimodal medical images and multispectral satellite images. Specially, we can

observe that our distance has successfully been employed in these kinds of applications.

Another advantage of this new vectorial version is that it can be instantly computed on the tree of shapes. Thanks to a clever representation of images, the multivariate tree of shapes, the execution time to compute this distances in the tree is extremely fast (and the tree can be computed in a quasi-linear time w.r.t. the number of pixels of the images). This is useful when we want to compute multiple distances between pixels in the image.

With the diversity of advantages stated above, we have tested the Dahu pseudo-distance in multiple applications, such as interactive segmentation and salient object detection. Specifically, we have developed a new method for document detection in videos captured by smartphones. We have achieved high performance with very little *a priori* knowledge on the document and the images. In addition, we have shown the efficiency of the visual saliency for document detection. Our detection scheme is very fast and offers a good compromise between speed and accuracy. It is worth noting that severe images, such as, blurred, illumination variation, non-straight boundaries or partially occluded documents, can be processed efficiently by our approach. Based on the lessons that we have learned, one future direction clearly stands out. As our scheme is efficient in the aspect of processing ability, we plan to extend our work in embedded smartphone direction.

However, the Dahu pseudo-distance still has limitations. First, to compute the Dahu pseudo-distance, we have to construct a ToS, in the case of gray images and MToS, in the case of multivariate images. Although the execution time to compute the tree of shapes is fast, we can not reach 30 frames per second for FullHD images. Due to the additional time for the computation of the tree, the Dahu pseudo-distance can not be used in the real-time applications. Secondly, although our distance has some interesting properties, using only the Dahu pseudo-distance is not sufficient for many applications. Therefore, it is necessary to combine the Dahu pseudo-distance with many other features to improve the results.

The hierarchical representation in our segmentation method is computed by using the Dahu pseudo-distance. For future work, we plan to apply machine learning techniques to improve the results of hierarchical image segmentation. By learning other features, such as the gradient values and the textures in each region, we would be able to achieve good segmentation results.

For the salient object detection application, we applied our Dahu pseudo-distance to compute the shortest path between each pixel in the image to the border of the image, thereby generating the saliency map. This approach highly depends on the background assumption (objects do not touch or only partial touch the background), which might not be true in the natural images. Moreover, computing the saliency maps based on the Dahu pseudo-distance is not sufficient. We might need to combine our results with other approaches to improve the final results of our method, for example, contrast prior or graph-based methods. Furthermore, we can envision a multi-scale model that can take the best from hierarchical image segmentation to achieve better saliency maps.

This thesis is an opportunity for us to demonstrate the robustness of the Dahu pseudo-distance. We believe that this new pseudo-distance can be applied to many image processing and computer vision applications.

Bibliography

[1] R. Jones, "Component trees for image filtering and segmentation," in *Proceedings of the 1997 IEEE Workshop on Nonlinear Signal and Image Processing, Mackinac Island*, 1997.

[2] P. Salembier, A. Oliveras, and L. Garrido, "Antiextensive connected operators for image and sequence processing," *IEEE Transactions on Image Processing*, vol. 7, no. 4, pp. 555–570, 1998.

[3] V. Caselles, B. Coll, and J.-M. Morel, "Geometry and color in natural images," *Journal of Mathematical Imaging and Vision*, vol. 16, no. 2, pp. 89–105, 2002.

[4] P. J. Toivanen, "New geodosic distance transforms for gray-scale images," *Pattern Recognition Letters*, vol. 17, no. 5, pp. 437–450, 1996.

[5] R. Strand *et al.*, "The minimum barrier distance," *Comp. Vision and Image Understanding*, vol. 117, no. 4, pp. 429–437, 2013.

[6] K. C. Ciesielski *et al.*, "Efficient algorithm for finding the exact minimum barrier distance," *Computer Vision and Image Understanding*, vol. 123, pp. 53–64, 2014.

[7] R. Strand *et al.*, "The minimum barrier distance: A summary of recent advances," in *Proc. of DGCI*, ser. LNCS, vol. 10502. Springer, 2017, pp. 57–68.

[8] X. Huang and Y. Zhang, "Water flow driven salient object detection at 180 fps," *Patt. Rec.*, vol. 76, pp. 95–107, 2018.

[9] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech, "Minimum barrier salient object detection at 80 fps," in *Proc. of ICCV*, 2015, pp. 1404–1412.

[10] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien, "Real-time salient object detection with a minimum spanning tree," in *Proc. of CVPR*, 2016, pp. 2334–2342.

[11] B. Yang, X. Zhang, L. Chen, H. Yang, and Z. Gao, "Edge guided salient object detection," *Neurocomputing*, vol. 221, pp. 60–71, 2017.

[12] A. Wang and M. Wang, "Rgb-d salient object detection via minimum barrier distance transform and saliency fusion," *IEEE Signal Processing Letters*, vol. 24, no. 5, pp. 663–667, 2017.

[13] G. Wang, Y. Zhang, and J. Li, "High-level background prior based salient object detection," *Journal of Visual Communication and Image Representation*, vol. 48, pp. 432–441, 2017.

[14] F. Malmberg, R. Nordenskjöld, R. Strand, and J. Kullberg, "Smartpaint: a tool for interactive segmentation of medical volume images," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 5, no. 1, pp. 36–44, 2017.

[15] M. Grand-Brochier, A. Vacavant, R. Strand, G. Cerutti, and L. Tougne, "About the impact of pre-processing tools on segmentation methods applied for tree leaves extraction," in *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, vol. 1. IEEE, 2014, pp. 507–514.

[16] S. P. Bharati, S. Nandi, Y. Wu, Y. Sui, and G. Wang, "Fast and robust object tracking with adaptive detection," in *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*. IEEE, 2016, pp. 706–713.

[17] E. Carlinet and T. Géraud, "MToS: A tree of shapes for multivariate images," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5330–5342, 2015.

[18] 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 ISMM*, ser. LNCS, vol. 7883, 2013, pp. 98–110.

[19] T. Géraud, Y. Xu, E. Carlinet, and N. Boutry, "Introducing the Dahu pseudo-distance," in *Proc. of ISMM*, ser. LNCS, vol. 10225, 2017, pp. 55–67.

[20] A. Kårsnäs, R. Strand, and P. K. Saha, "The vectorial minimum barrier distance," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 792–795.

[21] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[22] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," *Proc. of ECCV*, pp. 29–42, 2012.

[23] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.

[24] C.-T. Chou, W.-C. Tu, and S.-Y. Chien, "Minimum spanning distance for image segmentation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 1568–1572.

[25] R. Achanta *et al.*, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[26] Y. Xu, E. Carlinet, T. Géraud, and L. Najman, "Hierarchical segmentation using tree-based shape spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 457–469, 2017.

[27] A. Zhukovsky *et al.*, "Segments graph-based approach for document capture in a smartphone video stream," in *Proc. of ICDAR*, vol. 1. IEEE, 2017, pp. 337–342.

[28] L. R. Leal and B. L. Bezerra, "Smartphone camera document detection via geodesic object proposals," in *Computational Intelligence (LA-CCI), 2016 IEEE Latin American Conference on*. IEEE, 2016, pp. 1–6.

[29] É. Puybareau and T. Géraud, "Real-time document detection in smartphone videos," in *Proc. of IEEE ICIP*, 2018, pp. 1498–1502.

[30] J.-C. Burie *et al.*, "ICDAR 2015 competition on smartphone document capture and OCR (SmartDoc)," in *Proc. of ICDAR*, 2015, pp. 1161–1165.

[31] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[32] E. Shusterman and M. Feder, "Image compression via improved quadtree decomposition algorithms," *IEEE Transactions on Image Processing*, vol. 3, no. 2, pp. 207–215, 1994.

[33] G. K. Ouzounis and P. Soille, "Pattern spectra from partition pyramids and hierarchies," in *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer, 2011, pp. 108–119.

[34] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," *IEEE transactions on Image Processing*, vol. 9, no. 4, pp. 561–576, 2000.

[35] J. F. Randrianasoa, C. Kurtz, E. Desjardin, and N. Passat, "Binary partition tree construction from multiple features for image segmentation," *Pattern Recognition*, vol. 84, pp. 237–250, 2018.

[36] V. Vilaplana, F. Marques, and P. Salembier, "Binary partition trees for object detection," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2201–2216, 2008.

[37] P. Hanusse and P. Guillataud, "Sémantique des images par analyse dendronique," in *8th Conf. Reconnaissance des Formes et Intelligence Artificielle*, vol. 2, 1992, pp. 577–588.

[38] M. Couprie and G. Bertrand, "Topological gray-scale watershed transformation," in *Vision Geometry Vi*, vol. 3168. International Society for Optics and Photonics, 1997, pp. 136–147.

[39] Y. Xu, T. Géraud, and L. Najman, "Connected filtering on tree-based shape-spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–14, 2015, to appear.

[40] ——, "Hierarchical image simplification and segmentation based on Mumford-Shah-salient level line selection," *Pattern Recognition Letters*, vol. 83, no. 3, pp. 278–286, 2016.

[41] G. K. Ouzounis and M. H. Wilkinson, "Second-order connected attribute filters using max-trees," in *Mathematical Morphology: 40 Years On*. Springer, 2005, pp. 65–74.

[42] Y. Xu, T. Géraud, and L. Najman, "Two applications of shape-based morphology: Blood vessels segmentation and a generalization of constrained connectivity," in *Proc. of the Intl. Symp. on Mathematical Morphology (ISMM)*, ser. Lecture Notes in Computer Science, vol. 7883. Springer, 2013, pp. 390–401.

[43] C. Berger, T. Géraud, R. Levillain, N. Widynski, A. Baillard, and E. Bertin, "Effective component tree computation with application to pattern recognition in astronomical imaging," in *Proc. of the IEEE Intl. Conf. on Image Processing (ICIP)*, vol. 4, 2007, pp. IV–41.

[44] J. Mattes, M. Richard, and J. Demongeot, "Tree representation for image matching and object recognition," in *International Conference on Discrete Geometry for Computer Imagery*. Springer, 1999, pp. 298–309.

[45] Y.-J. Chiang, T. Lenz, X. Lu, and G. Rote, "Simple and optimal output-sensitive construction of contour trees using monotone paths," *Computational Geometry*, vol. 30, no. 2, pp. 165–195, 2005.

[46] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on pure and applied mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[47] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *International journal of computer vision*, vol. 50, no. 3, pp. 271–293, 2002.

[48] B. R. Kiran and J. Serra, "Global–local optimizations by hierarchical cuts and climbing energies," *Pattern Recognition*, vol. 47, no. 1, pp. 12–24, 2014.

[49] T. Liu, M. Seyedhosseini, and T. Tasdizen, "Image segmentation using hierarchical merge tree," *IEEE transactions on image processing*, vol. 25, no. 10, pp. 4596–4607, 2016.

[50] M.-M. Cheng, Y. Liu, Q. Hou, J. Bian, P. Torr, S.-M. Hu, and Z. Tu, "Hfs: Hierarchical feature selection for efficient image segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 867–882.

[51] Y. Chen, D. Dai, J. Pont-Tuset, and L. Van Gool, "Scale-aware alignment of hierarchical image segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 364–372.

[52] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2010.

[53] P. Soille, *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.

[54] R. A. Finkel and J. L. Bentley, "Quad trees a data structure for retrieval on composite keys," *Acta informatica*, vol. 4, no. 1, pp. 1–9, 1974.

[55] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.

[56] P. Monasse and F. Guichard, "Fast computation of a contrast-invariant image representation," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 860–872, 2000.

[57] S. Valero, P. Salembier, and J. Chanussot, "Comparison of merging orders and pruning strategies for binary partition tree in hyperspectral data," in *2010 IEEE International Conference on Image Processing*. IEEE, 2010, pp. 2565–2568.

[58] P. Salembier and M. H. Wilkinson, "Connected operators," *IEEE Signal Processing Magazine*, vol. 26, no. 6, pp. 136–157, 2009.

[59] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman, "Geodesic star convexity for interactive image segmentation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 3129–3136.

[60] R. C. Gonzalez, R. E. Woods *et al.*, "Digital image processing [m]," *Publishing house of electronics industry*, vol. 141, no. 7, 2002.

[61] A. Rosenfeld, *Digital picture processing*. Academic press, 1976.

[62] L. Najman and J. Cousty, "A graph-based mathematical morphology reader," *Pattern Recognition Letters*, vol. 47, pp. 3–17, 2014.

[63] T. Y. Kong and A. Rosenfeld, "Digital topology: Introduction and survey," *Computer Vision, Graphics, and Image Processing*, vol. 48, no. 3, pp. 357–393, 1989.

[64] T. C. Hales, "The jordan curve theorem, formally and informally," *The American Mathematical Monthly*, vol. 114, no. 10, pp. 882–894, 2007.

[65] E. Khalimsky, R. Kopperman, and P. R. Meyer, "Computer graphics and connected topologies on finite ordered sets," *Topology and its Applications*, vol. 36, no. 1, pp. 1–17, 1990.

[66] J. Shen, X. Hao, Z. Liang, Y. Liu, W. Wang, and L. Shao, "Real-time superpixel segmentation by dbscan clustering algorithm," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5933–5942, 2016.

[67] S. Beucher and F. Meyer, "The morphological approach to segmentation: the watershed transformation," *Optical Engineering-New York-Marcel Dekker Incorporated-*, vol. 34, pp. 433–433, 1992.

[68] V. Machairas, M. Faessel, D. C.-P. na, T. Chabardes, T. Walter, and E. Decencière, "Waterpixels," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3707–3716, 2015.

[69] L. Guigues, J. P. Cocquerez, and H. Le Men, "Scale-sets image analysis," *International Journal of Computer Vision*, vol. 68, no. 3, pp. 289–317, 2006.

[70] L. Najman and M. Couprie, "Building the component tree in quasi-linear time," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3531–3539, 2006.

[71] I. Ahmad and W. I. Grosky, "Spatial similarity-based retrievals and image indexing by hierarchical decomposition," in *Proceedings of the 1997 International Database Engineering and Applications Symposium (Cat. No. 97TB100166)*. IEEE, 1997, pp. 269–278.

[72] E. Albuz, E. D. Kocalar, and A. A. Khokhar, "Quantized cielab* space and encoded spatial structure for scalable indexing of large color image archives," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, vol. 4. IEEE, 2000, pp. 1995–1998.

[73] H.-K. Kim and J.-D. Kim, "Region-based shape descriptor invariant to rotation, scale and translation," *Signal Processing: Image Communication*, vol. 16, no. 1-2, pp. 87–93, 2000.

[74] S. Lin, M. T. Ozsu, V. Oria, and R. Ng, "An extendible hash for multi-precision similarity querying of image databases," in *VLDB*, vol. 1, 2001, pp. 221–230.

[75] M. Rukoz, M. Manouvrier, and G. Jomier, "Distances de similarité d'images basées sur les arbres quaternaires," in *18èmes Journées Bases de Données Avancées (BDA'02), Evry (France)*, 2002, pp. 307–326.

[76] V. P. Baligar, L. M. Patnaik, and G. Nagabhushana, "High compression and low order linear predictor for lossless coding of grayscale images," *Image and Vision Computing*, vol. 21, no. 6, pp. 543–550, 2003.

[77] H. Cheng and L. Xiaobo, "On the application of image decomposition to image compression and encryption," in *Communications and Multimedia Security II*. Springer, 1996, pp. 116–127.

[78] H.-S. Kim and J.-Y. Lee, "Image coding by fitting rbf-surfaces to subimages," *Pattern Recognition Letters*, vol. 23, no. 11, pp. 1239–1251, 2002.

[79] D. J. Jackson, W. Mahmoud, W. A. Stapleton, and P. T. Gaughan, "Faster fractal image compression using quadtree recomposition," *Image and Vision Computing*, vol. 15, no. 10, pp. 759–767, 1997.

[80] T.-W. Lin, "Compressed quadtree representations for storing similar images," *Image and Vision Computing*, vol. 15, no. 11, pp. 833–843, 1997.

[81] ——, "Set operations on constant bit-length linear quadtrees," *Pattern Recognition*, vol. 30, no. 7, pp. 1239–1249, 1997.

[82] H. Samet and R. E. Webber, "Hierarchical data structures and algorithms for computer graphics. i. fundamentals," *IEEE Computer Graphics and applications*, vol. 8, no. 3, pp. 48–68, 1988.

[83] F. De Natale and F. Granelli, "Structured-based image retrieval using a structured color descriptor," in *Int. Workshop on Content-Based Multimedia Indexing (CBMI'01)*, 2001, pp. 109–115.

[84] O. Boruvka, "O jistém problému minimálním," *Práce Mor. Prırodved. Spol. v Brne (Acta Societ. Scienc. Natur. Moravicae)*, vol. 3, no. 3, pp. 37–58, 1926.

[85] R. C. Prim, "Shortest connection networks and some generalizations," *The Bell System Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957.

[86] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *Journal of the ACM (JACM)*, vol. 34, no. 3, pp. 596–615, 1987.

[87] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[88] L. An, Q.-S. Xiang, and S. Chavez, "A fast implementation of the minimum spanning tree method for phase unwrapping," *IEEE transactions on medical imaging*, vol. 19, no. 8, pp. 805–808, 2000.

[89] Y. Xu and E. C. Uberbacher, "2d image segmentation using minimum spanning trees," *Image and Vision Computing*, vol. 15, no. 1, pp. 47–57, 1997.

[90] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International journal of computer vision*, vol. 59, no. 2, pp. 167–181, 2004.

[91] A. Fahad and T. Morris, "A faster graph-based segmentation algorithm with statistical region merge," in *International Symposium on Visual Computing*. Springer, 2006, pp. 286–293.

[92] M. Zhang and R. Alhajj, "Improving the graph-based image segmentation method," in *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*. IEEE, 2006, pp. 617–624.

[93] M. Suk and O. Song, "Curvilinear feature extraction using minimum spanning trees," *Computer vision, graphics, and image processing*, vol. 26, no. 3, pp. 400–411, 1984.

[94] L. Najman and T. Géraud, "Discrete set-valued continuity and interpolation," in *Proc. of the Intl. Symp. on Mathematical Morphology (ISMM)*, ser. Lecture Notes in Computer Science, vol. 7883. Springer, 2013.

[95] Y. Xu, V. Olman, and D. Xu, "Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees," *Bioinformatics*, vol. 18, no. 4, pp. 536–545, 2002.

[96] C. T. Zahn, "Graph theoretical methods for detecting and describing gestalt clusters," *IEEE Trans. Comput.*, vol. 20, no. SLAC-PUB-0672-REV, p. 68, 1970.

[97] C. Zhong, D. Miao, and R. Wang, "A graph-theoretical clustering method based on two rounds of minimum spanning trees," *Pattern Recognition*, vol. 43, no. 3, pp. 752–766, 2010.

[98] C. Zhong, D. Miao, and P. Fränti, "Minimum spanning tree based split-and-merge: A hierarchical clustering method," *Information Sciences*, vol. 181, no. 16, pp. 3397–3410, 2011.

[99] P. Juszczak, D. M. Tax, E. Pe, R. P. Duin *et al.*, "Minimum spanning tree based one-class classifier," *Neurocomputing*, vol. 72, no. 7-9, pp. 1859–1869, 2009.

[100] K. Li, S. Kwong, J. Cao, M. Li, J. Zheng, and R. Shen, "Achieving balance between proximity and diversity in multi-objective evolutionary algorithm," *Information Sciences*, vol. 182, no. 1, pp. 220–242, 2012.

[101] F. Meyer and P. Maragos, "Nonlinear scale-space representation with morphological levelings," *Journal of Visual Communication and Image Representation*, vol. 11, no. 2, pp. 245–265, 2000.

[102] P. Soille, "Constrained connectivity for hierarchical image partitioning and simplification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 7, pp. 1132–1145, 2008.

[103] L. Najman, J. Cousty, and B. Perret, "Playing with kruskal: algorithms for morphological trees in edge-weighted graphs," in *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer, 2013, pp. 135–146.

[104] F. Merciol and S. Lefèvre, "Fast image and video segmentation based on alpha-tree multiscale representation," in *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems*. IEEE, 2012, pp. 336–342.

[105] P. Bosilj, S. Lefèvre, and E. Kijak, "Hierarchical image representation simplification driven by region complexity," in *International Conference on Image Analysis and Processing*. Springer, 2013, pp. 562–571.

[106] G. K. Ouzounis and P. Soille, "The alpha-tree algorithm," *Publications Office of the European Union*, 2012.

[107] H. Lu, J. C. Woods, and M. Ghanbari, "Binary partition tree for semantic object extraction and image segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 3, pp. 378–383, 2007.

[108] A. Alonso-Gonzalez, S. Valero, J. Chanussot, C. Lopez-Martinez, and P. Salembier, "Processing multidimensional sar and hyperspectral images with binary partition tree," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 723–747, 2012.

[109] A. Alonso-González, C. López-Martínez, and P. Salembier, "Filtering and segmentation of polarimetric sar data based on binary partition trees," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 2, pp. 593–605, 2011.

[110] G. Tochon, J.-B. Feret, S. Valero, R. E. Martin, D. E. Knapp, P. Salembier, J. Chanussot, and G. P. Asner, "On the use of binary partition trees for the tree crown segmentation of tropical rainforest hyperspectral images," *Remote sensing of environment*, vol. 159, pp. 318–331, 2015.

[111] M. A. Veganzones, G. Tochon, M. Dalla-Mura, A. J. Plaza, and J. Chanussot, "Hyperspectral image segmentation using a new spectral unmixing-based binary partition tree representation," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3574–3589, 2014.

[112] S. Valero, P. Salembier, and J. Chanussot, "Object recognition in hyperspectral images using binary partition tree representation," *Pattern Recognition Letters*, vol. 56, pp. 45–51, 2015.

[113] P. Monasse and F. Guichard, "Fast computation of a contrast-invariant image representation," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 860–872, 2000.

[114] E. Carlinet and T. Géraud, "A comparative review of component tree computation algorithms," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3885–3895, 2014.

[115] D. Nistér and H. Stewénius, "Linear time maximally stable extremal regions," in *European Conference on Computer Vision*. Springer, 2008, pp. 183–196.

[116] M. H. Wilkinson, "A fast component-tree algorithm for high dynamic-range images and second generation connectivity," in *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 1021–1024.

[117] J. Fabrizio and B. Marcotegui, "Fast implementation of the ultimate opening," in *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer, 2009, pp. 272–281.

[118] G. K. Ouzounis and M. H. Wilkinson, "A parallel implementation of the dual-input max-tree algorithm for attribute filtering," in *Proc. 8th Int. Symp. Math. Morphol.(ISMM)*, vol. 1, 2007, pp. 449–460.

[119] U. Moschini, A. Meijster, and M. H. Wilkinson, "A hybrid shared-memory parallel max-tree algorithm for extreme dynamic-range images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 513–526, 2017.

[120] R. E. Tarjan, "Efficiency of a good but not linear set union algorithm," *Journal of the ACM (JACM)*, vol. 22, no. 2, pp. 215–225, 1975.

[121] R. Jones, "Connected filtering and segmentation using component trees," *Computer Vision and Image Understanding*, vol. 75, no. 3, pp. 215–228, 1999.

[122] J. Drapeau, T. Géraud, M. Coustaty, J. Chazalon, J.-C. Burie, V. Eglin, and S. Bres, "Extraction of ancient map contents using trees of connected components," in *International Workshop on Graphics Recognition*. Springer, 2017, pp. 115–130.

[123] V. Caselles and P. Monasse, *Geometric Description of Images as Topographic Maps*, ser. LNM. Springer, 2009, vol. 1984.

[124] Y. Song, "A topdown algorithm for computation of level line trees," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2107–2116, 2007.

[125] S. Crozet and T. Géraud, "A first parallel algorithm to compute the morphological tree of shapes of nD images," in *Proc. of IEEE ICIP*, 2014, pp. 2933–2937.

[126] Y. Xu, T. Géraud, and L. Najman, "Salient level lines selection using the mumford-shah functional," in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 1227–1231.

[127] J. Cardelino, G. Randall, M. Bertalmio, and V. Caselles, "Region based segmentation using the tree of shapes," in *2006 International Conference on Image Processing*. IEEE, 2006, pp. 2421–2424.

[128] G. Cavallaro, M. Dalla Mura, J. A. Benediktsson, and A. Plaza, "Remote sensing image classification using attribute filters defined over the tree of shapes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 7, pp. 3899–3911, 2016.

[129] M. Ô. V. Ngoc, J. Fabrizio, and T. Géraud, "Saliency-based detection of identy documents captured by smartphones," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 2018, pp. 387–392.

[130] E. Carlinet, "A tree of shapes for multivariate images," Ph.D. dissertation, Universite Paris-Est, 2015.

[131] E. Aptoula and S. Lefèvre, "A comparative study on multivariate mathematical morphology," *Pattern Recognition*, vol. 40, no. 11, pp. 2914–2929, 2007.

[132] P. Salembier, A. Oliveras, and L. Garrido, "Motion connected operators for image sequences," in *1996 8th European Signal Processing Conference (EUSIPCO 1996)*. IEEE, 1996, pp. 1–4.

[133] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.

[134] B. Peng, L. Zhang, and D. Zhang, "A survey of graph theoretical approaches to image segmentation," *Pattern Recognition*, vol. 46, no. 3, pp. 1020–1038, 2013.

[135] W. Khan, "Image segmentation techniques: A survey," *Journal of Image and Graphics*, vol. 1, no. 4, pp. 166–170, 2013.

[136] T. Malisiewicz and A. A. Efros, "Improving spatial support for objects via multiple segmentations," 2007.

[137] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[138] I. Endres and D. Hoiem, "Category-independent object proposals with diverse ranking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 2, pp. 222–234, 2013.

[139] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 328–335.

[140] C. Yang *et al.*, "Saliency detection via graph-based manifold ranking," in *Proc. of ICPR*, 2013, pp. 3166–3173.

[141] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proc. of ICPR*, 2014, pp. 2814–2821.

[142] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 1323–1330.

[143] C. W. Chen, J. Luo, and K. J. Parker, "Image segmentation via adaptive k-mean clustering and knowledge-based morphological operations with biomedical applications," *IEEE transactions on image processing*, vol. 7, no. 12, pp. 1673–1683, 1998.

[144] L. G. Roberts, "Machine perception of three-dimensional solids," Ph.D. dissertation, Massachusetts Institute of Technology, 1963.

[145] I. Sobel, "Camera models and machine perception," Computer Science Department, Technion, Tech. Rep., 1972.

[146] J. Canny, "A computational approach to edge detection," in *Readings in computer vision*. Elsevier, 1987, pp. 184–203.

[147] J. M. Prewitt, "Object enhancement and extraction," *Picture processing and Psychopictorics*, vol. 10, no. 1, pp. 15–19, 1970.

[148] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *International journal of computer vision*, vol. 43, no. 1, pp. 7–27, 2001.

[149] P. Arbelaez, "Boundary extraction in natural images using ultrametric contour maps," in *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*. IEEE, 2006, pp. 182–182.

[150] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 5, pp. 530–549, 2004.

[151] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1841–1848.

[152] A. Bieniek and A. Moga, "An efficient watershed algorithm based on connected components," *Pattern recognition*, vol. 33, no. 6, pp. 907–916, 2000.

[153] F. Meyer, "Color image segmentation," in *1992 International Conference on Image Processing and its Applications*. IET, 1992, pp. 303–306.

[154] H. Digabel and C. Lantuéjoul, "Iterative algorithms," in *Proc. 2nd European Symp. Quantitative Analysis of Microstructures in Material Science, Biology and Medicine*, vol. 19, no. 7. Stuttgart, West Germany: Riederer Verlag, 1978, p. 8.

[155] S. Beucher, "Use of watersheds in contour detection," in *Proceedings of the International Workshop on Image Processing*. CCETT, 1979.

[156] J. B. Roerdink and A. Meijster, "The watershed transform: Definitions, algorithms and parallelization strategies," *Fundamenta informaticae*, vol. 41, no. 1, 2, pp. 187–228, 2000.

[157] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 583–598, 1991.

[158] F. Meyer and S. Beucher, "Morphological segmentation," *Journal of visual communication and image representation*, vol. 1, no. 1, pp. 21–46, 1990.

[159] F. Meyer, "Watershed topographic distance and watershed lines," *Signal Processing*, vol. 38, no. 1, pp. 113–125, 1994.

[160] P. Yadollahpour, "Exploring and exploiting diversity for image segmentation," *arXiv preprint arXiv:1709.01625*, 2017.

[161] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Departmental Papers (CIS)*, p. 107, 2000.

[162] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 11, pp. 1101–1113, 1993.

[163] L. R. Ford Jr and D. R. Fulkerson, *Flows in networks*. Princeton university press, 2015.

[164] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 9, pp. 1124–1137, 2004.

[165] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," *Journal of the ACM (JACM)*, vol. 35, no. 4, pp. 921–940, 1988.

[166] A. INCORP, "Adobe photoshop user guide," 2002.

[167] E. N. Mortensen and W. A. Barrett, "Intelligent scissors for image composition," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 191–198.

[168] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski, "A bayesian approach to digital matting," in *CVPR (2)*, 2001, pp. 264–271.

[169] P. Corporation, "Knockout user guide," 2002.

[170] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient nd image segmentation," *International journal of computer vision*, vol. 70, no. 2, pp. 109–131, 2006.

[171] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[172] J. Liang, T. McInerney, and D. Terzopoulos, "Interactive medical image segmentation with united snakes," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 1999, pp. 116–127.

[173] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

[174] G. Friedland, K. Jantz, and R. Rojas, "Siox: Simple interactive object extraction in still images," in *Seventh IEEE International Symposium on Multimedia (ISM'05)*. IEEE, 2005, pp. 7–pp.

[175] B. L. Price, B. Morse, and S. Cohen, "Geodesic graph cut for interactive image segmentation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 3161–3168.

[176] X. Bai and G. Sapiro, "Geodesic matting: A framework for fast interactive image and video segmentation and matting," *International journal of computer vision*, vol. 82, no. 2, pp. 113–132, 2009.

[177] A. Criminisi, T. Sharp, and A. Blake, "Geos: Geodesic image segmentation," in *European Conference on Computer Vision*. Springer, 2008, pp. 99–112.

[178] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, vol. 40, no. 2, pp. 99–121, 2000.

[179] C. Couprie, L. J. Grady, L. Najman, and H. Talbot, "Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest." in *ICCV*, vol. 9, 2009, pp. 731–738.

[180] P. Soille, Ed., *Morphological Image Analysis—Principles and Applications*, 2nd ed. Springer-Verlag, 2004.

[181] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[182] R. W. Floyd, "Algorithm 97: shortest path," *Communications of the ACM*, vol. 5, no. 6, p. 345, 1962.

[183] R. Kimmel and J. A. Sethian, "Optimal algorithm for shape from shading and path planning," *Journal of Mathematical Imaging and Vision*, vol. 14, no. 3, pp. 237–244, 2001.

[184] F. Benmansour and L. D. Cohen, "Fast object segmentation by growing minimal paths from a single point on 2d or 3d images," *Journal of Mathematical Imaging and Vision*, vol. 33, no. 2, pp. 209–221, 2009.

[185] T. Deschamps and L. D. Cohen, "Fast extraction of minimal paths in 3d images and applications to virtual endoscopy," *Medical image analysis*, vol. 5, no. 4, pp. 281–299, 2001.

[186] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3395–3402.

[187] M. R. Abkenar, H. Sadreazami, and M. O. Ahmad, "Graph-based salient object detection using background and foreground connectivity cues," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–5.

[188] Y. Hu, Y. Li, R. Song, P. Rao, and Y. Wang, "Minimum barrier superpixel segmentation," *Image and Vision Computing*, vol. 70, pp. 1–10, 2018.

[189] N. Liu, R. Ju, T. Ren, and G. Wu, "A saliency-guided method for automatic photo refocusing," in *Proceedings of the International Conference on Internet Multimedia Computing and Service*. ACM, 2016, pp. 264–267.

[190] X. Huang, Y. Zheng, J. Huang, and Y.-J. Zhang, "A minimum barrier distance based saliency box for object proposals generation," *IEEE Signal Processing Letters*, 2018.

[191] H. Xiao, J. Feng, G. Lin, Y. Liu, and M. Zhang, "Monet: Deep motion exploitation for video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1140–1148.

[192] J. M. Zhang and Y. X. Shen, "Spectral segmentation via minimum barrier distance," *Multimedia Tools and Applications*, vol. 76, no. 24, pp. 25 713–25 729, 2017.

[193] J.-P. Aubin and H. Frankowska, *Set-valued analysis*. Springer Science & Business Media, 2009.

[194] V. Kovalevsky, "On the topology of discrete spaces," *Studientexte, Digitale Bildverarbeitung*, no. 93/86, 1986.

[195] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 11, pp. 1254–1259, 1998.

[196] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 2007, pp. 1–8.

[197] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE transactions on image processing*, vol. 19, no. 1, pp. 185–198, 2009.

[198] J. Yang and M.-H. Yang, "Top-down visual saliency via joint crf and dictionary learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 3, pp. 576–588, 2016.

[199] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?" in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2. IEEE, 2004, pp. II–II.

[200] J. Han, K. N. Ngan, M. Li, and H.-J. Zhang, "Unsupervised extraction of visual attention objects in color images," *IEEE transactions on circuits and systems for video technology*, vol. 16, no. 1, pp. 141–145, 2005.

[201] F. Stentiford, "Attention based auto image cropping," in *Workshop on Computational Attention and Applications, ICVS*, vol. 1. Citeseer, 2007, pp. 253–261.

[202] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 10, pp. 1915–1926, 2011.

[203] Y. Gao, M. Shi, D. Tao, and C. Xu, "Database saliency for fast image retrieval," *IEEE Transactions on Multimedia*, vol. 17, no. 3, pp. 359–369, 2015.

[204] M.-M. Cheng, N. J. Mitra, X. Huang, and S.-M. Hu, "Salientshape: Group saliency in image collections," *The Visual Computer*, vol. 30, no. 4, pp. 443–453, 2014.

[205] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 185–207, 2012.

[206] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[207] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," in *Matters of intelligence*. Springer, 1987, pp. 115–141.

[208] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk, "Frequency-tuned salient region detection," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, no. CONF, 2009, pp. 1597–1604.

[209] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE TPAMI*, vol. 37, no. 3, pp. 569–582, 2015.

[210] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 733–740.

[211] J. Dolson, J. Baek, C. Plagemann, and S. Thrun, "Upsampling range data in dynamic environments," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 1141–1148.

[212] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE transactions on image processing*, vol. 24, no. 12, pp. 5706–5722, 2015.

[213] P.-H. Tseng, R. Carmi, I. G. Cameron, D. P. Munoz, and L. Itti, "Quantifying center bias of observers in free viewing of dynamic natural scenes," *Journal of vision*, vol. 9, no. 7, pp. 4–4, 2009.

[214] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 2106–2113.

[215] L. Grady, M.-P. Jolly, and A. Seitz, "Segmentation from a box," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 367–374.

[216] V. Gopalakrishnan, Y. Hu, and D. Rajan, "Random walks on graphs to model saliency in images," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1698–1705.

[217] ——, "Random walks on graphs for salient object detection in images," *IEEE Transactions on Image processing*, vol. 19, no. 12, pp. 3232–3242, 2010.

[218] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2083–2090.

[219] Q. Zhang, J. Lin, W. Li, Y. Shi, and G. Cao, "Salient object detection via compactness and objectness cues," *The Visual Computer*, vol. 34, no. 4, pp. 473–489, 2018.

[220] Y. Ji, H. Zhang, K.-K. Tseng, T. W. Chow, and Q. J. Wu, "Graph model-based salient object detection using objectness and multiple saliency cues," *Neurocomputing*, vol. 323, pp. 188–202, 2019.

[221] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 33, no. 2, pp. 353–367, 2010.

[222] P. Khuwuthyakorn, A. Robles-Kelly, and J. Zhou, "Object of interest detection by saliency learning," in *European conference on Computer vision*. Springer, 2010, pp. 636–649.

[223] N. Tong, H. Lu, X. Ruan, and M.-H. Yang, "Salient object detection via bootstrap learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1884–1892.

[224] L. Mai, Y. Niu, and F. Liu, "Saliency aggregation: A data-driven approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1131–1138.

[225] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 73–80.

[226] P. Mehrani and O. Veksler, "Saliency segmentation based on learning and graph cut refinement." in *BMVC*, 2010, pp. 1–12.

[227] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3183–3192.

[228] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1265–1274.

[229] T. Chen, L. Lin, L. Liu, X. Luo, and X. Li, "Disc: Deep image saliency computing via progressive representation learning," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 6, pp. 1135–1149, 2016.

[230] L. Ding and A. Goshtasby, "On the canny edge detector," *Pattern Recognition*, vol. 34, no. 3, pp. 721–725, 2001.

[231] R. Dida, "Use of the hough transformation to detect lines and curves in pictures," *Magazine Communications of the ACM*, vol. 15, no. 1, 1972.

[232] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 4, pp. 722–732, 2008.

[233] K. Bulatov, V. V. Arlazarov, T. Chernov, O. Slavin, and D. Nikolaev, "Smart idreader: Document recognition in video stream," in *Proc. of ICDAR*, vol. 6. IEEE, 2017, pp. 39–44.

[234] A. Minkina, D. Nikolaev, S. Usilin, and V. Kozyrev, "Generalization of the viola-jones method as a decision tree of strong classifiers for real-time object recognition in video stream," in *Seventh International Conference on Machine Vision (ICMV 2014)*, vol. 9445. International Society for Optics and Photonics, 2015, p. 944517.

[235] K. Javed and F. Shafait, "Real-time document localization in natural images by recursive application of a cnn," in *Proc. of ICDAR*, vol. 1. IEEE, 2017, pp. 105–110.

[236] G. Borgefors, "Distance transformations in arbitrary dimensions," *CVGIP*, vol. 27, no. 3, pp. 321–345, 1984.

[237] F. Cao, J.-L. Lisani, J.-M. Morel, P. Musé, and F. Sur, *A Theory of Shape Identification*, ser. LNM. Springer, 2008, vol. 1948.

[238] V. Caselles, B. Coll, and J.-M. Morel, "Topographic maps and local contrast changes in natural images," *International Journal on Computer Vision*, vol. 33, no. 1, pp. 5–27, 1999.

[239] G. Wang, Y. Zhang, and J. Li, "High-level background prior based salient object detection," *Journal of Visual Communication and Image Representation*, vol. 48, pp. 432–441, 2017.

[240] E. Carlinet, S. Crozet, and T. Géraud, "The tree of shapes turned into a max-tree: A simple and efficient linear algorithm," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1488–1492.

[241] L. Zhang, C. Yang, H. Lu, X. Ruan, and M.-H. Yang, "Ranking saliency," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1892–1904, 2017.

[242] J. Shi, Q. Yan, L. Xu, and J. Jia, "Hierarchical image saliency detection on extended cssd," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 717–729, 2016.

[243] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 280–287.

[244] S. Calarasanu, J. Fabrizio, and S. Dubuisson, "Using histogram representation and Earth Mover's Distance as an evaluation tool for text detection," in *Proc. of the Intl. Conf. on Document Analysis and Recognition (ICDAR)*, 2015, pp. 221–225.

[245] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to evaluate foreground maps?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 248–255.

[246] M. Holuša and E. Sojka, "The k-max distance in graphs and images," *Pattern Recognition Letters*, vol. 98, pp. 103–109, 2017.

[247] L. Vincent, "Minimal path algorithms for the robust detection of linear features in gray images," *Computational Imaging and Vision*, vol. 12, pp. 331–338, 1998.

[248] A. M. Mendrik, K. L. Vincken, H. J. Kuijf, M. Breeuwer, W. H. Bouvy, J. De Bresser, A. Alansary, M. De Bruijne, A. Carass, A. El-Baz *et al.*, "Mrbrains challenge: online evaluation framework for brain image segmentation in 3t mri scans," *Computational intelligence and neuroscience*, vol. 2015, p. 1, 2015.

[249] L. Martí-Bonmatí, R. Sopena, P. Bartumeus, and P. Sopena, "Multimodality imaging techniques," *Contrast media & molecular imaging*, vol. 5, no. 4, pp. 180–189, 2010.

[250] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geoscience and remote sensing magazine*, vol. 1, no. 2, pp. 6–36, 2013.

[251] G. Licciardi, F. Pacifici, D. Tuia, S. Prasad, T. West, F. Giacco, C. Thiel, J. Inglada, E. Christophe, J. Chanussot *et al.*, "Decision fusion for the classification of hyperspectral data: Outcome of the 2008 grs-s data fusion contest," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 11, pp. 3857–3865, 2009.

[252] I. Jolliffe, *Principal component analysis*. Springer, 2011.