



UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS



École Doctorale Santé, Sciences, Technologies

Laboratoire d'Informatique (EA 2101)

Equipe Reconnaissance des Formes et Analyse d'Images

THÈSE présentée par :

Julien OLIVIER

soutenue le : 16 septembre 2009

pour obtenir le grade de : Docteur de l'université François - Rabelais

Discipline/ Spécialité : Informatique

**Méthodes d'accélération et approches supervisées pour les
contours actifs.
Applications à la segmentation d'images 2D, 3D et texturées**

BONÉ Romuald	Examineur	Professeur, Ecole Nationale d'Ingénieurs du Val de Loire Blois
CARDOT Hubert	Directeur	Professeur, Université François Rabelais de Tours
CHASSERY Jean-Marc	Rapporteur	Directeur de Recherche CNRS, Université Joseph Fourier de Grenoble
COHEN Laurent	Examineur	Directeur de Recherche CNRS, Université Paris IX Dauphine
DAOUDI Mohamed	Rapporteur	Professeur, Institut Telecom, Telecom Lille 1
ROSENBERGER Christophe	Examineur	Professeur, Université de Caen Basse-Normandie
ROUSSELLE Jean-Jacques	Examineur	Maître de conférences, Université François Rabelais de Tours

A mon grand père, incarnation à mes yeux de l'esprit scientifique
A mon père, qui n'a jamais abandonné son esprit critique
A ma fille, dont l'esprit de découverte ne cesse de m'étonner

Remerciements

Je tiens tout d'abord à exprimer ma gratitude à Messieurs Laurent Cohen et Christophe Rosenberger pour avoir accepté de participer à mon jury. Je remercie également chaleureusement Messieurs Jean-Marc Chassery et Mohamed Daoudi d'avoir pris de leur temps pour rapporter ce manuscrit.

Je tiens également à remercier Monsieur Jean-Charles Billaut pour m'avoir accepté au sein du Laboratoire d'Informatique de Tours.

Je remercie Monsieur Hubert Cardot pour avoir dirigé cette thèse, ainsi que Messieurs Romuald Boné et Jean-Jacques Rousselle pour leur encadrement et leurs conseils.

Je remercie également mes parents pour ne pas avoir pris peur lorsque je me suis engagé dans ces études et m'avoir régulièrement aidé financièrement lorsque j'en avais besoin.

Je remercie Derwella pour m'avoir supporté durant ces années qui se sont avérés parfois difficiles, et Lila, la seule personne capable de vous redonner une motivation digne des premiers jours par un simple éclat de rire.

Enfin, même si il est évident que certaines seront oubliées, je tenais à remercier les personnes suivantes car elles ont toutes contribué de près ou de loin à ces années de doctorat : Steph, Seb, Julien, Mathieu, Alain, les Funky Warriors, les Dirty, Mike Portnoy, Luffy et ma double ...

Résumé

Les contours actifs font partie des méthodes de segmentation les plus répandues. Ils sont utilisés dans de nombreuses applications de reconstruction, de détection ou encore de suivi d'objet aussi bien 2D que 3D. Au delà de la distinction faite entre les modèles basés contour et basés région, on discerne deux familles de représentation des contours actifs. La représentation paramétrique échantillonne la courbe selon son abscisse curviligne de manière à guider la déformation du contour actif à l'aide de points de contrôle. La représentation implicite, appelée également représentation en ensembles de niveaux, détermine la déformation du contour actif en analysant le comportement d'un modèle de dimension supérieure. Le principal avantage de la représentation paramétrique est la rapidité de segmentation alors que la représentation implicite reste la plus efficace en terme de précision, autorisant la courbe à gérer naturellement les changements de topologie. L'objectif de cette thèse est de proposer des améliorations pour chaque famille de représentation.

Basées sur l'algorithme d'évolution gloutonne des modèles paramétriques, trois méthodes d'accélération utilisant une gestion dynamique de la grille de voisinage de chaque point de contrôle ont été développées. La première approche propose d'alterner, lors de chaque itération de l'évolution gloutonne, une étape de déformation du contour actif avec une grille de voisinage classique et une étape utilisant un voisinage linéaire. La deuxième modifie la grille de voisinage de manière à la décaler dans la direction paraissant la plus intéressante. Enfin, la troisième autorise la grille de voisinage à se déformer afin d'augmenter l'espace de recherche de chaque point de contrôle selon certaines directions. Ces trois méthodes d'accélération sont appliquées à la segmentation 2D et 3D et comparées à l'évolution gloutonne classique ainsi qu'à une implémentation implicite rapide récente.

Deux modèles de contours actifs ont également été développés dans le but d'améliorer la précision des approches implicites sur les images texturées. Afin d'être en mesure d'appréhender ce type d'image, les deux modèles proposés utilisent les coefficients d'Haralick pour guider leur évolution. De plus, pour garantir leur indépendance par rapport aux types de textures segmentés, ils sont introduits dans un processus de segmentation supervisé utilisant une image d'apprentissage comprenant une segmentation experte. Le premier contour actif développé, inspiré du modèle basé région développé dans [Chan et Vese, 2001], utilise le principe de la programmation linéaire afin de déterminer le poids optimal de chaque coefficient d'Haralick, proportionnel à sa capacité à séparer correctement les textures de l'image d'apprentissage. Le deuxième modèle utilise l'implémentation en ensembles de niveaux développée dans [Shi et Karl, 2005a] pour introduire un classificateur binaire supervisé directement dans l'équation d'évolution du modèle. Un ensemble d'apprentissage est extrait à partir de la segmentation experte, et utilisé pour réaliser l'algorithme d'apprentissage du classificateur. Ces deux approches sont appliquées à la segmentation d'images texturées de différents types, 2D et 3D, et leur précision est comparée à celle du modèle de Chan et Vese.

Mots clés : Contours actifs, modèles déformables, évolution gloutonne, accélération, ensembles de niveaux, segmentation supervisée, classification supervisée, segmentation de textures, segmentation d'images échographiques, 3D.

Abstract

Active contours are very popular segmentation tools. They are involved in various application fields such as reconstruction, detection or tracking of moving objects, in two or three dimensions. Active contours are usually separated in two groups regarding their energy functional: contour-based models and region-based models. Two other families can be defined according to the representation of the model. The parametric representation samples the curve in order to guide its deformation with control points. The implicit representation, also known as level set representation, determine the deformation of the curve by analyzing the behaviour of a higher dimension model. Main advantage of the parametric representation is its segmentation quickness, whereas models implicitly represented are more precise, due to their ability to handle topological changes automatically. The purpose of the work carried out during this PhD was to propose ameliorations for both of these representation families.

Based on the greedy algorithm for parametric models, three acceleration methods using a dynamical control of the neighbourhood of each control point have been developed. In the first approach, we propose to alternate an evolution step of the active contour using a classical neighbourhood grid and an evolution step using a linear neighbourhood, during each iteration of the greedy algorithm. The second approach is based on a modification of the neighbourhood grid in order to switch it in the direction appearing as the most relevant. At last, the third method allows the neighbourhood to be deformed so that the search space of each control point can be increased in certain directions. These three acceleration methods are applied to 2D and 3D segmentation problems, and compared to the classical greedy algorithm and to a recent fast implicit implementation.

Two active contour models have also been developed in order to improve the precision of implicit approaches on textured images. To handle this type of image, the two proposed models evolve using Haralick texture features. Furthermore, to ensure their independence to the type of segmented texture, they are introduced in a supervised segmentation process using a learning image with an expert segmentation. The first model is inspired by the region-based model developed in [Chan et Vese, 2001] and uses the principle of linear programming to determine the optimal weight of each Haralick coefficient, proportional to its ability to separate textures of the learning image. The second model uses the recent level set implementation developed in [Shi et Karl, 2005a] to introduce a supervised binary classifier in the motion equation of the model. Extracted from the expert segmentation, a learning dataset is used to carry out the learning task of the classifier. These two approaches are applied to different kind of 2D and 3D textured image segmentation problems. Their precision is compared to the Chan and Vese model's.

Keywords : Active contours, deformable models, greedy algorithm, acceleration, level sets, supervised segmentation, supervised classification, texture segmentation, echographic image segmentation, 3D.

Table des matières

Introduction	23
1 Segmentation d'images par contour actif : état de l'art	27
1.1 Forme générale d'un contour actif	28
1.1.1 Définition énergétique du contour	28
1.1.2 Représentation du contour	30
1.2 Modèles de contours actifs	46
1.2.1 Modèles de contours actifs basés sur l'information de contour	47
1.2.2 Modèles de contours actifs basés sur l'information de région	51
1.2.3 Contours actifs basés sur une approche géométrique	59
1.3 Analyse des caractéristiques des contours actifs et améliorations proposées .	61
2 Analyse d'images grâce à l'information de texture	63
2.1 Notion de texture	64
2.2 Les descripteurs de textures	66
2.2.1 Méthodes statistiques	66
2.2.2 Méthodes géométriques	69
2.2.3 Méthodes basées modèle	71
2.2.4 Méthodes empruntées au traitement du signal	77
2.3 Caractériser une texture : dans quel but ?	82
2.3.1 Classification de textures	83
2.3.2 Synthèse de textures	84
2.3.3 Segmentation basée sur la texture	84
2.3.4 Interprétation des formes à partir de l'information de texture (<i>Shape from Texture</i>)	86
2.4 Choisir un descripteur de textures	86
3 Améliorations apportées aux propriétés de segmentation des contours actifs	89

TABLE DES MATIÈRES

3.1	Accélération de contours actifs paramétriques par calcul de voisinages dynamiques	90
3.1.1	Voisinage à anticipation linéaire	90
3.1.2	Voisinage à anticipation par décalage	94
3.1.3	Voisinage à anticipation par déformation	98
3.2	Contours actifs supervisés pour la segmentation d'images texturées par ensembles de niveaux	102
3.2.1	Contour actif supervisé basé sur la programmation linéaire pour la segmentation d'images texturées	103
3.2.2	Contour actif guidé par classificateur binaire supervisé pour la segmentation de textures	112
4	Application des modèles proposés à la segmentation d'images 2D, 3D et texturées	123
4.1	Evaluation de la qualité de segmentation	123
4.2	Segmentation d'images 2D par modèle paramétrique à anticipation par calcul de voisinages dynamiques	126
4.3	Segmentation d'images 3D par modèle paramétrique à anticipation par calcul de voisinages dynamiques	128
4.3.1	Segmentation d'images synthétiques	129
4.3.2	Segmentation d'images naturelles	132
4.4	Choix de la méthode	134
4.5	Segmentation d'images texturées 2D par contour actif supervisé	135
4.5.1	Segmentation d'images synthétiques	135
4.5.2	Segmentation d'images échographiques	145
4.5.3	Segmentation de zones urbaines dans des images satellites	149
4.5.4	Analyse des performances des modèles implicites supervisés	152
4.6	Segmentation d'images échographiques 3D par contour actif supervisé . . .	153
	Conclusion	161
	A Principaux coefficients d'Haralick	165
	B De la fonctionnelle d'énergie d'un contour actif à son équation d'évolution	167
	C Résultats de segmentation des modèles guidés par classificateur sur les images synthétiques	169
	D Résultats de segmentation des modèles guidés par classificateur sur les images échographiques	177

TABLE DES MATIÈRES

E Résultats de segmentation des modèles guidés par classificateur sur les images satellites de la base ASTER	181
---	------------

TABLE DES MATIÈRES

Liste des tableaux

4.1	Evaluation de la vitesse de segmentation des méthodes proposées et de l'évolution gloutonne classique sur les quatre images 2D testées.	127
4.2	Comparaison des temps d'exécution et de la précision de segmentation des quatre méthodes testées sur les trois images synthétiques 3D.	131
4.3	Comparaison des temps d'exécution et de la précision de segmentation des quatre méthodes testées sur une image tomodensitométrie.	133
4.4	Comparaison de la précision de segmentation des quatre méthodes testées sur les deux premières images synthétiques. PRA_m correspond au critère de Pratt modifié, et d_{mga} à la mesure générique d'anomalies. Les mesures de précision sont multipliées par un facteur 100.	137
4.5	Comparaison de la précision de segmentation des quatre méthodes testées sur les deux premières images synthétiques. Les mesures de précision sont multipliées par un facteur 100.	138
4.6	Comparaison de la précision de segmentation des méthodes testées sur la série d'images échographiques. Les mesures de précision sont multipliées par un facteur 100.	147
4.7	Comparaison de la précision de segmentation des méthodes testées sur la série d'images satellites. Les mesures de précision sont multipliées par un facteur 100.	150

LISTE DES TABLEAUX

Table des figures

1.1	Principe d'échantillonnage de la représentation paramétrique d'un contour actif.	31
1.2	Représentation de $\phi(\mathbf{x}, t)$ et de son intersection avec le plan de niveau zéro : la forme en trois dimensions représentée sur l'image de gauche est la fonction d'ensembles de niveaux $\phi(\mathbf{x}, t)$ définie pour tout pixel $\mathbf{x}$ de l'image. Le contour $\mathbf{C}(s, t)$ représenté sur l'image de droite est déterminé par l'intersection de $\phi(\mathbf{x}, t)$ et du plan de niveau zéro.	37
1.3	Initialisation de ϕ . Pour chaque pixel $\mathbf{x}$, $\phi(\mathbf{x}, 0)$ est définie comme la distance signée au contour. L'image de gauche représente la valeur absolue de ϕ lors de l'initialisation pour chaque pixel de l'image (normalisée entre 0 et 255), l'image de droite illustre la forme de cône alors prise par ϕ	38
1.4	Illustration de la gestion des changements de topologie de l'implémentation en ensembles de niveaux : la fonction ϕ se déforme et, bien qu'elle ne change pas de topologie directement, le sous-ensemble observé (son intersection avec le plan de niveau zéro) change de topologie en donnant deux contours distincts	39
1.5	Différentes classes de pixels pour l'algorithme <i>Fast Marching</i> . Les flèches représentent le sens de propagation du front. Les pixels noirs sont les points calculés, les pixels gris les points estimés, et les pixels blancs représentent les points lointains.	42
1.6	Principe de l'implémentation des ensembles de niveaux par bande étroite. Les pixels gris représentent la bande étroite et les pixels noirs le contour actif. a) la mise à jour de ϕ n'est effectuée qu'à l'intérieur de la bande b) le contour est déplacé et se rapproche de l'extrémité de la bande c) la bande étroite est alors réinitialisée.	43
1.7	Différentes valeurs prises par la fonction d'ensembles de niveaux pour l'implémentation rapide de Shi et Karl : les points blancs sont les pixels intérieurs pour lesquels $\phi = -3$, les points bleus représentent les pixels appartenant à la liste L_{in} ($\phi = -1$), les points rouges sont les pixels appartenant à L_{out} ($\phi = 1$), et enfin les points gris sont les pixels extérieurs pour lesquels $\phi = 3$	44
1.8	a) Fonctions de Heaviside classique et modifiée b) Fonctions de Dirac classique et modifiée.	56

TABLE DES FIGURES

2.1	Illustration des limites de l'analyse d'images basée uniquement sur l'étude des niveaux de gris : les images a) et b) ont des statistiques d'intensité de niveaux de gris presque identiques (moyenne, écart type et histogramme), mais sont visuellement très différentes. Une analyse de plus haut niveau telle que l'analyse de textures est donc nécessaire	64
2.2	Illustration des trois principaux types de textures	65
2.3	Mesure d'autocorrélation pour trois types de textures. a) Texture fine : la fonction décroît rapidement depuis le centre. b) Texture grossière : la fonction décroît lentement. c) Texture régulière : la fonction devient périodique et fait apparaître une forme de type "pics et vallées".	68
2.4	Illustration du principe de la caractérisation de textures par modèle binaire local pour un voisinage carré de largeur 3. a) Le pixel à caractériser (valeur de 97) est au centre de son voisinage. b) Chaque voisin de valeur supérieure à celle du pixel traité prend la valeur 1, les autres prennent la valeur 0. c) Répartition de la pondération par une puissance 2 de chaque voisin.	69
2.5	Trois premiers ordres de voisinage pour un site $s \in S$	71
2.6	Illustration de l'influence des différents paramètres du filtre de Gabor sur le domaine fréquentiel observé.	79
2.7	Décomposition en ondelettes discrète d'un signal 1D : chaque étape j de la décomposition fournit un signal d'approximation $a_j[n]$ et un signal de détail $d_j[n]$, chacun de résolution deux fois inférieure au signal d'approximation précédent (" $\downarrow 2$ " exprime la division de la résolution par deux).	81
2.8	Principe de la décomposition en ondelettes discrète en deux dimensions. Les filtres H et G sont convolués une première fois dans le sens des lignes. Leurs sorties sont alors convoluées une nouvelle fois avec les deux filtres dans le sens des colonnes. Une étape de décomposition donnera donc trois images de détails et une image d'approximation.	82
2.9	Trois premières étapes de la décomposition en ondelettes discrète d'une image A_0 . Chaque étape détermine quatre sous-images de résolution deux fois inférieure : trois images de détails (D^1 , D^2 et D^3) et une image d'approximation (A). Cette dernière est alors utilisée comme base de la décomposition suivante.	83
3.1	Illustration de la création du voisinage linéaire. a) Itération classique de l'évolution gloutonne pour un point $\mathbf{v}_i^{(t)}$. La nouvelle position choisie pour $\mathbf{v}_i^{(t)}$ est $\mathbf{v}'_i^{(t)}$. Le déplacement effectué étant $(2, 1)$, le vecteur unitaire $\mathbf{d}_i^{(t)}$ défini par l'équation (3.2) vaut $(0, 9; 0, 45)$. b) Le voisinage linéaire est créé selon l'équation (3.3) $\forall k \in [1, 6]$, donnant l'ensemble des coordonnées des pixels dans le voisinage linéaire relativement à $\mathbf{v}'_i^{(t)}$: $\{(0; 0), (1; 0), (2; 1), (3; 1), (4; 2), (5; 2)\}$	92
3.2	Illustration d'une itération complète de l'évolution gloutonne avec anticipation par voisinage linéaire.	93

3.3	Illustration d'un effet non désiré de l'utilisation du voisinage à anticipation linéaire. a) Le pixel à l'extrémité droite de la courbe représente le point de contrôle $\mathbf{v}_i^{(t)}$ actuellement en cours d'étude. b) Premier déplacement de $\mathbf{v}_i^{(t)}$ vers $\mathbf{v}_i'^{(t)}$. c) Le voisinage linéaire $N_{L_i}^{(t)}$ déterminé est trop restrictif et guide le contour à coté de l'objet à segmenter. Celui-ci sera atteint quelques itérations plus tard, mais cela influencera la rapidité de segmentation du modèle.	94
3.4	Différentes valeurs prises par $\mathbf{d}_i^{(t)}$. Sur chaque imagerie, la figure de gauche correspond aux différents déplacements $\mathbf{v}_i^{(t)} \rightarrow \mathbf{v}_i^{(t+1)}$ (représentés par les pixels gris) pouvant être réalisés lors d'une itération de l'évolution gloutonne avec un voisinage classique. La figure de droite représente le déplacement $\mathbf{d}_i^{(t)}$ approximé.	96
3.5	Illustration de l'utilisation du voisinage à anticipation par décalage pour un point de contrôle. a-b-c) Déroulement de l'itération classique de l'évolution gloutonne. d) Le déplacement du point de contrôle entraîne le décalage de la grille de voisinage dans la direction suivie. Ainsi, lors de la prochaine itération, un plus grand nombre de pixels sera exploré dans la direction semblant la plus intéressante.	97
3.6	Illustration de la méthode d'anticipation par déformation pour deux points de contrôle.	101
3.7	Image d'apprentissage pour une série d'images échographiques. a) Image d'origine b) Définition de C^{in} , la région intérieure de l'objet. c) Définition de C^{out} , la région extérieure de l'objet (région frontière). d) Les coefficients d'Haralick des pixels définissant C^{in} (en blanc) et C^{out} (en gris) sont déterminés afin de composer l'ensemble d'apprentissage $\mathbf{X}$	109
3.8	Processus complet de segmentation : les coefficients d'Haralick $\overline{k_j^{in}}$ et $\overline{k_j^{out}}$ sont extraits des régions C^{in} et C^{out} de l'image d'apprentissage puis réinjectés dans le modèle de programmation linéaire. Celui-ci détermine alors les poids optimaux w_j et les fournit au modèle de contour actif qui sera utilisé pour segmenter toute la série d'images.	112
3.9	Modèle de perceptron multicouche utilisé dans notre approche. La première couche est composée de 13 neurones (12 neurones propageant chacun sans le modifier un coefficient d'Haralick de l'individu traité et un neurone de biais). La deuxième couche comporte n neurones cachés (hn_i). La couche de sortie se compose de 2 neurones (on_i). Le neurone de sortie possédant le plus fort potentiel détermine alors la classe d'appartenance de l'exemple. $hw_{i,j}$ représente le poids synaptique reliant le $j^{ème}$ neurone de la couche d'entrée au $i^{ème}$ neurone de la couche cachée et $ow_{i,j}$ celui reliant le $j^{ème}$ neurone de la couche cachée au $i^{ème}$ neurone de la couche de sortie.	119
3.10	Processus complet de segmentation : dans un premier temps les ensembles $\mathbf{X}_l$ et $\mathbf{X}_e$ extraits de l'image d'apprentissage sont utilisés afin de réaliser l'algorithme d'apprentissage du classificateur. Dans un deuxième temps, celui-ci est utilisé pour guider le contour actif sur les autres images de la série. . . .	120

TABLE DES FIGURES

4.1	Images utilisées pour comparer les performances des méthodes proposées : a) Coquillage cassé b) Image aux rayons X du condyle de la jonction tibio-fémorale c) Image aux rayons X de la tête humérale d'une épaule humaine d) Image synthétique de losange.	126
4.2	Illustration des résultats de segmentation obtenues par la méthode la plus rapide sur chaque image. a-c) EVADec. d) EVAL.	128
4.3	Voisinage cubique utilisé autour de chaque point de contrôle par l'évolution gloutonne 3D.	129
4.4	Principe du remaillage : a) Maillage initial. b) Ajout du sommet $\mathbf{v}_{n+1}$. c) Suppression du sommet $\mathbf{v}_j$	129
4.5	Représentation d'une coupe de chaque image synthétique utilisée : a) Spirale. b) Vase. c) Ellipsoïdes.	130
4.6	Segmentations obtenues par l'EVAL sur les trois images synthétiques.	132
4.7	Segmentation de l'aorte obtenue par l'évolution gloutonne sur une image tomomodensitométrique.	134
4.8	Illustration des quatre séries d'images synthétiques créées (une série par colonne). La première ligne représente les images d'apprentissage, la ligne du milieu les images de test possédant un bruit nul et la ligne du bas les images de test possédant le bruit maximal.	136
4.9	Résultats de segmentation sur la première série d'images. a) $\alpha = 0$. b) $\alpha = 3$. c) $\alpha = 5$. La première ligne illustre les vérités terrain, la deuxième les segmentations du modèle de Chan-Vese et la troisième celles du modèle guidé par réseau de neurones.	139
4.10	Résultats de segmentation sur la deuxième série d'images synthétiques.	140
4.11	Résultats de segmentation sur la troisième série d'images synthétiques.	141
4.12	Résultats de segmentation sur la quatrième série d'images synthétiques.	142
4.13	Comparaison des valeurs moyennes du critère de Pratt modifié en fonction du niveau de bruit introduit dans l'image pour chaque modèle.	143
4.14	Comparaison des valeurs moyennes de la mesure générique d'anomalies en fonction du niveau de bruit introduit dans l'image pour chaque modèle.	143
4.15	Illustration de la faible dépendance à l'initialisation des contours actifs guidés par classificateur. a) CA-RNA. b) Modèle de Chan et Vese. Seuls quelques pixels appartenant à l'objet recherché doivent être situés à l'intérieur de la courbe initiale pour assurer une segmentation correcte du modèle guidé par classificateur. En revanche, le modèle de Chan et Vese demeure très sensible à l'initialisation car elle détermine les intensités recherchées.	144
4.16	Image d'apprentissage pour la série d'images échographiques. a) Image d'origine. b) Régions C^{in} (blanc) et C^{out} (gris) créées pour réaliser l'ensemble d'apprentissage.	145
4.17	Images échographiques utilisées pour comparer les différents modèles.	146
4.18	Résultats de segmentation sur la première image échographique. a) Vérité terrain. b) Modèle de Chan et Vese. c) CA-PLS.	147

TABLE DES FIGURES

4.19	Résultats de segmentation sur la deuxième image échographique.	148
4.20	Résultats de segmentation sur la troisième image échographique.	148
4.21	Résultats de segmentation sur la quatrième image échographique.	148
4.22	Image d'apprentissage pour la série d'images satellites. a) Image d'origine. b) Régions C^{in} (blanc) et C^{out} (gris) créées pour réaliser l'ensemble d'apprentissage.	149
4.23	Images satellites utilisées pour tester les différents modèles.	149
4.24	Résultats de segmentation sur la première image satellite. a) Vérité terrain. b) Modèle de Chan et Vese. c) CA-RNA.	150
4.25	Résultats de segmentation sur la deuxième image satellite.	151
4.26	Résultats de segmentation sur la troisième image satellite.	151
4.27	Résultats de segmentation sur la quatrième image satellite.	151
4.28	Illustration de la succession de coupes composant une image échographique 3D.	153
4.29	Illustration de la création de l'ensemble d'apprentissage sur une coupe de l'échographie 3D. a) Coupe initiale. b) Identification manuelle des régions C^{in} et C^{out} par leurs frontières (bleue pour C^{in} et rouge pour C^{out}). c) Calcul des coefficients d'Haralick 3D et extraction de l'ensemble d'apprentissage.	154
4.30	Illustration de la première échographie 3D à traiter. Le point d'observation reste fixe et les différents axes de coupes sont affichés sur chaque image. La pathologie à segmenter est représentée par la zone sombre au centre.	155
4.31	Segmentation réalisée sur la première échographie 3D.	156
4.32	Maillage reconstruit pour la première échographie 3D.	156
4.33	Deuxième échographie 3D.	157
4.34	Segmentation réalisée sur la deuxième échographie 3D.	157
4.35	Maillage reconstruit pour la deuxième échographie 3D.	158
4.36	Troisième échographie 3D.	158
4.37	Segmentation réalisée sur la troisième échographie 3D.	159
4.38	Maillage reconstruit pour la troisième échographie 3D.	159
C.1	Première série d'image, niveau de bruit 0 : a) Modèle de Chan et Vese. b) Modèle guidé par KPPV. c) Modèle guidé par SVM. d) Modèle guidé par réseau de neurones.	169
C.2	Première série d'images, niveau de bruit 1.	169
C.3	Première série d'images, niveau de bruit 2.	170
C.4	Première série d'images, niveau de bruit 3.	170
C.5	Première série d'images, niveau de bruit 4.	170
C.6	Première série d'images, niveau de bruit 5.	170
C.7	Deuxième série d'images, niveau de bruit 0.	171
C.8	Deuxième série d'images, niveau de bruit 1.	171

TABLE DES FIGURES

C.9 Deuxième série d'images, niveau de bruit 2.	171
C.10 Deuxième série d'images, niveau de bruit 3.	171
C.11 Deuxième série d'images, niveau de bruit 4.	172
C.12 Deuxième série d'images, niveau de bruit 5.	172
C.13 Troisième série d'images, niveau de bruit 0.	172
C.14 Troisième série d'images, niveau de bruit 1.	172
C.15 Troisième série d'images, niveau de bruit 2.	173
C.16 Troisième série d'images, niveau de bruit 3.	173
C.17 Troisième série d'images, niveau de bruit 4.	173
C.18 Troisième série d'images, niveau de bruit 5.	173
C.19 Quatrième série d'images, niveau de bruit 0.	174
C.20 Quatrième série d'images, niveau de bruit 1.	174
C.21 Quatrième série d'images, niveau de bruit 2.	174
C.22 Quatrième série d'images, niveau de bruit 3.	175
C.23 Quatrième série d'images, niveau de bruit 4.	175
C.24 Quatrième série d'images, niveau de bruit 5.	175
D.1 Résultats de segmentation sur la première image échographique : a) Modèle de Chan et Vese. b) contour actif supervisé par programmation linéaire. c) Modèle guidé par KPPV. d) Modèle guidé par SVM. e) Modèle guidé par réseau de neurones.	177
D.2 Résultats de segmentation sur la deuxième image échographique.	178
D.3 Résultats de segmentation sur la troisième image échographique.	178
D.4 Résultats de segmentation sur la quatrième image échographique.	179
E.1 Première image satellite : a) Modèle de Chan et Vese. b) Modèle guidé par KPPV. c) Modèle guidé par SVM. d) Modèle guidé par réseau de neurones.	181
E.2 Deuxième image satellite.	182
E.3 Troisième image satellite.	183
E.4 Quatrième image satellite.	183

Introduction

La vision par ordinateur représente un domaine très actif des sciences de l'informatique. Les principales raisons de cet engouement, en plus de l'intérêt applicatif évident, sont la grande difficulté que représente le développement d'approches d'analyse d'images ainsi que l'augmentation importante du nombre d'images numériques générées. Il est de prime abord naturel de s'étonner de la complexité de l'analyse d'images car l'être humain utilise le processus de vision de manière instinctive, sans se rendre compte de la subtilité des éléments mis en place.

Pour expliquer ce phénomène, il suffit de s'appuyer sur un exemple pour se poser une simple question : quel processus de réflexion permet à une personne d'identifier un nuage dans une image ? En d'autres termes, de quelle manière un être humain part-il d'un objet quelconque extrait d'une image pour finalement être capable de l'identifier comme étant un nuage ? Instinctivement, une première réponse peut être formulée : la personne identifie plusieurs caractéristiques propres à l'objet, telles que sa couleur ou encore sa forme. Déjà, une première opération complexe apparaît car il est nécessaire de détecter l'objet avec précision avant d'être en mesure de le caractériser. Si la vision humaine sépare naturellement les objets, un ordinateur ne peut se fier qu'à une analyse des caractéristiques de l'image, basées sur la valeur des pixels, afin de déterminer les différents objets. Il pourra par exemple s'appuyer sur les changements de couleurs ou encore chercher à déterminer des zones possédant des caractéristiques homogènes. De plus, le système de vision humain utilise en parallèle une notion de connaissance, sur laquelle nous reviendrons par la suite.

En réalité les caractéristiques précédemment citées ne sont pas suffisantes pour identifier un nuage car elles seront exactement les mêmes pour une boule de coton par exemple. La taille de l'objet peut alors être prise en compte, mais cette notion s'avère elle aussi complexe et délicate à estimer car elle est déterminée relativement aux autres objets identifiés dans l'image (par exemple, on saura que le nuage est grand car la voiture qui est à côté paraît petite).

Cette première réflexion nous conduit déjà à appréhender deux concepts complexes : la détection d'un objet et sa caractérisation. Ces deux notions représentent deux domaines majeurs pour la communauté de vision par ordinateur.

Malheureusement, même si la détection et la caractérisation s'avèrent parfaites, il reste toujours difficile de différencier un nuage classique d'une avalanche par exemple. Une notion de connaissance entre alors en jeu. Celle-ci peut être interprétée comme suit : un humain

reconnait un nuage car il en a observé des milliers et qu'il "sait" à quoi cela ressemble. En d'autres termes, à force d'observer un objet, l'humain intègre automatiquement et de manière intuitive énormément de caractéristiques qu'il est presque impossible d'identifier. C'est uniquement lorsqu'il perçoit un objet possédant des caractéristiques très proches que son cerveau lui permet de faire le lien avec un objet de sa base de connaissances, rendant alors l'identification possible.

Plusieurs modèles ont été développés de manière à simuler ce comportement d'apprentissage intuitif. Ceux-ci sont appelés classificateurs. Les réseaux de neurones artificiels, par exemple, sont capables d'apprendre automatiquement sur des données connues afin de déterminer des règles internes permettant de prendre une décision lorsque des données similaires leur sont présentées. Cette deuxième série de concepts est identifiée dans la communauté de la recherche en informatique comme le domaine de la reconnaissance des formes.

Cette notion de connaissance est également utilisée en amont dans les étapes de vision précédemment citées. En effet, comme nous l'avons précisé, la simple séparation des objets est déjà influencée par les connaissances : l'homme est capable d'isoler un nuage sur un ciel bleu car les couleurs des deux objets sont différentes, mais il est également capable d'isoler ce même nuage lorsqu'il est devant un montagne enneigée. Une explication de ce phénomène est que l'homme a appris à de nombreuses reprises la forme habituelle des nuages, et il est désormais capable de les isoler même si leurs arrières-plans sont très similaires au niveau de sa perception visuelle. Ainsi, l'humain ne s'appuie pas uniquement sur ce qu'il perçoit mais également sur ce qu'il a appris, ce qui lui permet d'améliorer ses capacités d'analyse et d'interprétation.

En observant le nombre de notions intervenant dans l'identification d'un unique objet, créer un programme permettant l'identification de tous les objets d'une image apparaît comme un défi intellectuel très fort car il faut additionner un nombre important de concepts : la séparation des différents objets, la gestion des occlusions (reconnaitre un objet alors que celui-ci est partiellement caché), l'utilisation de connaissances, etc. De plus, cela ne constitue qu'une première étape d'un processus d'interprétation de l'image où l'on cherchera à comprendre la scène observée.

Le nombre imposant de concepts mis en œuvre par le système de vision et de reconnaissance humain explique la séparation qu'il existe entre les différents domaines de recherche traitant de cette problématique. Si l'on se fixe comme objectif le développement d'un système de vision automatique, il est alors nécessaire d'optimiser chaque "brique" du processus afin d'être en mesure de fournir les meilleurs résultats à la brique suivante.

Cette thèse se place dans le domaine général de la séparation des différents objets d'une image (appelée plus fréquemment segmentation), et plus particulièrement au niveau d'une famille de méthodes appelées les contours actifs. Leur rôle est de détecter les frontières d'un objet afin de pouvoir l'extraire le plus précisément et rapidement possible. Plusieurs modèles de contours actifs ont été développés pendant ce travail de thèse, chacun permettant d'améliorer des capacités de segmentation telles que la rapidité ou la précision. La première série de modèles, basée sur la représentation paramétrique, améliore la rapidité

de segmentation des contours actifs en utilisant une gestion dynamique du voisinage de chaque point du contour. La deuxième série améliore la précision des modèles représentés à l'aide des ensembles de niveaux sur les images texturées en introduisant une notion de connaissance à l'aide d'un classificateur supervisé utilisant des caractéristiques de textures.

Dans le premier chapitre, nous allons nous intéresser à la forme générale d'un contour actif et aux principaux modèles ayant eu une influence significative sur ce domaine de la segmentation d'images. Nous verrons qu'un contour actif est constitué d'une énergie, définissant ses propriétés géométriques et relatives à l'image, et d'un mode de représentation influençant sa vitesse et sa précision de segmentation. Les différents modèles détaillés ensuite seront classés en trois catégories : modèles basés contour, modèles basés région et modèles géométriques.

Le deuxième chapitre traitera des principales caractéristiques de textures utilisées en analyse d'images. Nous les classerons en quatre familles : les méthodes statistiques, les méthodes géométriques, les méthodes basées modèle et enfin les méthodes empruntées au traitement du signal. Nous verrons également dans quels types d'applications les caractéristiques de textures sont utilisées, et de quelle manière.

Dans le troisième chapitre, nos approches originales vont être présentées. La première partie concernera les trois techniques d'accélération des contours actifs représentés de manière paramétrique. Celles-ci sont basées sur un calcul de voisinages dynamiques lors de l'évolution gloutonne du modèle. La première méthode utilise un voisinage linéaire suivant la direction privilégiée par l'évolution gloutonne, la deuxième décale la grille de voisinage de chaque point dans cette même direction et la troisième autorise la grille à se déformer de manière à augmenter l'espace de recherche du contour actif en ajoutant des pixels susceptibles d'être intéressants.

Dans la deuxième partie du troisième chapitre, les deux modèles supervisés représentés avec les ensembles de niveaux seront présentés. Leur but est d'améliorer la précision des contours actifs sur les images texturées. Pour cela, l'image est analysée à l'aide des caractéristiques de textures d'Haralick et une notion de connaissance est alors introduite dans le processus de segmentation, par le biais d'une segmentation experte réalisée dans une image d'apprentissage. Le premier modèle, un contour actif basé région inspiré du contour actif de Chan et Vese, utilise cette segmentation pour déterminer le poids optimal de chaque coefficient d'Haralick à l'aide du principe de la programmation linéaire (le poids de chaque coefficient d'Haralick sera proportionnel à sa capacité à séparer correctement l'objet et l'arrière plan dans l'image d'apprentissage). Le deuxième modèle introduit directement un classificateur binaire supervisé, appris grâce à l'image d'apprentissage, dans l'équation d'évolution d'un contour actif en utilisant l'implémentation récente en ensembles de niveaux de Shi et Karl. Le classificateur est ensuite utilisé pour guider la déformation de la courbe.

Le quatrième chapitre présentera les différentes applications de nos modèles et comparera leurs performances à celles d'approches connues. Les méthodes d'accélération sont

appliquées à la segmentation d’objets synthétiques et naturels, 2D et 3D, et évaluées tant au niveau de leur vitesse que de leur précision. Les modèles supervisés sont appliqués à la segmentation d’images texturées synthétiques, d’images satellites et d’images échographiques 2D et 3D. La précision des modèles est évaluée par plusieurs critères.

Nous finirons par une conclusion dans laquelle nous réaliserons un bilan des différentes approches développées durant cette thèse et nous présenterons plusieurs pistes envisagées pour augmenter encore leur efficacité.

S’il le souhaite, le lecteur pourra trouver à la fin de ce manuscrit plusieurs annexes qui, si elles ne sont pas impératives à la compréhension des travaux présentés ici, pourront fournir plus de détails sur certaines notions, et des résultats additionnels.

Chapitre 1

Segmentation d'images par contour actif : état de l'art

Depuis leur apparition dans [Kass *et al.*, 1988], les contours actifs sont devenus une des méthodes de segmentation les plus populaires. Leur capacité de segmentation est utilisée dans de nombreux domaines tels que le suivi d'objet [Paragios et Deriche, 2000, Yagi *et al.*, 2000, Chen *et al.*, 2008], la segmentation d'images médicales [Bossart *et al.*, 1996, Schupp *et al.*, 2000, Iakovidis *et al.*, 2006, Alemán-Flores *et al.*, 2007, Han *et al.*, 2007] ou encore la reconstruction 3D [Chung et Ho, 1996, Safont et Marroquin, 1999, Wan et Hou, 2006]. En outre, indépendamment du domaine d'application, la théorie même des contours actifs a donné lieu à de nombreuses contributions, tant à ses débuts que récemment [Osher et Sethian, 1988, Cohen, 1991, Cohen et Cohen, 1993, McInerney et Terzopoulos, 1995, Chan et Vese, 2001, Sundaramoorthi *et al.*, 2007].

Une conséquence directe de cet engouement pour la segmentation par contour actif est la grande diversité de modèles présents dans la littérature. Ainsi, même si ces nombreuses approches permettent d'appréhender la problématique de la segmentation sous des angles variés, identifier les particularités de chaque modèle peut en contrepartie s'avérer délicat : certains, par exemple, privilégieront la vitesse d'exécution alors que d'autres mettront en avant une grande précision de segmentation. D'autres encore seront rapides et efficaces sur des images possédant des zones relativement homogènes mais éprouveront plus de difficultés sur des images bruitées ou texturées.

Dans ce cadre, il s'avère difficile au premier abord d'identifier le modèle de contour actif le mieux adapté pour un type d'application donné. Aussi dans cet état de l'art, un schéma général permettant d'identifier chaque type de contour actif va être dégagé et les principaux modèles présents dans la littérature vont être présentés.

Cet état de l'art des contours actifs va être constitué de deux parties distinctes. Dans un premier temps, nous allons tenter de définir la manière dont un contour actif est construit en restant dans un cadre général, sans nous appuyer sur des modèles précis. Nous verrons qu'un modèle de contour actif se divise en deux principales composantes : une énergie et un mode de représentation. Nous détaillerons alors les propriétés de chaque type de contour et tenterons d'en extraire les avantages et les inconvénients, essayant ainsi d'identifier les

familles d'applications sur lesquelles il peut être considéré efficace.

Dans la deuxième partie de cet état de l'art, nous présenterons les principaux modèles de contours actifs ayant influencé ce domaine de la segmentation d'images. Nous verrons alors que chaque modèle est défini selon le schéma exposé dans la première partie de cet état de l'art.

1.1 Forme générale d'un contour actif

Avant de nous intéresser aux principaux modèles de contours actifs présents dans la littérature nous allons tout d'abord identifier les différents éléments permettant de définir un modèle de contour actif.

1.1.0.1 Préambule

Un contour actif, ou *snake*, est une courbe paramétrée $\mathbf{C}(s)$ d'extrémités a et b , définie dans une image Ω_I et associant au paramètre s un pixel $\mathbf{x} = (x, y)$ tel que :

$$\begin{aligned} \mathbf{C} : [a, b] \in \mathbb{R} &\rightarrow \mathbb{R}^2 \\ s &\rightarrow \mathbf{C}(s) = \mathbf{x}(s) = \begin{bmatrix} x(s) \\ y(s) \end{bmatrix}, \forall s \in \Omega_I \end{aligned} \quad (1.1)$$

Un contour actif peut être ouvert ($a \neq b$) ou fermé ($a = b$).

La courbe $\mathbf{C}(s)$ est initialisée dans l'image Ω_I et évolue sous certaines contraintes selon ses directions normale et tangentielle jusqu'à ce qu'elle s'arrête sur les bords de l'objet à segmenter. Le contour est dit actif car il évolue au cours du temps, de son initialisation jusqu'à son état final. Dans ce cadre, $\mathbf{C}(s, t)$ représente alors la famille de courbes obtenues durant toute l'évolution du contour.

Même s'il existe de nombreux modèles de contours actifs, ceux-ci suivent tous la même forme générale. Ils sont définis par une fonctionnelle d'énergie associée à la courbe et un mode de représentation. La fonctionnelle d'énergie définira le comportement géométrique du modèle ainsi que les données de l'image dont il se servira pour évoluer. La représentation déterminera la manière dont le contour sera implémenté et influencera son comportement durant l'évolution. Alors qu'il existe presque autant de fonctionnelles d'énergie que de modèles de contours actifs, seules deux implémentations différentes sont principalement utilisées : l'implémentation paramétrique et l'implémentation implicite.

Dans cette partie de l'état de l'art des contours actifs nous allons, dans un premier temps, nous intéresser à la définition de l'énergie d'un contour en tentant d'analyser son influence sur le modèle. Dans un deuxième temps nous détaillerons les deux principaux modes de représentation des contours actifs.

1.1.1 Définition énergétique du contour

Afin de faire évoluer le contour actif, une fonctionnelle d'énergie lui est associée. Cette énergie doit être définie de manière à ce que la segmentation idéale corresponde à un

1.1. FORME GÉNÉRALE D'UN CONTOUR ACTIF

contour d'énergie minimale. Le principe de la segmentation par contour actif va donc être de minimiser cette énergie afin que la courbe évolue vers la meilleure segmentation accessible.

L'énergie d'un contour actif est composée de critères appartenant à deux principales familles.

1. Les critères énergétiques intrinsèques : ils prennent en compte les contraintes géométriques et structurantes sur $\mathbf{C}$ et vont permettre de contrôler la géométrie du contour. Ils sont généralement définis comme des énergies déterminées par la courbure de $\mathbf{C}$, son périmètre ou encore son aire intérieure.
2. Les critères extrinsèques : appelés également critères d'attache aux données, ils représentent les contraintes d'adéquation du modèle à l'image et vont permettre de guider la déformation du contour vers les zones d'intérêt puis de la stopper en prenant en compte les caractéristiques de l'image. Nous reviendrons par la suite sur cette partie de la fonctionnelle car c'est elle qui va définir les deux grandes familles de contours actifs : les contours actifs basés contour et basés région.

La forme générale de la fonctionnelle d'énergie associée à un contour actif est donnée par :

$$E(\mathbf{C}(s, t)) = \alpha \int_a^b E_{int} ds + \beta \int E_{ext} ds . \quad (1.2)$$

L'énergie interne E_{int} est déterminée par les critères géométriques intrinsèques de $\mathbf{C}(s, t)$ et l'énergie externe E_{ext} par les critères extrinsèques d'attache aux données. α et β sont utilisés pour pondérer l'influence de chaque type de critère. Nous reviendrons dans quelques paragraphes sur les raisons de l'absence de bornes sur la deuxième intégrale, lorsque nous détaillerons les deux grandes familles de fonctionnelles.

La fonctionnelle d'énergie $E(\mathbf{C}(s, t))$ peut donc être considérée comme un indicateur de la précision de la segmentation réalisée par le contour actif. Idéalement, $E(\mathbf{C}(s, t))$ doit être minimale lorsque $\mathbf{C}(s, t)$ est placée sur les bords de l'objet que l'on cherche à segmenter. Le contour actif va donc évoluer par minimisation successive de $E(\mathbf{C}(s, t))$ jusqu'à atteindre un état stable représenté par une courbe dont l'énergie sera un minimum, généralement local, de $E(\mathbf{C}(s, t))$. Chaque critère de la fonctionnelle doit donc être pensé de manière à faire tendre la fonctionnelle vers zéro lorsque le contour est placé sur les frontières de l'objet recherché.

Le choix de la fonctionnelle d'énergie associée au contour détermine en grande partie les capacités de segmentation du modèle. Il est donc indispensable qu'elle modélise au mieux le comportement souhaité du contour actif. Deux principales familles de fonctionnelles peuvent être définies : les fonctionnelles basées contour et celles basées région. Cette distinction sera faite en fonction du terme d'attache aux données.

Dans le cas des contours actifs basés contour, seule l'information présente le long de $\mathbf{C}$ est prise en compte pour lors de l'évolution. Il s'agit généralement de critères basés sur le gradient d'intensité des pixels de l'image (l'inverse de la norme du gradient, par exemple, permet de faire tendre la fonctionnelle vers zéro sur les fortes différences d'intensités de l'image). Cette mesure est donc intégrée uniquement le long de la courbe. Les contours

actifs basés contour sont efficaces mais ils souffrent d'une grande sensibilité au bruit et leur utilisation reste limitée aux images où les frontières entre les différents objets peuvent être définies par de simples gradients. Ainsi, leur évolution est plus délicate dans des images complexes telles que les images texturées.

Les contours actifs basés région utilisent une information plus globale sur l'image sous la forme de critères appelés descripteurs de régions. Ces derniers sont le plus souvent déterminés par les caractéristiques de la région formée par $\mathbf{C}$ et peuvent être de différentes formes (moyenne ou écart type des intensités des pixels, histogramme des intensités, descripteurs de textures...). Une fois ajouté à la fonctionnelle d'énergie, un critère d'attache aux données basé région ne sera pas intégré le long de la courbe mais sur l'aire de la région située à l'intérieur de celle-ci [Zhu et Yuille, 1996, Paragios et Deriche, 2002], et même parfois sur l'image toute entière [Chan *et al.*, 2000, Chan et Vese, 2001]. Les contours actifs basés région possèdent donc une meilleure capacité de segmentation, mais celle-ci reste fortement dépendante du descripteur de régions choisi pour guider la courbe.

Nous présenterons, dans la section 1.2, les principales fonctionnelles d'énergie basées contour et basées région connues et détaillerons leurs particularités.

1.1.2 Représentation du contour

Le mode de représentation d'un contour actif va déterminer son implémentation et influencera la manière dont il va évoluer. Nous allons détailler les deux grandes familles de représentation : les représentations paramétrique et implicite.

1.1.2.1 Représentation paramétrique

Le mode de représentation paramétrique, appelé également explicite, est le premier apparu dans la littérature. Il fut introduit dans [Kass *et al.*, 1988] pour le premier modèle de contour actif.

La définition de la courbe $\mathbf{C}$ dans l'équation (1.1) correspond à une courbe continue dans le plan. Une image étant une grille discrète et régulière de pixels, $\mathbf{C}$ va être discrétisée et sous-échantillonnée selon son paramètre s (abscisse curviligne). Le contour actif sera ainsi représenté par des points de contrôle reliés entre eux par une notion de voisinage deux à deux. Les déformations successives devant être appliquées au contour seront déterminées uniquement pour les points de contrôle et l'évolution complète du contour actif suivra alors celle des points de contrôle. La figure (1.1) illustre les étapes de construction de la représentation paramétrique discrète d'un contour actif.

Représenter un contour de manière paramétrique présente plusieurs avantages :

- les différentes opérations à réaliser sur l'ensemble du contour (calcul de déplacements, de caractéristiques...) ne sont réalisées que sur les points de contrôle, ce qui entraîne un gain de temps d'exécution important ;
- les composantes intrinsèques telles que la courbure, la normale ou la tangente restent utilisables, car calculables grâce à la notion de voisinage deux à deux des différents points de contrôle.

Cependant, cette représentation possède aussi ses limites :

- le gain de temps de calcul apporté par la discrétisation amène en contrepartie une perte de précision sur la reconstruction du contour de l'objet à segmenter. Cela conduit à un dilemme pour choisir entre le temps de réponse du modèle et sa précision, lié au nombre de points de contrôle. En effet, augmenter le nombre de points de contrôle permettra d'améliorer la précision du contour mais entrainera alors une baisse de sa vitesse d'évolution ;
- lorsque le contour a tendance à changer de taille de manière importante, un ré-échantillonnage de la courbe devient nécessaire de manière à conserver une précision suffisante. Pour certains modèles, cela signifie recalculer un certain nombre de paramètres dépendant directement du nombre de points de contrôle ;
- ce type de représentation ne permet pas de gérer naturellement les changements de topologie du contour. Des algorithmes additionnels doivent être implémentés pour permettre à plusieurs contours de fusionner ou de se séparer afin, par exemple, de segmenter plusieurs objets.

Plusieurs algorithmes de déformation s'appuient sur la représentation paramétrique : l'approche par calcul des variations [Kass *et al.*, 1988], la programmation dynamique [Amini *et al.*, 1990] et l'évolution gloutonne [Williams et Shah, 1992]. Ces trois méthodes sont indépendantes de la fonctionnelle d'énergie et peuvent donc s'appliquer à tout contour actif.

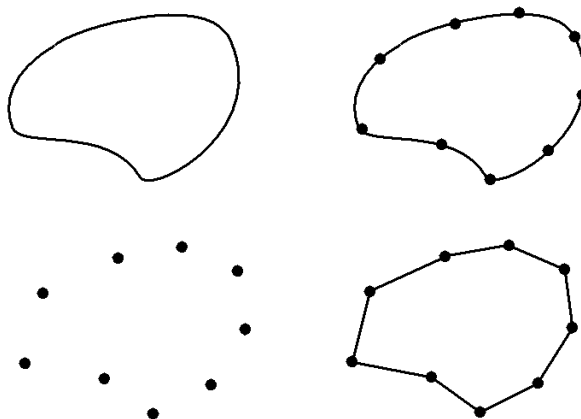


FIG. 1.1 – Principe d'échantillonnage de la représentation paramétrique d'un contour actif.

Calcul des variations

La première approche permettant de déformer un contour actif a été développée dans [Kass *et al.*, 1988]. Bien qu'applicable à tous les contours actifs, l'approche par calcul des variations reste complexe à appréhender dans un cadre complètement général, c'est pour-

1.1. FORME GÉNÉRALE D'UN CONTOUR ACTIF

quoi nous allons nous appuyer sur l'expression de la fonctionnelle d'énergie du modèle de Kass *et al*, que nous détaillerons au 1.2.1.

Le modèle de contour développé par Kass *et al* évolue d'après la minimisation de l'énergie suivante :

$$E(\mathbf{C}(s, t)) = \alpha \int_a^b |C'(s, t)|^2 ds + \beta \int_a^b |C''(s, t)| ds - \gamma \int_a^b P(s) ds . \quad (1.3)$$

Les termes en α et β sont les critères intrinsèques, P est le critère d'attache aux données. Dans le modèle présenté ici P est choisi proportionnel à la norme du gradient d'intensité de chaque pixel de la courbe (ainsi le terme $-\gamma \int_a^b P(s) ds$ tend vers 0 à mesure que la courbe se place sur des zones de fort gradient).

Le principe du calcul des variations est alors utilisé pour déterminer, à partir de l'équation (1.3), l'équation d'Euler-Lagrange régissant l'évolution du système. Celle-ci est composée des dérivées des différents critères composant la fonctionnelle d'énergie :

$$-\alpha \frac{\partial^2 \mathbf{C}(s, t)}{\partial p^2} + \beta \frac{\partial^4 \mathbf{C}(s, t)}{\partial p^4} = \gamma \frac{\partial P(s)}{\partial \mathbf{C}(s, t)} . \quad (1.4)$$

Minimiser l'énergie de l'équation (1.3) revient alors à résoudre l'équation différentielle (1.4).

Le modèle étant paramétrique, la courbe est discrétisée en n nœuds. En calculant les dérivées partielles de l'équation (1.4) à l'aide des différences finies, les auteurs aboutissent à l'expression matricielle linéaire :

$$\mathbf{A}\mathbf{V} = \mathbf{F} , \quad (1.5)$$

où $\mathbf{V}$ est le vecteur de coordonnées des n points du contour discrétisé, $\mathbf{F}$ est la dérivée du potentiel externe P . $\mathbf{A}$ est la matrice dite de lissage, de type matrice de Toeplitz pentadiagonale (matrice dont les coefficients sont constants pour chaque diagonale descendante), construite à partir des dérivées des critères intrinsèques (termes en α et β de l'équation (1.4)). $\mathbf{A}$ est symétrique si la courbe est fermée.

$\mathbf{V}$ est donc la solution recherchée (les coordonnées des points de contrôle minimisant l'énergie E), mais la matrice A n'est pas inversible (0 est une valeur propre). Les auteurs proposent alors de résoudre l'équation (1.5) de manière itérative en rendant le système dynamique. L'expression de la dérivée de la courbe $\mathbf{C}(s, t)$ par rapport au temps t est ainsi introduite dans l'équation (1.5) (celle-ci étant nulle lorsque la courbe a atteint un état stable) :

$$\frac{\partial \mathbf{V}}{\partial t} + \mathbf{A}\mathbf{V} = \mathbf{F} . \quad (1.6)$$

En discrétisant l'expression de la dérivée temporelle, l'expression itérative correspondant à l'équation (1.6) est alors déterminée :

$$(\mathbf{A} + \gamma I)\mathbf{V}^{t+1} = \mathbf{F} + \gamma \mathbf{V}^t . \quad (1.7)$$

$(\mathbf{A} + \gamma I)$ est inversible par décomposition LU (*Low Up*), $\mathbf{F}$ est fonction de $\mathbf{V}^t$. $1/\gamma$ représente le pas de temps, introduit suite à la discrétisation de la dérivée temporelle de l'équation (1.6).

Notons que l'utilisation des différences finies pour discrétiser les dérivées de l'équation (1.4) afin d'atteindre l'expression linéaire (1.5) n'est pas nécessaire. Une discrétisation par éléments finis est proposée dans [Cohen et Cohen, 1993] et [McInerney et Terzopoulos, 1995]. Le principe de cette approche est d'utiliser moins de points de contrôle et d'estimer la courbe entre chacun à l'aide de fonctions génératrices (fonctions de Bogner-Fix-Schmidt pour Cohen et Cohen).

Les deux approches peuvent être considérées comme équivalentes en temps de calcul car l'utilisation des éléments finis, bien qu'augmentant la complexité du modèle, utilise un nombre de points de contrôle moindre.

Les problèmes majeurs de ce type de méthode de déformation restent l'instabilité numérique due au calcul de dérivées d'ordre élevé et le risque de voir le contour s'arrêter sur un minimum local de l'image. Nous verrons l'application directe du calcul des variations utilisant les différences finies sur le modèle de Kass dans la section consacrée aux principaux modèles de contours actifs.

Programmation dynamique

Développée pour pallier les problèmes de l'évolution matricielle utilisée par le calcul des variations, l'implémentation par programmation dynamique introduite dans [Amini *et al.*, 1990] permet d'atteindre un *snake* d'énergie minimum sans calculer de dérivée d'ordre élevé mais en utilisant une technique de recherche plus efficace. Le principe de cette méthode est d'améliorer, itération après itération, un contour non définitif en calculant à chaque fois le meilleur contour accessible.

La fonctionnelle d'énergie générale des contours actifs (équation (1.2)) est discrétisée et devient (pour un *snake* à n points) :

$$E_{total}^* = \sum_{i=0}^{n-1} E_{int}(\mathbf{v}_i) + E_{ext}(\mathbf{v}_i) . \quad (1.8)$$

E_{total}^* représente l'énergie du contour optimal recherché, $\mathbf{v}_i$ est le $i^{ème}$ point du *snake*. Si l'on considère uniquement les dérivées d'ordre un comme énergie interne, la minimisation de l'équation (1.8) correspond à la minimisation d'une fonction de la forme :

$$\begin{aligned} E(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n) &= E_1(\mathbf{v}_1, \mathbf{v}_2) + E_2(\mathbf{v}_2, \mathbf{v}_3) + \dots \\ &+ E_{n-1}(\mathbf{v}_{n-1}, \mathbf{v}_n) , \end{aligned} \quad (1.9)$$

où $(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n)$ sont les n variables représentées par les points de contrôle du contour actif, ne possédant qu'un nombre fini m de valeurs possibles (le voisinage de chaque point).

Partant d'un point initial $\mathbf{v}_1$, le principe de la programmation dynamique est de traiter le problème de minimisation de telle manière qu'à chaque étape $i \in [1, n-1]$, une décision est prise à partir d'un ensemble fini de décisions possibles. Ainsi la programmation dynamique va générer un ensemble de solutions incomplètes $\{s_i\}_{i=1}^{n-1}$, où chaque s_i est obtenue en suivant :

$$s_i(\mathbf{v}_{i+1}) = \min_{\mathbf{v}_k} \{s_{k-1}\mathbf{v}_k + E_k(\mathbf{v}_k, \mathbf{v}_{k+1})\} \quad (1.10)$$

La solution complète est obtenue pour $i = n - 1$. Ainsi, pour un *snake* avec $n = 5$ défini suivant l'équation (1.9), nous aurons :

$$s_1(\mathbf{v}_2) = \min_{\mathbf{v}_1} E_1(\mathbf{v}_1, \mathbf{v}_2) \quad (1.11)$$

$$s_2(\mathbf{v}_3) = \min_{\mathbf{v}_2} s_1(\mathbf{v}_2) + E_2(\mathbf{v}_2, \mathbf{v}_3) \quad (1.12)$$

$$s_3(\mathbf{v}_4) = \min_{\mathbf{v}_3} s_2(\mathbf{v}_3) + E_3(\mathbf{v}_3, \mathbf{v}_4) \quad (1.13)$$

$$s_4(\mathbf{v}_5) = \min_{\mathbf{v}_1, \dots, \mathbf{v}_5} E(\mathbf{v}_1, \dots, \mathbf{v}_5) = \min_{\mathbf{v}_4} s_3(\mathbf{v}_4) + E_4(\mathbf{v}_4, \mathbf{v}_5) \quad (1.14)$$

A chaque itération, le contour optimal est donné par :

$$E_{min}(t) = \min_{\mathbf{v}_n} s_{n-1} \mathbf{v}_n \quad (1.15)$$

Cette méthode de déformation permet de résoudre certains problèmes de stabilité de l'approche par calcul des variations, engendrés par les dérivées d'ordre élevé présentes dans l'équation d'Euler-Lagrange du système. Elle garantit la convergence du modèle, mais au prix d'une perte de performance au niveau de la rapidité d'exécution (la complexité est en $O(nm^3)$, avec m la taille du voisinage et n le nombre de points de contrôle). En revanche, l'algorithme devient parallélisable.

Evolution gloutonne

Développée dans [Williams et Shah, 1992], l'évolution gloutonne reste une excellente alternative aux deux méthodes précédentes. Son principe, inspiré de l'approche gloutonne utilisée en optimisation, part de l'hypothèse qu'une succession de choix optimaux localement peut conduire à une solution optimale globalement. La position de chaque point de contrôle $\mathbf{v}_i = (x_i, y_i)$ est donc modifiée en limitant l'espace de recherche à son voisinage. Cette implémentation permet de passer à une complexité en $O(nm)$ où n est la taille de l'image et m la taille du voisinage considéré pour chaque point de contrôle.

Pour détailler le fonctionnement de l'évolution gloutonne, nous allons nous appuyer sur la fonctionnelle d'énergie de Kass *et al*, bien que cette approche reste applicable à tout type de fonctionnelle.

La fonctionnelle d'énergie basée sur celle du modèle de Kass possède trois énergies :

$$E_{Snake} = \int_0^1 (\alpha(s)E_{Cont} + \beta(s)E_{Courb} + \gamma(s)E_{Image})ds . \quad (1.16)$$

E_{Cont} et E_{Courb} correspondent aux critères intrinsèques de l'équation (1.3). E_{Image} est le critère d'attache aux données basé sur le gradient d'intensité. Cette fonctionnelle est discrétisée sur la grille de pixels et le long du contour. Elle est alors considérée comme la somme discrète des énergies des n points de contrôle :

$$E_{Snake} = \sum_{i=1}^n (\alpha_i E_{Cont}(\mathbf{v}_i) + \beta_i E_{Courb}(\mathbf{v}_i) + \gamma_i E_{Image}(\mathbf{v}_i))ds . \quad (1.17)$$

1.1. FORME GÉNÉRALE D'UN CONTOUR ACTIF

Le rôle de l'énergie de continuité $E_{Cont}(\mathbf{v}_i)$ est de garantir une certaine régularité dans la répartition des points de contrôle autour du contour. Elle est définie par :

$$E_{Cont}(\mathbf{v}_i) = \left| \bar{d} - \|\mathbf{v}_i - \mathbf{v}_{i-1}\| \right| . \quad (1.18)$$

$\bar{d}$ est la distance moyenne entre deux points de contrôle consécutifs. Plus la distance d'un point de contrôle par rapport à son voisin sera proche de la moyenne des distances, plus son énergie sera faible. $\bar{d}$ est recalculée à chaque itération. Bien que cette formulation soit efficace, il a été remarqué que le sens de parcours des points de contrôle introduisait un biais dans les calculs. Une énergie bilatérale permet de pallier cet inconvénient :

$$E_{Cont}(\mathbf{v}_i) = \frac{1}{2} \left(\left| \bar{d}^2 - \|\mathbf{v}_i - \mathbf{v}_{i-1}\|^2 \right| + \left| \bar{d}^2 - \|\mathbf{v}_i - \mathbf{v}_{i+1}\|^2 \right| \right) . \quad (1.19)$$

Par soucis de rapidité de calcul, la distance quadratique entre deux points est préférée à la distance Euclidienne.

L'énergie de courbure E_{Courb} représente le degré de courbure du contour en chaque point de contrôle. Elle est déterminée à l'aide de l'approximation par différences finies du carré de la dérivée seconde du vecteur position :

$$E_{Courb}(\mathbf{v}_i) = \|\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}\|^2 . \quad (1.20)$$

Plus la courbure du contour sera élevée en un point de contrôle, plus l'énergie de ce dernier sera grande.

Le terme $E_{Image}(\mathbf{v}_i)$ est l'énergie externe de chaque point de contrôle. Elle correspond à l'amplitude normalisée du gradient d'intensité, généralement déterminée en appliquant un masque de Sobel sur l'image.

Les coefficients α_i , β_i et γ_i contrôlent l'influence de chaque énergie pour le point de contrôle i .

À chaque itération de l'algorithme, la nouvelle position de chaque point de contrôle est recherchée dans son voisinage réduit. La fonctionnelle d'énergie est calculée pour chaque pixel appartenant à ce voisinage et celui possédant l'énergie minimale est alors choisi comme nouvelle position pour le point de contrôle. L'algorithme s'arrête lorsqu'il n'existe aucun pixel dans le voisinage des points de contrôle permettant de réduire l'énergie totale ou lorsque le contour actif oscille entre un nombre restreint de positions.

Dans [Williams et Shah, 1992] le lecteur pourra trouver des expérimentations comparant la rapidité d'exécution de l'évolution gloutonne par rapport au calcul variationnel et la programmation dynamique. Ces résultats mettent en avant la très grande rapidité de l'évolution gloutonne sans perte de précision au niveau du rendu du contour.

Il faut également noter que cet algorithme est basé sur l'approche énergétique de Kass mais qu'il n'y est pas restreint ; il peut s'adapter très rapidement à d'autres modèles de contours actifs en insérant de nouvelles énergies dans la fonctionnelle ou en modifiant les énergies existantes. De par son adaptabilité et sa rapidité d'implémentation cette technique reste très populaire et encore très utilisée.

Les trois méthodes de déformation présentées ici sont comparées dans le cadre du suivi d'objet en temps réel dans [Denzler et Niemann, 1995]. Les conclusions sont que la

déformation par programmation dynamique est la plus lente des trois méthodes, l'implémentation variationnelle classique, bien que plus rapide, reste très sensible au bruit et à l'initialisation du modèle. L'implémentation gloutonne est plus rapide que les deux approches précédentes (de l'ordre de quatre-vingts fois plus rapide pour la programmation dynamique et dix fois pour l'implémentation variationnelle) tout en conservant une extraction précise des contours.

Malgré quelques inconvénients, le mode de représentation paramétrique des contours actifs reste populaire de part sa simplicité de mise en œuvre et sa complexité réduite. La déformation par évolution gloutonne est, à l'heure actuelle, encore utilisée dans de nombreux travaux subissant une contrainte forte sur le temps d'exécution, comme le suivi d'objets dans des vidéos lorsque la topologie de l'objet à segmenter est connue [Pisa et Ruiz, 1998, Lam et Yuen, 1998, Takaya, 2008].

1.1.2.2 Représentation implicite d'un contour actif

La deuxième représentation utilisée pour les contours actifs est la représentation dite implicite ou Eulérienne. Alors que la représentation paramétrique fait se déformer un modèle dans un espace jusqu'à un état final, le principe de l'approche Eulérienne est de déformer l'espace entier et non plus un sous-ensemble. Les déformations du modèle seront alors définies implicitement au travers des déformations de l'espace tout entier. Cette approche a été adaptée dans le cadre de la segmentation dans [Osher et Sethian, 1988] sous le nom d'approche par ensembles de niveaux (*level set*).

Avant d'explorer plus en détails le fonctionnement de la représentation implicite, il est important de définir l'expression d'un contour actif lorsqu'il est ainsi représenté.

Alors qu'une fonctionnelle d'énergie du type de l'équation (1.2) est suffisante pour déformer un contour représenté de manière paramétrique, un contour représenté de manière implicite doit nécessairement être défini par une équation d'évolution, appelée également équation de mouvement ou équation aux dérivées partielles (EDP). Celle-ci définit les déformations subies par la courbe $\mathbf{C}(s, t)$ selon ses directions normale et tangentielle lors de la minimisation de son énergie $E(\mathbf{C}(s, t))$:

$$\frac{\partial \mathbf{C}(s, t)}{\partial t} = F_{\mathbf{N}(s, t)} \mathbf{N}(s, t) + F_{\mathbf{T}(s, t)} \mathbf{T}(s, t) . \quad (1.21)$$

$\mathbf{N}(s, t)$ et $\mathbf{T}(s, t)$ sont respectivement l'expression du vecteur normal et du vecteur tangentiel à la courbe $\mathbf{C}(s, t)$ au point d'abscisse curviligne s . Les fonctions $F_{\mathbf{N}(s, t)}$ et $F_{\mathbf{T}(s, t)}$, appelées fonctions vitesses, régissent respectivement l'évolution de la courbe selon ses directions normale et tangentielle. D'après [Epstein et Gage, 1987], si la vitesse est indépendante de la paramétrisation de la courbe, la vitesse tangentielle n'influence pas la déformation. L'équation (1.21) peut donc être simplifiée :

$$\frac{\partial \mathbf{C}(s, t)}{\partial t} = F_{\mathbf{N}(s, t)} \mathbf{N}(s, t) . \quad (1.22)$$

Ainsi, seule la fonction vitesse définie selon la normale F_N (désormais notée F par souci de simplification) influence la déformation.

De manière similaire à la fonctionnelle d'énergie, F sera composée de termes géométriques et de termes d'attache aux données. L'EDP d'un contour actif représenté à l'aide des ensembles de niveaux est généralement déduite mathématiquement de la fonctionnelle d'énergie en utilisant les équations d'Euler-Lagrange, bien que les premiers modèles ainsi représentés aient été géométriques, directement définis par leur EDP sans passer par la formulation d'une énergie. Pour les contours actifs basés régions, le théorème de Green-Riemann est généralement utilisé en amont afin de transformer les intégrales de contour en intégrales de région. Nous détaillerons les différentes techniques d'obtention de l'EDP utilisées dans la littérature ainsi que les modèles géométriques au 1.2.

Le principe général de la représentation en ensembles de niveaux est de décrire un objet de dimension n comme un niveau particulier d'une fonction de dimension $n+1$. L'évolution de l'objet de dimension n (i.e. le contour actif) ne sera non plus calculée directement, mais déduite de celle de la fonction de dimension supérieure, appelée fonction d'ensembles de niveaux.

Soit $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ la fonction d'ensembles de niveaux appliquée à la segmentation par contour actif d'une image en deux dimensions. Cette fonction peut être représentée comme une fonction en trois dimensions définie sur les deux dimensions de l'image, plus une hauteur correspondant à différents niveaux de plans. En d'autres termes, à chaque pixel $\mathbf{x}$ de l'image, la fonction $\phi(\mathbf{x})$ fait correspondre un niveau dans la représentation en trois dimensions. Le contour recherché sera alors donné par l'intersection de ϕ et du plan défini par $\phi = 0$. Initialement, le contour $\mathbf{C}(s, 0)$ est donné par le niveau zéro initial de $\phi(\mathbf{x}, 0)$ (voir figure 1.2). Cette relation étant conservée avec le temps, nous avons :

$$\forall \mathbf{x} \in \mathbb{R}^2, \forall s \in [a, b], \mathbf{C}(s, t) = \{ \mathbf{x} \in \mathbb{R}^2 | \phi(\mathbf{x}, t) = 0 \} . \quad (1.23)$$

Ainsi, pour tout pixel $\mathbf{x}$ appartenant au contour $\mathbf{C}(s, t)$, défini selon l'abscisse curviligne s , $\phi(\mathbf{x}, t) = 0$.

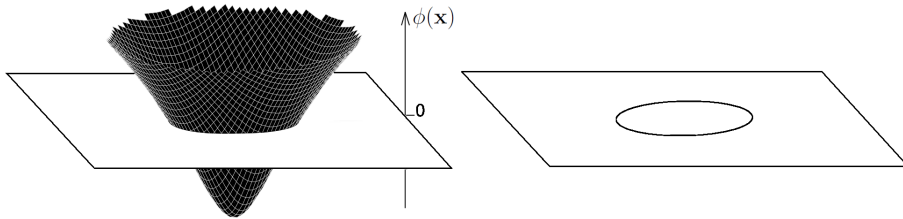


FIG. 1.2 – Représentation de $\phi(\mathbf{x}, t)$ et de son intersection avec le plan de niveau zéro : la forme en trois dimensions représentée sur l'image de gauche est la fonction d'ensembles de niveaux $\phi(\mathbf{x}, t)$ définie pour tout pixel $\mathbf{x}$ de l'image. Le contour $\mathbf{C}(s, t)$ représenté sur l'image de droite est déterminé par l'intersection de $\phi(\mathbf{x}, t)$ et du plan de niveau zéro.

En appliquant le principe de l'approche Eulérienne, ce n'est plus $\mathbf{C}$ qui va se déformer mais ϕ . L'évolution de $\mathbf{C}$ sera alors directement donnée par celle du niveau zéro de ϕ .

Avant de s'intéresser à l'évolution de la fonction d'ensembles de niveaux, il est nécessaire de déterminer sa valeur initiale. La fonction ϕ est généralement initialisée en chaque pixel de l'image comme la distance Euclidienne au point de contour initial le plus proche.

Ces distances sont signées par convention comme négatives pour les points à l'intérieur du contour et positives pour les points à l'extérieur. Ainsi initialement, ϕ est la carte des distances signées au contour et sa représentation en trois dimensions, dans le cadre d'une initialisation d'un contour circulaire, prend une forme de cône. La figure (1.3) illustre les différentes valeurs de ϕ lors de l'initialisation. D'autres initialisations de la fonction ϕ ont été proposées dans [Haker *et al.*, 2000].

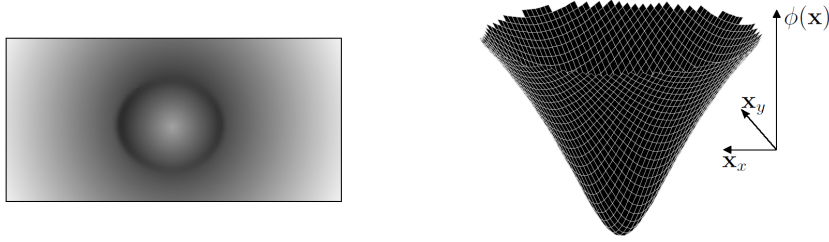


FIG. 1.3 – Initialisation de ϕ . Pour chaque pixel $\mathbf{x}$, $\phi(\mathbf{x}, 0)$ est définie comme la distance signée au contour. L'image de gauche représente la valeur absolue de ϕ lors de l'initialisation pour chaque pixel de l'image (normalisée entre 0 et 255), l'image de droite illustre la forme de cône alors prise par ϕ .

Les déformations de la fonction d'ensembles de niveaux sont déterminées grâce à l'équation d'évolution du modèle de contour actif (1.22). Afin de déterminer l'équation d'évolution de la fonction d'ensembles de niveaux, il est nécessaire d'exprimer la dérivée spatio-temporelle de l'équation (1.23). En utilisant dans un premier temps la dérivée spatiale, nous obtenons :

$$\nabla_{/s} \mathbf{C}(s, t) \cdot \nabla_{/\mathbf{x}} \phi(\mathbf{C}(s, t), t) = 0 . \quad (1.24)$$

$\nabla_{/s} \mathbf{C}(s, t)$ étant la dérivée de $\mathbf{C}(s, t)$ par rapport à son abscisse curviligne s , elle représente l'expression de la direction de la tangente à $\mathbf{C}(s)$ au point s . Ainsi, comme le produit scalaire $\nabla_{/s} \mathbf{C} \cdot \nabla_{/x} \phi$ est nul, alors $\nabla_{/x} \phi$ et le vecteur normal au contour $\mathbf{N}$ sont colinéaires. $\mathbf{N}$ s'exprime donc en fonction de ϕ comme :

$$\mathbf{N} = -\frac{\nabla \phi}{\|\nabla \phi\|} . \quad (1.25)$$

La dérivée temporelle de l'équation (1.23) nous donne :

$$\frac{\partial(\phi(\mathbf{C}(s, t), t))}{\partial t} + \nabla \phi(\mathbf{C}(s, t), t) \cdot \frac{\partial(\mathbf{C}(s, t), t)}{\partial t} = 0 . \quad (1.26)$$

D'après l'équation (1.22), $\partial(\mathbf{C}(s, t)/\partial t$ peut être remplacé par $F(s, t)\mathbf{N}$:

$$\frac{\partial(\phi(\mathbf{C}(s, t), t))}{\partial t} = -\nabla \phi(\mathbf{C}(s, t), t) \cdot F(s, t)\mathbf{N} , \quad (1.27)$$

qui devient (d'après (1.25)) :

$$\frac{\partial(\phi(\mathbf{C}(s, t), t))}{\partial t} = F(s, t) \|\nabla \phi(\mathbf{C}(s, t), t)\| . \quad (1.28)$$

Si la fonction vitesse F est définie sur toute l'image, l'équation (1.28) peut être étendue à tout le domaine de définition de l'image Ω_I :

$$\frac{\partial(\phi(\mathbf{x}, t))}{\partial t} = F(\mathbf{x}, t) \|\nabla\phi(\mathbf{x}, t)\| \quad \forall \mathbf{x} \in \Omega_I . \quad (1.29)$$

Cette extension ne conserve pas la fonction distance. Il est donc nécessaire de réinitialiser régulièrement les valeurs de ϕ en fonction de la position de son niveau zéro (et donc de $\mathbf{C}(s, t)$). Pour cela, on pourra utiliser un algorithme de calcul de distance d'un point à un objet du type transformation de distances [Roach et Mason, 1995].

Le principal avantage des contours actifs représentés implicitement est leur capacité à gérer naturellement les changements de topologie. En effet, l'intérêt de la représentation Eulérienne du modèle est que ce n'est plus le contour qui se déforme mais la fonction d'ensembles de niveaux. Ainsi, bien qu'un niveau change de topologie, la fonction en elle-même reste dans la topologie de départ (voir figure (1.4)). Donc ce n'est pas le modèle déformé qui change de topologie à proprement parler, mais simplement le sous-ensemble qui est observé (le contour actif).

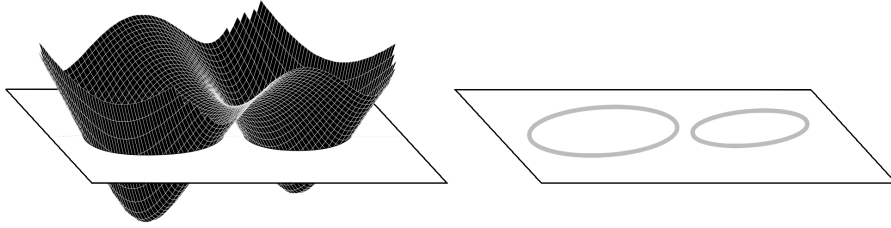


FIG. 1.4 – Illustration de la gestion des changements de topologie de l'implémentation en ensembles de niveaux : la fonction ϕ se déforme et, bien qu'elle ne change pas de topologie directement, le sous-ensemble observé (son intersection avec le plan de niveau zéro) change de topologie en donnant deux contours distincts

Aspects numériques

L'implémentation d'un contour actif à l'aide des ensembles de niveaux nécessite de nombreuses discrétisations pouvant parfois s'avérer complexes. Dans cette partie nous allons tenter d'éclaircir certains calculs.

Comme il est précisé à l'équation (1.25), le vecteur normal à la courbe $\mathbf{N}$ s'exprime comme le gradient normalisé de la fonction d'ensembles de niveaux. La définition de $\nabla\phi(\mathbf{x}, t)$ est donnée par :

$$\nabla\phi(\mathbf{x}, t) = \begin{pmatrix} \phi_x \\ \phi_y \end{pmatrix} . \quad (1.30)$$

$\phi_x = \partial\phi(x, y)/\partial x$ est l'approximation centrée de la dérivée partielle du premier ordre de la fonction d'ensembles de niveaux ϕ suivant la coordonnée x , et ϕ_y son équivalent suivant la coordonnée y . Les expressions de ϕ_x et ϕ_y sont données par :

$$\phi_x = \frac{\phi(x+1, y) - \phi(x-1, y)}{2\Delta x} \quad (1.31)$$

$$\phi_y = \frac{\phi(x, y+1) - \phi(x, y-1)}{2\Delta y} . \quad (1.32)$$

Ainsi, après discrétisation, la norme du gradient de ϕ devient :

$$\|\nabla\phi(\mathbf{x}, t)\|^2 = \phi_x^2 + \phi_y^2 . \quad (1.33)$$

Généralement, afin de garantir un contour lisse, la fonction vitesse F des contours actifs implémentés à l'aide des ensembles de niveaux possèdent un terme de courbure k . L'expression de k se détermine à l'aide de l'opérateur de divergence du gradient normalisé de ϕ :

$$\begin{aligned} k &= \operatorname{div} \frac{\nabla\phi(\mathbf{x}, t)}{\|\nabla\phi(\mathbf{x}, t)\|} \\ &= \frac{\phi_{xx}\phi_y^2 - 2\phi_{xy}\phi_x\phi_y + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}} . \end{aligned} \quad (1.34)$$

ϕ_{xx} , ϕ_{yy} et ϕ_{xy} sont les expressions des dérivées centrées d'ordre deux de la fonction ϕ selon chaque coordonnée, définies comme :

$$\begin{aligned} \phi_{xx} &= \frac{\partial^2\phi(x, y)}{\partial x^2} \\ &= \frac{\phi(x+1, y) - 2\phi(x, y) + \phi(x-1, y)}{4\Delta x^2} \end{aligned} \quad (1.35)$$

$$\begin{aligned} \phi_{yy} &= \frac{\partial^2\phi(x, y)}{\partial y^2} \\ &= \frac{\phi(x, y+1) - 2\phi(x, y) + \phi(x, y-1)}{4\Delta y^2} \end{aligned} \quad (1.36)$$

$$\begin{aligned} \phi_{xy} &= \frac{\partial^2\phi(x, y)}{\partial x\partial y} \\ &= \frac{\phi(x+1, y+1) - \phi(x+1, y-1) - \phi(x-1, y+1) + \phi(x-1, y-1)}{4\Delta x\Delta y} . \end{aligned} \quad (1.37)$$

A l'instar des différents termes présentés ici, l'équation d'évolution finale (1.29) nécessite également une discrétisation afin de permettre une évolution itérative du modèle. Celle-ci est réalisée en faisant intervenir un pas de temps Δt :

$$\phi(\mathbf{x})^{t+1} = \phi(\mathbf{x})^t + \Delta t F(\mathbf{x})^t \|\nabla\phi(\mathbf{x})^t\| . \quad (1.38)$$

La représentation implicite d'un contour actif accroît la complexité de la méthode en $O(n^2)$, où n est la taille de l'image. Un contour actif représenté de manière implicite sera donc plus lent qu'un paramétrique. Cela est notamment dû à la résolution des EDP et au calcul de k qui sont réalisés pour chaque pixel de l'image à chaque itération.

Pour cette raison, de nombreux algorithmes d'accélération ont été développés : la méthode *Fast Marching* [Sethian, 1995, Sethian, 1996], l'implémentation en bande étroite [Adalsteinsson et Sethian, 1995], l'algorithme Hermes [Paragios et Deriche, 2000] ou récemment l'algorithme de Shi et Karl permettant de s'affranchir de la résolution des EDP [Shi et Karl, 2005a, Shi et Karl, 2005b].

Fast Marching level sets

L'évolution *Fast Marching* proposée dans [Sethian, 1995, Sethian, 1996] a été développée dans le cadre de front se propageant de manière monotone. Ainsi la fonction vitesse F permettant de déformer la courbe est soit positive (correspondant à une dilatation de la courbe), soit négative (correspondant à un rétrécissement), mais ne changera jamais de signe.

Pour chaque pixel $\mathbf{x}$ de l'image, un temps $T(\mathbf{x})$ de passage du front est calculé. Les pixels de l'image sont alors répartis en trois classes :

- les pixels ayant déjà été rejoints par le front, pour lesquels le temps de passage est connu : les points calculés ;
- les pixels n'ayant pas encore été rejoints par le front mais dont la proximité par rapport aux points calculés leur permet d'être atteints dans l'itération courante : les points estimés ;
- les pixels n'ayant pas encore été rejoints par le front mais ne pouvant pas être atteints dans l'itération courante : les points lointains.

La figure 1.5 illustre les différentes classes de pixels.

L'évolution du temps de passage du front en chaque pixel $\mathbf{x}$ de l'image est donnée par :

$$\nabla T(\mathbf{x}) \cdot F(\mathbf{x}) = 1, \text{ avec } T = 0, \forall \mathbf{x} \in \mathbf{C}(s) . \quad (1.39)$$

L'algorithme de propagation est constitué d'une phase d'initialisation et d'une phase d'évolution. La fonction vitesse F est considérée positive.

1 - Initialisation

- a) Les points calculés (ceux situés à l'intérieur de la courbe initiale) sont initialisés à un temps de passage nul : $T(\mathbf{x}) = 0$.
- b) Les points estimés sont initialisés comme inversement proportionnels à leur fonction vitesse F : $T(\mathbf{x}) = \frac{1}{F(\mathbf{x})}$.
- c) Les points lointains sont initialisés à une valeur infinie : $T(\mathbf{x}) = \infty$.

2 - Evolution

- a) Parmi les points estimés, celui possédant la plus petite valeur de T est sélectionné ($\mathbf{x}_{min}$).
- b) $\mathbf{x}_{min}$ devient un point calculé.
- c) Pour tous les voisins $\mathbf{x}_v$ de $\mathbf{x}_{min}$ qui ne sont pas des points calculés (voisinage 4-connexe en 2D) :
 - si $\mathbf{x}_v$ est un point lointain, l'étiqueter comme point estimé ;
 - mettre à jour $T(\mathbf{x}_v)$ en résolvant l'équation (1.39).

L'algorithme s'arrête lorsque tous les points de l'image sont calculés.

Contrairement à l'implémentation classique en ensembles de niveaux, la méthode *Fast Marching* ne donne pas immédiatement les contours de l'objet recherché. En effet, l'information obtenue par cette approche est globale, le temps de passage du front est connu pour tous les pixels de l'image et il sera élevé pour les pixels situés sur les bords de l'objet. Il est donc nécessaire d'analyser les temps de passage sur l'image entière afin de déterminer les zones où le front a été fortement ralenti, correspondant aux frontières recherchées.

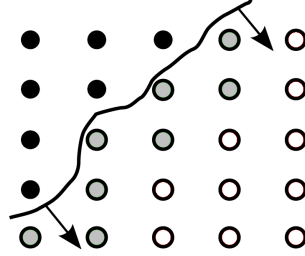


FIG. 1.5 – Différentes classes de pixels pour l'algorithme *Fast Marching*. Les flèches représentent le sens de propagation du front. Les pixels noirs sont les points calculés, les pixels gris les points estimés, et les pixels blancs représentent les points lointains.

Implémentation en bande étroite

Le principe de l'implémentation en bande étroite est d'optimiser les calculs en ne mettant plus à jour la fonction d'ensembles de niveaux $\phi(\mathbf{x})$ sur l'image entière, opération s'avérant très gourmande en temps de calcul, mais seulement à l'intérieur d'une bande autour du contour actif. A mesure que la courbe évolue, elle va se rapprocher d'une limite intérieure ou extérieure de la bande étroite. Celle-ci est alors réajustée de manière à retrouver une position symétrique par rapport à $\mathbf{C}(s)$. L'implémentation en bande étroite permet de passer d'une complexité en $O(n^2)$ pour l'implémentation classique (avec n le nombre de pixels de l'image), à une complexité en $O(n)$. La figure (1.6) illustre le principe de l'approche.

Cette implémentation reste plus lente que l'évolution *Fast Marching* mais, alors que cette dernière requière un front à propagation monotone, l'approche par bande étroite permet de gérer tout type de propagation de front.

Plus récemment, une amélioration de l'implémentation en bande étroite appelée *Fast Narrow Band* a été proposée dans [Yui *et al.*, 2002]. Le principe est de calculer la bande étroite en utilisant des voisinages circulaires autour des pixels appartenant au niveau zéro de ϕ .

Algorithme Hermes

Développé dans [Paragios et Deriche, 2000], l'algorithme Hermes part de l'hypothèse que l'implémentation en bande étroite réalise des mises à jour inutiles de la fonction d'ensembles de niveaux ϕ car certains pixels de la bande étroite n'évoluent plus (leur fonction vitesse F est nulle). L'algorithme Hermes est une combinaison des implémentations en bande étroite et *Fast Marching*. De l'implémentation en bande étroite est conservé le concept de fenêtre locale de mise à jour de la fonction d'ensembles de niveaux et de l'algorithme *Fast Marching* est gardé le concept de règle de sélection du prochain pixel pour lequel la fonction sera mise à jour.

La fonction ϕ est initialisée comme la distance signée au contour initial, mais de manière locale à l'intérieur d'une bande étroite considérée autour de chaque pixel du front (une

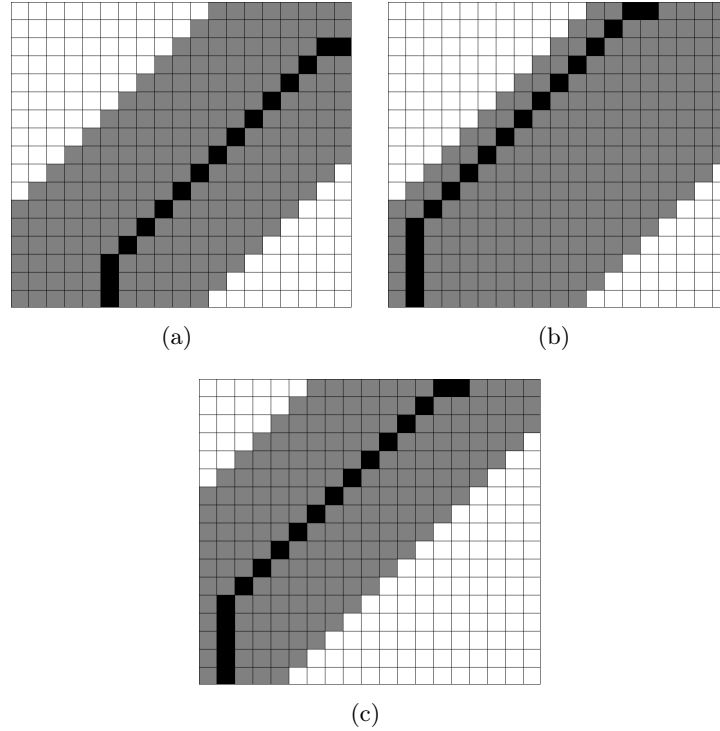


FIG. 1.6 – Principe de l'implémentation des ensembles de niveaux par bande étroite. Les pixels gris représentent la bande étroite et les pixels noirs le contour actif. a) la mise à jour de ϕ n'est effectuée qu'à l'intérieur de la bande b) le contour est déplacé et se rapproche de l'extrémité de la bande c) la bande étroite est alors réinitialisée.

bande de deux pixels de rayon suffit en pratique). Les fonctions vitesses F des pixels appartenant au front sont alors calculées et le pixel $\mathbf{x}$ possédant la plus grande valeur absolue de $F(\mathbf{x})$ (i.e. celle qui fera le plus évoluer le front) est sélectionné. La fonction d'ensembles de niveaux ϕ est ensuite mise à jour uniquement à l'intérieur de la bande étroite autour de $\mathbf{x}$ selon l'équation (1.38), permettant une évolution du front. La fonction ϕ est ensuite réinitialisée comme la distance au front pour tous les pixels de la bande étroite et seuls les pixels qui sont restés sur le front voient leur fonction vitesse recalculée. L'algorithme est alors répété pour le pixel du front possédant la plus grande valeur absolue de F jusqu'à ce que le front se stabilise.

En terme de rapidité, l'algorithme Hermes est très proche de l'évolution *Fast Marching*, mais permet de conserver les propriétés d'indépendance de l'implémentation en bande étroite par rapport au sens de propagation du front.

Implémentation rapide de Shi et Karl

L'implémentation rapide sans résolution des EDP, développée dans [Shi et Karl, 2005a, Shi et Karl, 2005b], part du constat que c'est lors de la résolution d'équations de type (1.38)

que les différentes implémentations en ensembles de niveaux perdent le plus de temps. Les auteurs proposent donc de s'affranchir du calcul coûteux que représente la résolution des EDP de l'image, et de n'utiliser que la fonction vitesse pour guider l'évolution.

Deux listes sont utilisées pour représenter le contour : celle des points frontières extérieurs L_{out} et celle des points frontières intérieurs L_{in} .

$$\begin{aligned} L_{out} &= \{\mathbf{x} \mid \phi(\mathbf{x}) > 0, \exists \mathbf{y} \in N(\mathbf{x}), \phi(\mathbf{y}) < 0\} , \\ L_{in} &= \{\mathbf{x} \mid \phi(\mathbf{x}) < 0, \exists \mathbf{y} \in N(\mathbf{x}), \phi(\mathbf{y}) > 0\} , \end{aligned} \quad (1.40)$$

avec $N(\mathbf{x})$ le voisinage discret autour de $\mathbf{x}$ composé de ses huit plus proches voisins.

Dans les implémentations classiques, la fonction d'ensembles de niveaux ϕ est conservée lisse, de manière à la garder cohérente par rapport à l'initialisation du modèle (distance au contour). Afin de simplifier la résolution de l'équation (1.38), et donc la mise à jour de $\phi(\mathbf{x})$, seules quatre valeurs possibles lui sont affectées, relativement à la position de $\mathbf{x}$:

$$\phi(\mathbf{x}) = \begin{cases} +1 & \text{si } \mathbf{x} \in L_{out} \\ -1 & \text{si } \mathbf{x} \in L_{in} \\ +3 & \text{si } \mathbf{x} \text{ est un point intérieur à } \mathcal{C} \text{ et } \mathbf{x} \notin L_{out} \\ -3 & \text{si } \mathbf{x} \text{ est un point extérieur à } \mathcal{C} \text{ et } \mathbf{x} \notin L_{in} . \end{cases} \quad (1.41)$$

La figure (1.7) décrit l'état initial de la méthode en illustrant les différentes valeurs prises par ϕ et les deux listes de points frontières.

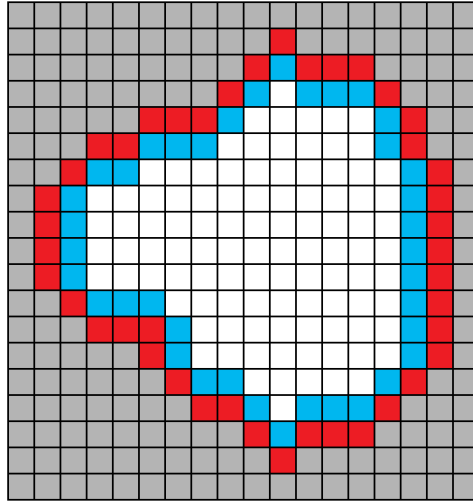


FIG. 1.7 – Différentes valeurs prises par la fonction d'ensembles de niveaux pour l'implémentation rapide de Shi et Karl : les points blancs sont les pixels intérieurs pour lesquels $\phi = -3$, les points bleus représentent les pixels appartenant à la liste L_{in} ($\phi = -1$), les points rouges sont les pixels appartenant à L_{out} ($\phi = 1$), et enfin les points gris sont les pixels extérieurs pour lesquels $\phi = 3$.

Une courbe est considérée comme ayant terminé son évolution si elle satisfait la condition suivante d'optimalité du contour :

1.1. FORME GÉNÉRALE D'UN CONTOUR ACTIF

La courbe $\mathbf{C}$, composée des points frontières L_{in} et L_{out} , est optimale si la fonction vitesse F satisfait :

$$F(\mathbf{x}) < 0, \forall \mathbf{x} \in L_{out} \text{ et } F(\mathbf{x}) > 0, \forall \mathbf{x} \in L_{in} . \quad (1.42)$$

Tant que cette condition d'optimalité n'est pas vérifiée, la courbe évolue en testant une condition de non-convergence pour chaque point de L_{in} et L_{out} :

$$Con(\mathbf{x}) = \begin{cases} 1 & \text{si } \exists \mathbf{y} \in N(\mathbf{x}) \text{ tel que } \phi(\mathbf{x})\phi(\mathbf{y}) < 0 \text{ et } F(\mathbf{x})F(\mathbf{y}) > 0 \\ 0 & \text{sinon.} \end{cases} \quad (1.43)$$

Si $Con(\mathbf{x})$ vaut 0, le point $\mathbf{x}$ satisfait alors la condition d'optimalité et n'évolue plus. Si $Con(\mathbf{x})$ vaut 1, le point $\mathbf{x}$ ne satisfait pas la condition d'optimalité et il est donc nécessaire de faire évoluer $\phi(\mathbf{x})$.

Deux fonctions sont définies pour faire évoluer un point de la courbe : *check_in* et *check_out*

check_in($\mathbf{x}$)

1. Supprimer $\mathbf{x}$ de L_{out} et l'ajouter à L_{in} . Fixer $\phi(\mathbf{x})$ à -1 et calculer $F(\mathbf{x})$.
2. $\forall \mathbf{y} \in N(\mathbf{x})$ tel que $\phi(\mathbf{y}) = 3$, ajouter $\mathbf{y}$ à L_{out} . Fixer $\phi(\mathbf{y})$ à 1 et calculer $F(\mathbf{y})$.

check_out($\mathbf{x}$)

1. Supprimer $\mathbf{x}$ de L_{in} et l'ajouter à L_{out} . Fixer $\phi(\mathbf{x})$ à 1 et calculer $F(\mathbf{x})$.
2. $\forall \mathbf{y} \in N(\mathbf{x})$ tel que $\phi(\mathbf{y}) = -3$, ajouter $\mathbf{y}$ à L_{in} . Fixer $\phi(\mathbf{y})$ à -1 et calculer $F(\mathbf{y})$.

Ainsi, à chaque itération, l'algorithme parcourt tous les points de L_{in} et L_{out} . Chaque point frontière $\mathbf{x}$ ne satisfaisant pas la condition d'optimalité (i.e. $Con(\mathbf{x}) = 1$) est alors déplacé vers l'intérieur ou l'extérieur au regard du signe de sa fonction vitesse $F(\mathbf{x})$. Si $F(\mathbf{x}) > 0$ pour un point de L_{out} , *check_in*($\mathbf{x}$) est alors réalisée, déplaçant ainsi la frontière vers l'extérieur. Si $F(\mathbf{x}) < 0$ pour un point de L_{in} , *check_out*($\mathbf{x}$) est réalisée, faisant alors évoluer la frontière vers l'intérieur. L'étape d'évolution peut faire apparaître certains points, n'appartenant pas au contour, dans L_{in} et L_{out} . Ceux-ci sont alors supprimés de manière à ce que L_{in} et L_{out} satisfassent toujours l'équation (1.40). La fonction vitesse F est n'est calculée pour chaque point de L_{in} et L_{out} .

De nombreuses fonctions vitesse possèdent un terme de courbure permettant de lisser la courbe. Ce terme est généralement proportionnel à la courbure du contour et calculé à partir de la fonction ϕ (voir équation (1.34)). D'une part les calculs entraînés par la courbure sont coûteux, et d'autre part ils requièrent une fonction ϕ lisse, avec des valeurs proches de la distance signée au contour. L'implémentation proposée par Shi et Karl autorise ϕ à ne prendre que quatre valeurs, rendant impossible les calculs précédents.

En conséquence, et afin de conserver les propriétés de lissage apportées par le terme de courbure classique, les auteurs réalisent le lissage de la courbe lors d'une étape séparée. A la fin de chaque itération, la courbe est modifiée après l'application d'un filtre gaussien G sur les valeurs discrètes de $\phi(\mathbf{x})$. Une étape complète de lissage de la courbe est réalisée comme suit :

Etape de lissage :

- $\forall \mathbf{x} \in L_{out}$, si $G \otimes \phi(\mathbf{x}) < 0$, $check_in(\mathbf{x})$
- $\forall \mathbf{x} \in L_{in}$, si $G \otimes \phi(\mathbf{x}) > 0$, $check_out(\mathbf{x})$

Chaque phase d'évolution de la courbe est donc suivie par une étape de lissage. L'algorithme détaillé de la méthode est disponible dans [Shi et Karl, 2005a].

L'accélération apportée par cette implémentation rapide a permis aux auteurs d'appliquer des contours actifs représentés à l'aide des ensembles de niveaux au suivi d'objet en temps réel dans des vidéos [Shi et Karl, 2005b].

Parmi toutes les implémentations en ensembles de niveaux présentées, nous avons décidé de réaliser nos modèles implicites en utilisant celle de Shi et Karl. Dans un premier temps, nous avons profité du gain de temps de calcul apporté par la méthode et, dans un deuxième temps (comme nous le détaillerons au 3.2.2.2), nous avons profité du principe d'évolution d'un contour actif ne reposant que sur sa fonction vitesse pour développer un nouveau modèle de contour actif supervisé.

Les deux modes de représentation décrits dans cette section permettent donc d'aborder différemment le problème de la segmentation par contour actif. Le mode de représentation d'un contour actif doit être choisi en fonction de l'application souhaitée. Des contraintes de temps fortes (telles qu'une application temps réel) ou une forte connaissance de la topologie de l'objet à segmenter orienteront vers une représentation paramétrique, alors que la segmentation d'objets multiples de topologies inconnues et une grande dépendance à la précision du résultat feront choisir une représentation implicite.

La structure générale d'un contour actif étant connue, nous allons détailler dans la section suivante les principaux modèles présents dans la littérature. Ceux-ci sont connus pour avoir eu une grande influence sur ce domaine de la segmentation d'images durant les vingt dernières années.

1.2 Modèles de contours actifs

La littérature concernant les contours actifs est abondante et de nombreux modèles ont été développés au cours des vingt dernières années. Cependant, bien que chaque modèle de contour actif possède des avantages, seuls certains ont eu une importance capitale et ont permis de faire évoluer de manière significative ce domaine de la segmentation d'images.

Dans cette seconde partie de notre état de l'art des contours actifs, ces modèles majeurs ont été regroupés en trois principales familles : les deux premières comprennent les contours définis par une fonctionnelle d'énergie, basés contour et basés région, alors que la troisième est composée de modèles exclusivement géométriques. Ces derniers sont des contours actifs basés contour possédant la particularité de ne pas être définis par une fonctionnelle d'énergie, mais directement par leur équation d'évolution.

1.2.1 Modèles de contours actifs basés sur l'information de contour

1.2.1.1 Modèle de Kass *et al*

Les premières approches de segmentation par contours actifs ont été développées dans [Kass *et al.*, 1988]. Dans leur approche variationnelle, les auteurs définissent l'énergie $E(C(s, t))$ associée au contour actif comme :

$$E(\mathbf{C}(s, t)) = \alpha \int_a^b |\mathbf{C}'(s, t)|^2 ds + \beta \int_a^b |\mathbf{C}''(s, t)| ds - \gamma \int_a^b P ds, \quad (1.44)$$

avec $\{\alpha, \beta, \gamma\} \in \mathbb{R}^+$ des constantes permettant de pondérer l'influence de chaque type de critère énergétique.

Les deux premiers termes de l'énergie (termes en α et β) sont les critères intrinsèques. Ces deux termes de régularisation sont en réalité un cas particulier de l'opérateur de Tikhonov d'ordre deux en choisissant des poids w_k constants [Tikhonov et Arsénine, 1976] :

$$E_{Tikhonov}(\mathbf{C}(s, t)) = \sum_{k=0}^n \int_{\mathbf{C}(s, t)} w_k(s) \left| \frac{\partial^k \mathbf{C}(s, t)}{\partial p^k} \right|^2. \quad (1.45)$$

Le critère intrinsèque du premier ordre de l'équation (1.44) ajoute une contrainte élastique sur la courbe $\mathbf{C}(s, t)$. Sa minimisation correspond en physique à l'équation d'évolution d'une membrane pour des déplacements faibles. Le terme du second ordre est une contrainte sur la rigidité de la courbe. Sa minimisation en physique est similaire à l'équation d'évolution d'une plaque mince pour des déplacements faibles.

Le premier critère intrinsèque agit donc sur la longueur de la courbe alors que le deuxième permet de contrôler sa courbure.

Le troisième terme (en γ) est le critère extrinsèque d'attache aux données. $P = \|\nabla I(\mathbf{C}(s, t))\|^2$ est le potentiel associé basé sur les gradients d'intensité de l'image. P augmente à mesure que la courbe se place sur des zones possédant de fortes variations d'intensité (i.e. les frontières d'un objet), entraînant ainsi une diminution de l'énergie globale du contour.

Le modèle de Kass *et al* est représenté de manière paramétrique. En utilisant le principe du calcul des variations et de la discrétisation à l'aide des différences finies, les auteurs déterminent l'équation d'évolution matricielle discrète du modèle (se référer au (1.1.2.1) pour plus de détails) :

$$(\mathbf{A} + \gamma I) \mathbf{V}^{t+1} = \mathbf{F} + \gamma \mathbf{V}^t. \quad (1.46)$$

$\mathbf{V}$ est le vecteur de coordonnées des n points de contrôle du contour discrétisé. $\mathbf{F}$ est la dérivée du potentiel externe P , fonction de $\mathbf{V}^t$. $\mathbf{A}$ est la matrice dite de lissage, de taille $n \times n$, de type matrice de Toeplitz pentadiagonale (symétrique si la courbe est fermée), et circulante. Elle est déterminée à l'aide des critères intrinsèques de la fonctionnelle d'énergie

et est de la forme :

$$\begin{bmatrix} 2\alpha + 6\beta & -\alpha - 4\beta & \beta & 0 & \dots & 0 & \beta & -\alpha - 4\beta \\ -\alpha - 4\beta & 2\alpha + 6\beta & -\alpha - 4\beta & \beta & 0 & \dots & 0 & \beta \\ \beta & -\alpha - 4\beta & 2\alpha + 6\beta & -\alpha - 4\beta & \beta & 0 & \dots & 0 \\ 0 & \beta & -\alpha - 4\beta & 2\alpha + 6\beta & -\alpha - 4\beta & \beta & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \beta & -\alpha - 4\beta & 2\alpha + 6\beta & -\alpha - 4\beta & \beta \\ \beta & 0 & \dots & 0 & \beta & -\alpha - 4\beta & 2\alpha + 6\beta & -\alpha - 4\beta \\ -\alpha - 4\beta & \beta & 0 & \dots & 0 & \beta & -\alpha - 4\beta & 2\alpha + 6\beta \end{bmatrix} \quad (1.47)$$

$(\mathbf{A} + \gamma I)$ est inversible par décomposition LU. $1/\gamma$ représente le pas de temps, introduit suite à la discrétisation de la dérivée temporelle de l'équation (1.6). Il a été montré dans [Cohen, 1991] qu'il est préférable de choisir γ très grand.

Ce modèle, premier du genre à avoir introduit les contours actifs comme outils de segmentation, possède néanmoins, outre les inconvénients liés à la représentation paramétrique décrits au 1.1.2.1, des inconvénients propres.

- Le contour est très sensible à l'initialisation, qui doit être réalisée à proximité de l'objet à segmenter.
- Malgré la contrainte de régularité, les points de la courbe ont tendance à se regrouper sur les zones de fort gradient, donnant lieu à des auto-intersections que le modèle est incapable de gérer.
- Seules les formes convexes sont bien segmentées, le modèle est plus approximatif sur les formes concaves.
- Le terme d'ordre deux de l'équation (1.44) introduit un terme d'ordre quatre dans l'équation d'Euler-Lagrange (1.4), entraînant une instabilité numérique du modèle.
- Il n'existe pas de méthode pour déterminer les paramètres α , β et γ car ils sont dépendant de l'image. L'utilisateur doit s'en remettre à des choix empiriques.

De nombreux modèles ont alors été développés de manière à atténuer ou supprimer ces inconvénients.

1.2.1.2 Critère extrinsèque de Cohen *et al*

Cohen *et al* se sont intéressés au critère extrinsèque d'attache aux données de l'équation (1.44), et plus particulièrement au potentiel P [Cohen, 1991]. Une fois le processus de dérivation réalisé, ce critère devient alors dans l'équation (1.46) une force définie par :

$$\mathbf{F}(v) = -\nabla P(v) . \quad (1.48)$$

Les auteurs font alors plusieurs constats.

- Lorsque la courbe se place sur des zones à faibles gradients, ceux-ci n'affectent pas l'évolution, car $\mathbf{F}(v)$ devient négligeable par rapport aux forces intrinsèques.
- Le rapport $\mathbf{F}(v)/\gamma$ (en respectant nos notations) ne doit pas être trop grand, sous peine de voir le point v passer au travers de certains contours et se mettre à osciller sans atteindre l'équilibre.

- Lorsque le contour actif se trouve trop loin d’une frontière, il ne sera pas attiré par celle-ci, et si la courbe n’est soumise à aucune force d’attache aux données, elle va se resserrer sur elle-même jusqu’à disparaître.
- Si l’image est bruitée, certains pixels isolés possèdent une force $\mathbf{F}(v)$ maximale et peuvent parfois bloquer le contour.

Afin de pallier ces inconvénients, les auteurs proposent alors une nouvelle force d’attache aux données :

$$F = k_1 \mathbf{N}(s) - k \frac{\nabla P}{\|P\|} . \quad (1.49)$$

$\mathbf{N}(s)$ est le vecteur normal unitaire à la courbe au point $\mathbf{C}(s)$. P est le potentiel associé au critère d’attache aux données et doit augmenter à mesure que la courbe se place sur des zones de forts gradients (similaire au potentiel de l’équation (1.44)). k_1 et k sont deux constantes permettant de pondérer chaque terme.

Le terme $k \cdot \nabla P / \|P\|$ permet de normaliser l’influence du potentiel P . En addition à un choix de γk de l’ordre du pixel, cela permet au contour actif de se stabiliser sur les frontières d’un objet sans entrer en conflit avec les forces intrinsèques.

$k_1 \mathbf{N}(s)$ est une force externe permettant de rendre le contour plus dynamique. Considérant le *snake* tel un ballon, $k_1 \mathbf{N}(s)$ va avoir un effet similaire à celui que procurerait la pression dans un ballon à mesure qu’on y introduirait de l’air. k_1 représente alors l’amplitude donnée à la nouvelle force d’inflation et permet, par son signe, de contrôler le sens d’évolution du contour (expansion ou resserrement).

Ce terme permet donc de forcer le contour à évoluer même s’il se trouve dans une zone de potentiel P nul. En outre, la courbe n’est plus arrêtée par les pixels isolés de fort potentiel, considérés comme du bruit.

k_1 et k sont choisis de manière à être du même ordre de grandeur, mais k reste légèrement supérieur de manière à stopper la progression de la courbe quand un contour marqué est rencontré tout en permettant de ne pas l’arrêter lorsque celui-ci est trop faible.

Un inconvénient de la nouvelle force proposée par Cohen *et al* est que le contour subit désormais de grandes modifications de taille car il peut être initialisé plus loin de l’objet. Cela rend nécessaire une re-paramétrisation de la courbe et donc une nouvelle inversion de matrice pour résoudre l’équation (1.46). De plus, la normalisation du gradient d’intensité rend égale l’influence de tous les contours, qu’ils soient marqués ou non. L’auteur utilise donc une segmentation globale en amont (de type Canny-Deriche).

1.2.1.3 Diffusion du gradient par champ vectoriel (gradient vector flow)

Dans [Xu et Prince, 1997, Xu et Prince, 1998], un nouveau terme d’attache aux données est proposé. Baptisé *Gradient Vector Flow (GVF)*, celui-ci est basé sur la diffusion du gradient par champ vectoriel.

Le principal inconvénient des critères d’attache aux données basés uniquement sur le gradient d’intensité est le caractère local de cette mesure. En effet, si un tel critère permet de stopper l’évolution du contour actif sur les forts gradients, il n’est pas en mesure d’attirer la courbe vers une frontière lorsque celle-ci est située dans une zone très homogène, éloignée d’une zone de fort gradient. Xu et Prince suggèrent alors un critère permettant au contour

actif d'avoir une vision plus globale de la répartition du gradient d'intensité dans l'image. Une diffusion du gradient d'intensité est alors proposée de telle sorte qu'en chaque pixel de l'image soit accessible l'information des forts gradients présents dans son voisinage. Ainsi, à chaque pixel $\mathbf{x}$ correspondra un vecteur $\mathbf{V}$ à deux dimensions (v_x, v_y) tel que :

$$v : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \quad (1.50)$$

$$\mathbf{x} \rightarrow \mathbf{V}(\mathbf{x}) = (v_x(\mathbf{x}), v_y(\mathbf{x})) . \quad (1.51)$$

Le vecteur $\mathbf{V}$ permet de définir l'énergie externe :

$$E_{GVF}(\mathbf{V}) = \int_{\Omega_I} \mu \left(\left\| \frac{\partial \mathbf{V}}{\partial x} \right\|^2 + \left\| \frac{\partial \mathbf{V}}{\partial y} \right\|^2 \right) + \|\nabla I(\mathbf{x})\|^2 \|\mathbf{V} - \nabla I\|^2 d\mathbf{x} . \quad (1.52)$$

Chacun des deux termes de l'énergie (1.52) permet de prendre en compte les valeurs de ∇I respectivement faibles et fortes. Le premier terme prédomine sur les faibles gradients et permet une régularisation du champ. Le deuxième terme devient prépondérant sur les forts gradients, le champ $\mathbf{V}$ a alors un effet identique au gradient classique.

Les auteurs déterminent V en résolvant de manière itérative le système formé par les équations d'Euler-Lagrange basées sur l'énergie (1.52) :

$$\begin{cases} \mu \nabla^2 \mathbf{V}_x - \left(\mathbf{V}_x - \frac{\partial I(\mathbf{x})}{\partial x} \right) \|\nabla I(\mathbf{x})\|^2 = 0 \\ \mu \nabla^2 \mathbf{V}_y - \left(\mathbf{V}_y - \frac{\partial I(\mathbf{x})}{\partial y} \right) \|\nabla I(\mathbf{x})\|^2 = 0 . \end{cases} \quad (1.53)$$

Le GVF a été intégré avec succès dans plusieurs modèles de contours actifs. Xu et Prince l'ont initialement appliqué à un modèle représenté de manière paramétrique [Xu et Prince, 1997] mais celui-ci a également été intégré à un contour actif géométrique, modèle représenté implicitement, dans [Paragios *et al.*, 2004].

1.2.1.4 Contours actifs géodésiques

Alors que la plupart des modèles implémentés à l'aide des ensembles de niveaux étaient jusque là basés sur une approche géométrique (le contour était directement défini par son équation d'évolution et pas par une énergie), un modèle défini par une fonctionnelle d'énergie et implémenté à l'aide des ensembles de niveaux, baptisé contour actif géodésique, fut proposé dans [Caselles *et al.*, 1997].

Partant de la fonctionnelle d'énergie définie par Kass *et al* (voir équation (1.44)), seul le terme du premier ordre est conservé comme critère intrinsèque de la courbe. En effet, ainsi qu'il est remarqué dans [Caselles *et al.*, 1997], fixer $\beta = 0$ dans l'équation (1.44) permet de conserver des propriétés géométriques suffisantes pour faire évoluer une courbe de manière lisse. Le terme d'attache aux données est également remplacé par une fonction $g : [0, +\infty[\rightarrow \mathbb{R}^+$, strictement décroissante, telle que $g(r) \rightarrow 0$ quand $r \rightarrow +\infty$. Un exemple de g est donné dans [Caselles *et al.*, 1997] :

$$g(I) = \frac{1}{1 + |\nabla \hat{I}|^p} , \quad (1.54)$$

où $\hat{I}$ est une version lissée de $I(\mathbf{x})$, intensité du pixel $\mathbf{x}$, et $p = 1$ ou 2 .
 Une nouvelle fonctionnelle d'énergie est donc définie :

$$E(\mathbf{C}(s, t)) = \alpha \int_a^b |\mathbf{C}'(s, t)|^2 ds + \lambda \int_a^b g(|\nabla I(\mathbf{C}(s, t))|) ds . \quad (1.55)$$

Malheureusement, cette fonctionnelle reste dépendante de la paramétrisation de la courbe. Les auteurs utilisent alors le principe de Mauperthuis [Dubrovin *et al.*, 1984] et montrent que minimiser une fonctionnelle de type (1.55) est équivalent à déterminer une courbe géodésique dans un espace de Riemann induit par l'image Ω_I , une courbe géodésique étant un chemin de longueur minimale entre deux points. Ainsi, minimiser l'énergie (1.55) est équivalent à la minimisation de :

$$E_R(\mathbf{C}(s, t)) = \int_a^b g(|\nabla I(\mathbf{C}(s, t))|) |\mathbf{C}'(s, t)| ds . \quad (1.56)$$

En déterminant les équations d'Euler-Lagrange du modèle défini par la minimisation de (1.56), l'équation d'évolution du modèle de contour actif géodésique est alors obtenue :

$$\frac{\partial \mathbf{C}(s, t)}{\partial t} = (g(I(s))k - \nabla g \cdot \mathbf{N}(s, t)) \mathbf{N}(s, t) , \quad (1.57)$$

où k représente la courbure du modèle.

Le contour actif est représenté implicitement et implémenté à l'aide des ensembles de niveaux. En considérant $F(\mathbf{x}, t) = g(I(\mathbf{x}))k - \nabla g \cdot \mathbf{N}(\mathbf{x}, t)$, l'expression de l'équation d'évolution appliquée à la fonction d'ensembles de niveaux devient :

$$\begin{aligned} \frac{\partial \phi(\mathbf{x}, t)}{\partial t} &= \left(g(I(\mathbf{x}))k - \nabla g \cdot \frac{\nabla \phi}{\|\nabla \phi\|} \right) \|\nabla \phi\| \\ &= g(I(\mathbf{x}))k \|\nabla \phi\| + \nabla g \cdot \nabla \phi , \end{aligned} \quad (1.58)$$

avec $k = \text{div} \frac{\nabla \phi(\mathbf{x}, t)}{\|\nabla \phi(\mathbf{x}, t)\|}$.

Cette approche possède l'avantage de s'affranchir du réglage des nombreux paramètres présents jusque là dans les précédents modèles. Contrairement aux termes d'attache aux données classiquement utilisés, le terme ∇g ne se contente pas de stopper le contour sur les zones possédant de forts changements d'intensités, il va également l'attirer vers celles-ci. Enfin, ces travaux furent parmi les premiers à unifier les approches énergétiques (où le contour actif est défini par une fonctionnelle d'énergie) et les approches par ensembles de niveaux (où le contour actif était jusque là dit géométrique, défini directement par son équation d'évolution).

1.2.2 Modèles de contours actifs basés sur l'information de région

Le principe des approches par contours actifs basés région est d'utiliser un critère d'attache aux données prenant en compte des caractéristiques appelées descripteurs de régions. Ceux-ci sont généralement calculés séparément sur la région formée par le contour actif et sur le reste de l'image (le fond), puis comparés. La différence entre les deux régions sert alors de critère pour faire évoluer la courbe.

1.2.2.1 Premiers travaux

Les premières approches de segmentation par contours actifs basés région sont apparues dans [Cohen *et al.*, 1993] et [Ronfard, 1994].

Cohen *et al* proposent une approche, pour la reconstruction d'une surface u par contour actif, basée sur l'hypothèse que celle-ci est composée de deux surfaces de régularités différentes. La première surface est modélisée comme un "lac" possédant une altitude basse et constante. La deuxième région entoure la première et possède une altitude élevée. La frontière entre les deux surfaces est alors modélisée comme une augmentation rapide de l'altitude (à la manière d'une "falaise").

En addition à l'énergie classique E_{snake} , les auteurs définissent une énergie associée à chaque surface. Ainsi la surface intérieure (le "lac") est décrite par une énergie E_{const} et la surface extérieure par une énergie $E_{outside}$.

La minimisation de l'énergie totale $E_g = E_{snake} + E_{const} + E_{outside}$ conduit alors à la reconstruction de la surface recherchée.

Dans [Ronfard, 1994], les auteurs partent de l'hypothèse que la vitesse d'un contour actif selon sa normale doit être proportionnelle à la différence entre les caractéristiques statistiques des régions intérieure et extérieure à la courbe. Afin de déterminer l'équation d'évolution du modèle en respectant cette hypothèse, l'énergie permettant de caractériser une région quelconque de l'image par la variance de ses niveaux de gris est tout d'abord définie :

$$E_{region}(R) = \int_R (I(\mathbf{x}) - \bar{I}_R)^2 d\mathbf{x} , \quad (1.59)$$

avec $\bar{I}_R$ la moyenne des intensités de niveaux de gris dans la région R .

L'énergie E_{region} est alors utilisée pour déterminer la distance entre deux régions, définie par une mesure de contraste connue comme la distance de Ward dans les travaux par séparation-fusion (*split and merge*) [Beaulieu et Goldberg, 1989] :

$$D(R_A, R_B) = E_{region}(R_A + R_B) - E_{region}(R_A) - E_{region}(R_B) . \quad (1.60)$$

La fonctionnelle d'énergie du modèle est alors définie comme la distance de Ward entre la région intérieure au contour et la région formée par le fond de l'image :

$$E(\mathbf{C}(\mathbf{s})) = D(R_{in}, R_{out}) . \quad (1.61)$$

En analysant les variations de E_{region} durant la déformation du modèle, Ronfard détermine l'équation d'évolution du contour actif respectant son hypothèse.

Ces deux approches ont permis d'initier la communauté des contours actifs aux modèles basés région et de nombreux modèles ont été développés par la suite. De notre point de vue, les approches présentées dans les sections suivantes peuvent être considérées comme les plus influentes.

1.2.2.2 Compétition de régions

Zhu et Yuille ont développé un modèle de contour actif basé région évoluant en fonction de l'estimation de la probabilité d'appartenance de chaque pixel de l'image à la région

formée par le contour [Zhu et Yuille, 1996]. Cette approche possède la particularité de ne pas segmenter qu'un seul type d'objet dans l'image, mais de réaliser une partition totale de l'image en M régions homogènes.

Ainsi, une région R_i sera considérée homogène si ses valeurs d'intensité sont proches de celles générées par une fonction de distribution de probabilité prédéfinie $P(I : \alpha_i)$ avec α_i l'ensemble des paramètres de la distribution. Dans le cas d'une distribution gaussienne $\alpha = (\mu, \sigma)$.

Le but de la compétition de régions est de diviser l'image Ω_I en M régions homogènes $R_i, i \in [1, M]$, telles que $\Omega = \bigcup_{i=1}^M R_i$ et $R_i \cap R_j = \emptyset$ si $i \neq j$. Γ_i représente la frontière de la région R_i . Ainsi la segmentation complète de l'image sera donnée par l'ensemble des contours ou frontières de segmentation de l'image $\Gamma = \bigcup_{i=1}^M \Gamma_i$.

Le critère MDL (*Minimum description length*) de la partition à réaliser est alors définis (équivalent à une énergie globale attachée à la partition complète et non juste à une région). Le critère MDL de la méthode correspond à une fonctionnelle d'énergie qui est la limite continue donnée dans [Leclerc, 1989] pour un bon choix de la famille de distributions de probabilités.

$$E[\Gamma, \{\alpha_i\}] = \sum_{i=1}^M \left\{ \frac{\mu}{2} \oint_{\partial R_i} ds - \int \int_{R_i} \ln P(I(\mathbf{x}) : \alpha_i) dx dy + \lambda \right\}. \quad (1.62)$$

L'équation (1.62) comporte deux inconnues : la segmentation Γ et l'ensemble des paramètres α_i .

Dans un premier temps, la segmentation Γ est fixée en disposant M régions initiales R_i (appelées germes), dans l'image. Les paramètres de la fonction de distribution de probabilité de chaque région R_i sont alors estimés de manière à minimiser le critère MDL de l'équation (1.62). Dans le cas d'une distribution Gaussienne, les α_i correspondront aux moyennes et écarts-types des pixels à l'intérieur de chaque région i .

Dans un deuxième temps, une fois les paramètres de chaque région connus, $P(I : \alpha_i)$ devient une probabilité conditionnelle $P(I|\alpha_i)$ et la frontière Γ peut alors être déterminée. Le théorème de Green permet de transformer les intégrales de région de l'équation (1.62) en intégrales de contour et les équations d'Euler-Lagrange sont utilisées afin de définir l'équation d'évolution engendrée par la minimisation de (1.62) pour tout point $\mathbf{v}$ situé sur la frontière Γ :

$$\frac{\partial \mathbf{v}}{\partial t} = \frac{\delta E[\Gamma, \{\alpha_i\}]}{\delta \mathbf{v}} = \sum_{i \in Q(\mathbf{v})} \left\{ -\frac{\mu}{2} k_i(\mathbf{v}) \mathbf{n}_i(\mathbf{v}) + \ln P(I(\mathbf{v})|\alpha_i) \mathbf{n}_i(\mathbf{v}) \right\}, \quad (1.63)$$

avec $Q_{\mathbf{v}} = \{i | \mathbf{v} \text{ est sur } \Gamma_i\}$, i.e. la somme est réalisée sur toutes les régions partageant la frontière Γ_i . $k_i(\mathbf{v})$ est la courbure de Γ_i au point $\mathbf{v}$ et $\mathbf{n}_i(\mathbf{v})$ est le vecteur normal extérieur de Γ_i au point $\mathbf{v}$.

L'équation (1.63) s'interprète de manière intuitive. Le terme $-\frac{\mu}{2} k_i(\mathbf{v}) \mathbf{n}_i(\mathbf{v})$ peut être considéré comme une force de lissage devenant plus forte à mesure que l'on se trouve sur des points possédant une courbure élevée. Le terme $\ln P(I(\mathbf{v})|\alpha_i) \mathbf{n}_i(\mathbf{v})$ est la force statistique. $\ln(P)$ étant inférieur ou égal à 0, cette force va toujours compresser la région. Mais plus le pixel $\mathbf{v}$ satisfera au critère d'homogénéité de la région, plus $P(I(\mathbf{v})|\alpha_i)$ sera grande et plus

la force appliquée sera proche de 0. Ainsi, son influence en sera considérablement réduite. L'équation d'évolution (1.63) se simplifie lorsque seules deux régions sont en compétition. En effet, dans ce cas de figure les contours Γ_i et Γ_j ayant des vecteurs normaux opposés au point $\mathbf{v}$, nous avons $\mathbf{n}_i(\mathbf{v}) = -\mathbf{n}_j(\mathbf{v})$ et $k_i(\mathbf{v})\mathbf{n}_i(\mathbf{v}) = k_j(\mathbf{v})\mathbf{n}_j(\mathbf{v})$. L'équation d'évolution de Γ_i au point $\mathbf{v}$ devient donc :

$$\frac{d\mathbf{v}}{dt} = -\mu k_i(\mathbf{v})\mathbf{n}_i(\mathbf{v}) + (\ln P(I(\mathbf{v})|\alpha_i) - \ln P(I(\mathbf{v})|\alpha_j))\mathbf{n}_i(\mathbf{v}) \quad (1.64)$$

$$= -\mu k_i(\mathbf{v})\mathbf{n}_i(\mathbf{v}) + \ln \left(\frac{P(I(\mathbf{v})|\alpha_i)}{P(I(\mathbf{v})|\alpha_j)} \right) \cdot \mathbf{n}_i(\mathbf{v}) \quad (1.65)$$

De même, après son initialisation, chaque région R_i évolue dans un premier temps de manière similaire à une croissance de région. Ce mode d'évolution peut être vu comme un cas particulier de la compétition entre deux régions : R_i une région de fonction de distribution $P(I : \alpha_i)$, et R_0 la région de fond de fonction de distribution uniforme P_0 . L'équation d'évolution de la frontière entre ces deux régions devient donc :

$$\frac{d\mathbf{v}}{dt} = (\ln P(I(\mathbf{v})|\alpha_1) - \ln P_0)\mathbf{n}(\mathbf{v}) , \quad (1.66)$$

où $\mathbf{n}$ est le vecteur normal au contour de la région. Ce cas se ramène à celui où l'on compare la probabilité $P(I(\mathbf{v}))$ avec le seuil P_0 .

La méthode de la compétition de régions permet donc d'unifier croissance de régions, contours actifs et *MDL* en un unique modèle statistique à trois étapes : hypothèses, tests, décisions.

Les hypothèses sont réalisées par le critère *MDL*, qui part du principe que l'image est constituée de régions possédant des frontières lisses et ayant des propriétés d'intensité homogènes, définies par une famille de distributions de probabilités. Les tests sont réalisés en observant, pour chaque région, le comportement des pixels placés sur les frontières par rapport à la distribution $P(I : \alpha)$. Enfin, les décisions correspondent aux mouvements de ces frontières. Ceux-ci sont en outre proportionnels et non sujets à un seuil absolu.

La méthode de compétition de régions permet également de réaliser une segmentation totale de l'image plutôt qu'une segmentation différente par région.

1.2.2.3 Le modèle de *Chan* et *Vese*

Développé dans [Chan et Vese, 2001], ce modèle de contour actif basé région peut être considéré comme un cas particulier du modèle de Mumford-Shah [Mumford et Shah, 1989]. L'approche est définie comme un problème de minimisation d'énergie afin de déterminer la partition optimale de deux régions séparées par un contour $\mathcal{C}$. Les auteurs définissent c_1 comme la moyenne des intensités de niveaux de gris dans la région intérieure au contour $\mathcal{C}$, et c_2 son équivalent pour la région extérieure au contour. Des critères intrinsèques sont également ajoutés afin de garantir un contour lisse et régulier tels que la longueur de la courbe $\mathcal{C}$ ou la taille de sa région intérieure.

La fonctionnelle d'énergie attachée au contour est donnée par :

$$\begin{aligned}
 E(c_1, c_2, \mathbf{C}) &= \mu \cdot \text{Longueur}(\mathbf{C}) + v \cdot \text{Aire}(\mathbf{C}) \\
 &+ E_1(c_1, \mathbf{C}) \\
 &+ E_2(c_2, \mathbf{C}) , \\
 \text{avec } E_1(c_1, \mathbf{C}) &= \lambda_1 \int_{\text{Intérieur}(\mathbf{C})} |I(\mathbf{x}) - c_1|^2 d\mathbf{x} \\
 E_2(c_2, \mathbf{C}) &= \lambda_2 \int_{\text{Extérieur}(\mathbf{C})} |I(\mathbf{x}) - c_2|^2 d\mathbf{x} ,
 \end{aligned} \tag{1.67}$$

et avec $\mu, v \geq 0$, $\lambda_1, \lambda_2 > 0$. E_1 et E_2 sont les critères d'attache aux données utilisés pour guider la courbe vers les frontières de l'objet à segmenter en utilisant c_1 et c_2 , les moyennes des intensités de niveaux de gris respectivement à l'intérieur et à l'extérieur du contour $\mathbf{C}$. Dans un schéma idéal, lorsque la courbe $\mathbf{C}$ se place sur la frontière de l'objet à segmenter, alors $E_1 \approx E_2 \approx 0$. Ainsi, minimiser l'équation (1.67) revient à déterminer la meilleure partition de l'image en deux régions homogènes séparées par la courbe $\mathbf{C}$. L'objet à segmenter correspondra alors à une des deux régions et son contour sera la courbe $\mathbf{C}$.

Afin de représenter leur modèle de manière implicite les auteurs introduisent dans l'équation (1.67) des versions modifiées de la fonction de Heavyside $H(z)$ et de la mesure de Dirac $\delta_0(z)$ définies par :

$$H_\epsilon(z) = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan\left(\frac{z}{\epsilon}\right) \right) \quad \delta_\epsilon(z) = \frac{d}{dz} H_\epsilon(z) . \tag{1.68}$$

La représentation graphique de $H_\epsilon(z)$ et $\delta_\epsilon(z)$ pour ϵ fixé est donnée en figure (1.8).

L'introduction de $H_\epsilon(z)$ et $\delta_\epsilon(z)$ dans la fonctionnelle d'énergie permet aux auteurs de simplifier les intégrales de région et d'exprimer chaque terme de l'équation (1.67) en fonction de la fonction d'ensembles de niveaux $\phi(\mathbf{x})$. L'énergie $E(c_1, c_2, \phi)$ du modèle formulé de manière implicite devient alors :

$$\begin{aligned}
 E_\epsilon(c_1, c_2, \phi) &= \mu \int_{\Omega} \delta_\epsilon(\phi(\mathbf{x})) |\nabla \phi(\mathbf{x})| d\mathbf{x} \\
 &+ v \int_{\Omega} H_\epsilon(\phi(\mathbf{x})) d\mathbf{x} \\
 &+ \lambda_1 \int_{\omega} |I(\mathbf{x}) - c_1(\phi)|^2 H_\epsilon(\phi(x, y)) d\mathbf{x} \\
 &+ \lambda_2 \int_{\omega} |I(\mathbf{x}) - c_2(\phi)|^2 (1 - H_\epsilon(\phi(x, y))) d\mathbf{x} .
 \end{aligned} \tag{1.69}$$

$c_1(\phi)$ et $c_2(\phi)$ sont alors données par la moyenne des intensités des pixels en fonction du signe de ϕ :

$$\begin{cases} c_1(\phi) = \text{moyenne}(I(\mathbf{x})) \text{ pour } \phi(\mathbf{x}) \geq 0 \\ c_2(\phi) = \text{moyenne}(I(\mathbf{x})) \text{ pour } \phi(\mathbf{x}) < 0 . \end{cases} \tag{1.70}$$

En fixant c_1 et c_2 dans l'équation (1.70) et en minimisant $E_\epsilon(\phi)$, l'équation d'Euler-Lagrange du modèle est alors déterminée. Un processus similaire à celui déterminant l'équation d'évolution du modèle de Chan et Vese a été utilisé pour le modèle de contour actif

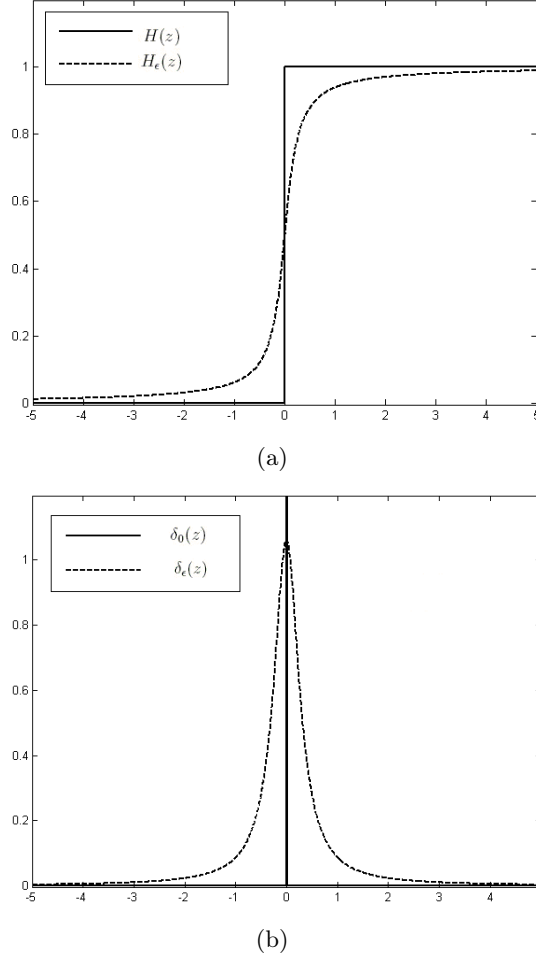


FIG. 1.8 – a) Fonctions de Heaviside classique et modifiée b) Fonctions de Dirac classique et modifiée.

supervisé utilisant la programmation linéaire que nous avons développé (voir au 3.2.1). Les détails mathématiques sont disponibles en annexe B.

L'EDP contrôlant la déformation de la fonction d'ensembles de niveaux devient alors :

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - v - \lambda_1 (I - c_1)^2 + \lambda_2 (I - c_2)^2 \right]. \quad (1.71)$$

Ce modèle s'affranchit donc de toute information de frontière et n'utilise que les moyennes des intensités de niveaux de gris dans les régions intérieure et extérieure comme critères d'attache aux données. Cette approche a été étendue à la segmentation d'images multi-canaux dans [Chan *et al.*, 2000].

Ce nouveau modèle de contour actif basé région s'est montré très efficace pour segmenter des images possédant des objets ne pouvant pas être séparés par des gradients d'intensité et a été utilisé dans de nombreux domaines [Solem *et al.*, 2006, Cao *et al.*, 2008, Peng *et al.*, 2009].

En revanche, le terme d'attache aux données basé sur les moyennes d'intensités reste limité pour une application sur des images texturées. En effet deux textures différentes peuvent avoir des moyennes d'intensités sensiblement égales (comme nous l'illustrerons au chapitre 2), ce qui rend l'utilisation du modèle de Chan et Vese délicate.

1.2.2.4 Régions actives géodésiques

Paragios et Deriche ont proposé un modèle de contour actif basé région pour la segmentation supervisée d'images texturées permettant de réaliser une partition de l'image en R_N régions possédant des caractéristiques de texture homogènes [Paragios et Deriche, 1999, Paragios et Deriche, 2002].

Afin de guider la courbe en utilisant l'information de texture, les auteurs utilisent un banc de F_N filtres (opérateurs gaussiens, isotropes et de Gabor). Malheureusement, en fonction du type de texture recherché, le pouvoir discriminant de chaque filtre varie. Une étape supervisée est alors utilisée pour permettre au modèle de sélectionner les filtres les plus efficaces parmi les F_N proposés.

Un ensemble de T_N modèles de textures est utilisé comme un ensemble d'apprentissage. Pour chaque modèle de texture, la réponse de chaque filtre est calculée et un modèle statistique y est associé. Cette étape permet de déterminer pour chaque type de texture i une mesure de probabilité $p_{i,j}$ pouvant être interprétée comme la probabilité que la texture i soit correctement caractérisée par le filtre j .

Ainsi, à chaque modèle de texture i est associé un vecteur de fonctions de densité de probabilité caractérisant son comportement par rapport à chaque filtre :

$$\mathbf{p}_i = \{p_{i,1}, p_{i,2}, \dots, p_{i,F_N}\} . \quad (1.72)$$

L'énergie du modèle de Paragios et Deriche peut être séparée en deux termes distincts : une énergie contour, inspirée des contours actifs géodésiques, et une énergie entièrement basée région.

Partant de l'équation (1.56) des contours actifs géodésiques, les auteurs proposent de remplacer le critère d'attache aux données $g(\nabla I)$ par $g(p_B(\mathbf{x}))$. $p_B(\mathbf{x})$ représente la plus grande probabilité d'appartenance du pixel $\mathbf{x}$ à un des T_N modèles de texture, déterminée en utilisant les vecteurs de fonctions de densité de probabilité de l'équation (1.72). L'énergie basée région est définie par :

$$E_{region}(s, t) = \sum_{i=0}^{R_N} \int_{R_i} \sum_{j=1}^{F_N} w_j \ln[p_{R_i,j}(I_j(s))] . \quad (1.73)$$

$I_j(s)$ est la réponse du filtre j au pixel du contour s , $p_{R_i,j}(I_j(\mathbf{x}))$ peut être interprétée comme la probabilité d'appartenance du pixel $\mathbf{x}$ à la région R_i pour le filtre j et α est une constante négative. Tous les filtres n'étant pas pertinents, cette probabilité est pondérée par w_j , coefficient déterminant la puissance de discrimination de chaque filtre. w_j est déterminé grâce à l'étape supervisée.

Finalement, l'énergie complète de la région active géodésique s'écrit :

$$E(\mathbf{C}(s, t) = (1 - \alpha) \int_a^b g(p_B(s)) |\mathbf{C}'(s, t)| ds + \alpha \sum_{i=0}^{R_N} \int_{R_i} \sum_{j=1}^{F_N} w_j \ln[p_{R_i, j}(I_j(s))] . \quad (1.74)$$

Ainsi, le terme contour va permettre d'obtenir une courbe lisse sur les frontières de chaque texture, pendant que le terme région assurera une homogénéité à l'intérieur même de chaque région.

Afin de représenter le modèle de manière implicite, cette énergie est minimisée en utilisant une technique de descente de gradient. Considérant le mouvement d'une région R_k par rapport au fond R_0 , les auteurs calculent les équations d'Euler-Lagrange du système pour obtenir l'équation d'évolution :

$$\frac{\partial \mathbf{C}(s, t)}{\partial t} = \left[(1 - \alpha) (g(p_B(s))k - \nabla g(p_B(s)) \cdot \mathbf{N}) + \alpha \sum_{j=1}^{F_N} w_j \ln \left(\frac{p_{R_0, j}(I_j(s))}{p_{R_k, j}(I_j(s))} \right) \right] \mathbf{N} . \quad (1.75)$$

Ainsi, un contour actif évoluant selon l'équation (1.75) cherchera à créer une partition de l'image où la région extérieure correspondra au modèle de texture du "fond" et où la région intérieure correspondra aux autres modèles.

L'équation du modèle implicite devient alors évidente :

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \left[(1 - \alpha) \left(g(p_B(\mathbf{x}))k - \nabla g(p_B(\mathbf{x})) \cdot \frac{\nabla \phi}{\|\nabla \phi\|} \right) + \alpha \sum_{j=1}^{F_N} w_j \ln \left(\frac{p_{R_0, j}(I_j(\mathbf{x}))}{p_{R_k, j}(I_j(\mathbf{x}))} \right) \right] \|\nabla \phi\| . \quad (1.76)$$

Cette approche fut une des premières à utiliser des caractéristiques de textures comme descripteurs de régions. Le problème de la sélection des caractéristiques (ici les différents filtres) est un problème connu de l'analyse de textures que nous approfondirons au chapitre 2. La solution trouvée dans le cas présent est de placer le contour actif dans un processus de segmentation supervisé.

Les régions actives géodésiques ont été adaptées avec succès au suivi d'objet dans [Paragios et Deriche, 2005].

1.2.2.5 Schéma général pour les contours actifs basés région

Des travaux intéressants ont été menés dans [Jehan-Besson *et al.*, 2003] afin de déterminer un schéma général pour les contours actifs basés région. Etant donnés $k^{(in)}$ et $k^{(out)}$ les descripteurs des régions intérieure et extérieure à $\mathbf{C}$ respectivement, et k^b un decripteur utilisant uniquement l'information de contour, une équation d'évolution générale est définie telle que :

$$\frac{\partial C(t)}{\partial t} = \left[k^{(in)} - k^{(out)} + k^b \cdot k - \nabla k^b \cdot \mathbf{N} \right] \mathbf{N} . \quad (1.77)$$

La plupart des modèles basés région présentés dans cet état de l'art respectent ce schéma général.

Ainsi, dans le cas de la compétition entre deux régions (équation (1.64)), le modèle de Zhu et Yuille peut être intégré dans ce schéma général tel que :

$$\begin{cases} k^{(in)} &= (\log P(I_{\mathbf{v}}|\alpha_i) - \log P(I_{\mathbf{v}}|\alpha_j)) \\ k^{(out)} &= \log P(I_{\mathbf{v}}|\alpha_j) \\ k^{(b)} &= -\mu . \end{cases} \quad (1.78)$$

Le modèle de Chan-Vese (équation (1.71)) peut être défini au travers de ce schéma par :

$$\begin{cases} k^{(in)} &= -\lambda_1(I - c_1)^2 \\ k^{(out)} &= -\lambda_2(I - c_2)^2 \\ k^{(b)} &= \mu . \end{cases} \quad (1.79)$$

Le modèle de région active géodésique (équation (1.75)) peut également être formulé en prenant :

$$\begin{cases} k^{(in)} &= \alpha \sum_{j=1}^M w_j \ln(p_{R_{k,j}}(I_j(s))) \\ k^{(out)} &= \alpha \sum_{j=1}^M w_j \ln(p_{R_{0,j}}(I_j(s))) \\ k^{(b)} &= (1 - \alpha)g(p_B(s)) . \end{cases} \quad (1.80)$$

1.2.3 Contours actifs basés sur une approche géométrique

Les contours actifs géométriques ont la particularité de déterminer l'EDP d'évolution du contour sans passer par la minimisation d'une énergie. Ces modèles sont généralement développés par analogie avec la physique ou purement mathématiques.

Les premières approches sur les contours actifs géométriques ont été développées dans [Malladi et Sethian, 1995, Malladi et Sethian, 1996, Malladi *et al.*, 1995]. Les auteurs considèrent une courbe évoluant selon l'équation :

$$\frac{\partial(\mathbf{C}(s,t))}{\partial t} = -k\vec{\mathbf{N}} . \quad (1.81)$$

où k représente la courbure locale de la courbe $C(s,t)$.

Cette équation correspond à une version géométrique de l'équation de la chaleur. Les courbes évoluant selon l'équation (1.81) possèdent des propriétés de lissage. Malheureusement, sur le long terme, toute l'information est perdue alors que les contours se resserrent et finissent par disparaître. Afin de pallier cet inconvénient, un algorithme basé sur les flots de courbure est proposé. Celui-ci utilise une fonction vitesse de courbure min/max permettant de changer l'équation d'évolution du modèle telle que :

$$\begin{aligned} \frac{\partial(\mathbf{C}(s,t))}{\partial t} &= F_{min/max} \vec{\mathbf{N}} \\ \text{avec } F_{min/max} &= \begin{cases} \min(k, 0) & \text{si } f(s) < 0 \\ \max(k, 0) & \text{sinon.} \end{cases} \end{aligned} \quad (1.82)$$

avec $f(s)$ la valeur moyenne des intensités de niveaux de gris dans un voisinage circulaire de taille fixe autour de s . Ce modèle est appliqué avec succès au filtrage de bruit dans des images naturelles.

Une équation d'évolution pour la segmentation d'objet par contour actif est également proposée dans [Malladi et Sethian, 1995, Malladi *et al.*, 1995], avec la fonction vitesse suivante :

$$\begin{aligned} F &= k_i(1 - \epsilon k) \\ k_i &= \frac{1}{1 + |\nabla I|} . \end{aligned} \tag{1.83}$$

Le front s'étend à la vitesse constante $\alpha = 1$ ou se contracte à la vitesse constante $\alpha = -1$. Ce terme est donc une analogie avec la force d'inflation introduite dans [Cohen, 1991] pour les contours à approche variationnelle. La constante ϵ est utilisée pour modérer les propriétés de lissage du terme de courbure. Ce dernier possède les mêmes propriétés régularisantes que l'énergie interne du premier modèle de contour actif variationnel introduit dans [Kass *et al.*, 1988].

Le terme k_i permet de forcer le contour à s'arrêter sur les forts gradients d'intensité présents dans l'image (i.e. l'objet recherché).

Parallèlement à Malladi *et al.*, Caselles *et al.* développent également dans [Caselles *et al.*, 1993] un modèle de contour actif géométrique évoluant selon l'équation :

$$\begin{aligned} \frac{\partial(\mathbf{C}(s, t))}{\partial t} &= g(\nabla I)(v + k)\vec{\mathbf{N}} \\ g(\nabla I) &= \frac{1}{1 + (\nabla(G_\sigma * I))} . \end{aligned} \tag{1.84}$$

k représente la courbure de chaque point du contour. Son rôle, équivalent à celui de l'énergie interne des contours variationnels [Kass *et al.*, 1988], est de contrôler l'évolution de $\mathbf{C}(s, t)$ en fonction de sa courbure locale, ajoutant un effet régularisant. Le terme v peut être interprété comme une force permettant à $\mathbf{C}(s, t)$ de continuer à se rapprocher de l'objet à segmenter quand la courbure devient nulle ou négative. Son action est similaire à la force ballon développée dans [Cohen, 1991]. Enfin $g(\nabla I)$ permet de contrôler la vitesse d'évolution de la courbe $\mathbf{C}(s, t)$, lui permettant de s'arrêter sur les forts gradients de l'image.

L'auteur propose également dans [Caselles *et al.*, 1993] une preuve d'existence d'une solution de l'équation (1.84) en s'appuyant sur la théorie des solutions de viscosité et une méthode d'estimation du paramètre v en fonction des connaissances *a priori* de l'image.

1.3 Analyse des caractéristiques des contours actifs et améliorations proposées

Dans cet état de l'art, nous avons tenté dans un premier temps, de dégager une structure générale pour les contours actifs, puis de présenter les modèles ayant eu une influence majeure sur ce domaine de la segmentation d'images.

En résumant les caractéristiques de chaque type de contour actif, les modèles basés sur une approche contour et une représentation paramétrique peuvent être considérés comme étant les plus rapides, et les modèles basés sur une approche région et une représentation implicite comme étant les plus précis.

Il est néanmoins important de remarquer que le choix d'un modèle de contour actif doit toujours être guidé par l'application finale. Il n'est pas nécessaire de développer des modèles complexes, tels que des approches région, si les données traitées ne le nécessitent pas (images possédant des contours nets par exemple). En revanche, si les données traitées sont très bruitées avec des frontières mal définies, un modèle basé région sera nécessaire. De la même façon, lorsque la topologie de l'objet recherché est connue, il est parfois plus intéressant d'utiliser une représentation paramétrique du modèle afin de bénéficier de sa rapidité d'exécution. Réciproquement, si la précision de la segmentation et la gestion des changements de topologie est une contrainte forte, une représentation implicite sera préférée.

Les travaux sur les contours actifs proposés dans cette thèse ont pour but d'améliorer sensiblement les méthodes existantes. Or, combiner les avantages de chaque type de modèle en un contour actif unique peut être considéré comme une utopie. En effet, il s'avère par exemple difficile pour un modèle représenté de manière paramétrique d'égaliser la précision des meilleurs modèles implicites. De même, un modèle implicite éprouvera beaucoup de difficultés à concurrencer un modèle paramétrique en terme de rapidité d'exécution.

Ainsi, il a été décidé de travailler sur chacun des deux modes de représentation de manière à le rendre encore plus efficace sur ses point forts : la rapidité pour la représentation paramétrique et la précision pour la représentation implicite. Les travaux concernant la représentation paramétrique sont présentés au 3.1 et ceux concernant la représentation implicite au 3.2.

Afin d'améliorer la précision des modèles implicites, une approche texture a été retenue. Un modèle de contour actif basé texture peut être vu comme un modèle région où les descripteurs de régions ont été remplacés par des descripteurs de textures (de manière similaire au modèle développé dans [Paragios et Deriche, 1999, Paragios et Deriche, 2002]). La plus grande difficulté réside alors dans un choix approprié des descripteurs de textures utilisés.

Afin de guider notre choix, un état de l'art des descripteurs de textures est proposé dans le chapitre suivant.

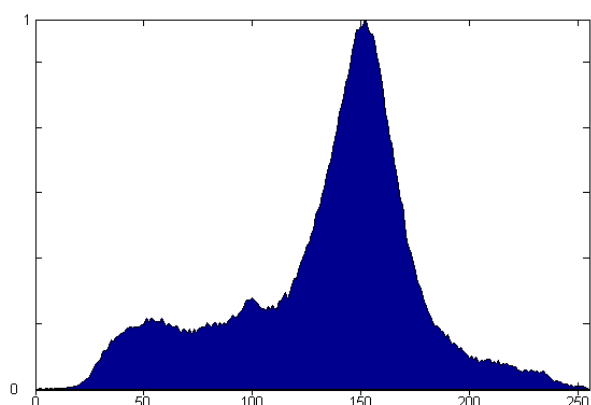
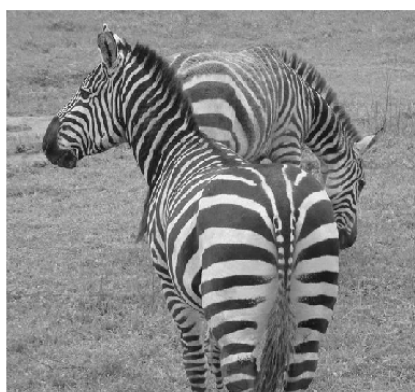
1.3. ANALYSE DES CARACTÉRISTIQUES DES CONTOURS ACTIFS ET AMÉLIORATIONS PROPOSÉES

Chapitre 2

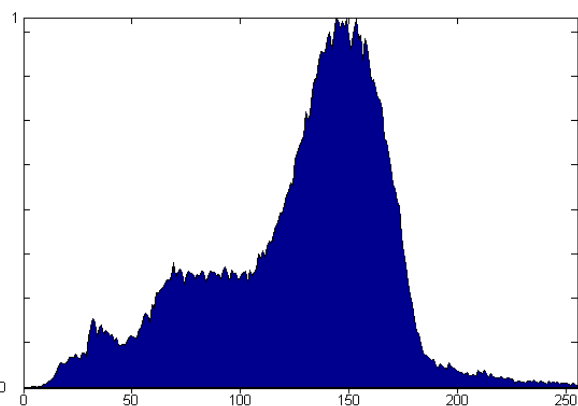
Analyse d'images grâce à l'information de texture

Malgré de bons résultats, l'analyse d'images basée uniquement sur l'étude statistique des intensités de pixels reste limitée dans le cas d'images complexes telles que les images texturées. Les images naturelles sont généralement composées d'objets texturés, caractérisés par des motifs pouvant avoir des statistiques d'intensités équivalentes, mais ayant des rendus visuels complètement différents comme l'illustre la figure (2.1). De tels objets nécessitent des méthodes d'analyses de plus haut niveau telles que l'analyse de textures. Cette dernière permet en effet d'étudier les pixels d'une image non seulement à partir de statistiques sur leurs intensités (niveaux de gris ou couleurs), mais également au niveau de leur répartition spatiale.

Cette partie de l'état de l'art s'intéresse donc à la caractérisation des textures en analyse d'images en niveaux de gris. Avant toute chose, nous allons tenter d'établir une définition de la notion de texture. Nous nous intéresserons ensuite aux principaux descripteurs de textures connus et enfin nous discuterons des diverses applications de l'analyse de textures dans le domaine de la vision par ordinateur.



(a) Moyenne des intensités : 133.9; Ecart type : 42.7



(b) Moyenne des intensités : 126.5; Ecart type : 40.7

FIG. 2.1 – Illustration des limites de l’analyse d’images basée uniquement sur l’étude des niveaux de gris : les images a) et b) ont des statistiques d’intensité de niveaux de gris presque identiques (moyenne, écart type et histogramme), mais sont visuellement très différentes. Une analyse de plus haut niveau telle que l’analyse de textures est donc nécessaire

2.1 Notion de texture

Définir de manière unique la notion de texture est une tâche difficile car il existe de nombreuses définitions dans la littérature. Tuceryan et Jain proposent un panel de définitions connues dans [Tuceryan et Jain, 1993] et expliquent leur diversité en précisant que celles-ci sont, la plupart du temps, formulées au travers d’un contexte applicatif particulier. En essayant de se placer dans un cadre général, nous pouvons définir une zone texturée d’une image comme une distribution de niveaux de gris respectant un schéma ordonné pour laquelle il est possible de déterminer des caractéristiques uniques.

Deux principaux types de textures émergent néanmoins couramment [Richards et Polit, 1974].

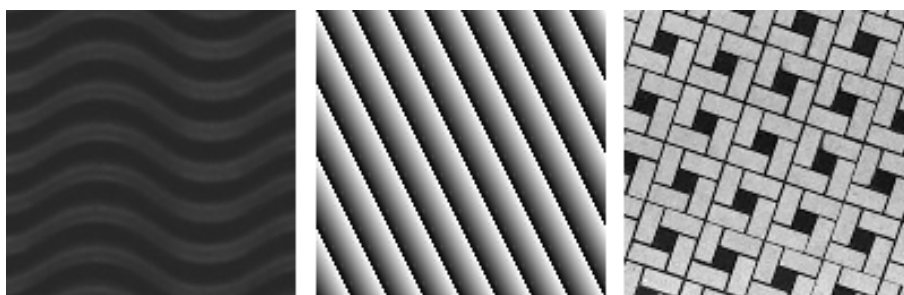
1. Les textures déterministes : elles sont composées d’un unique élément, appelé texton (introduit dans [Julesz, 1975]), qui est répété spatialement de manière régulière se-

2.1. NOTION DE TEXTURE

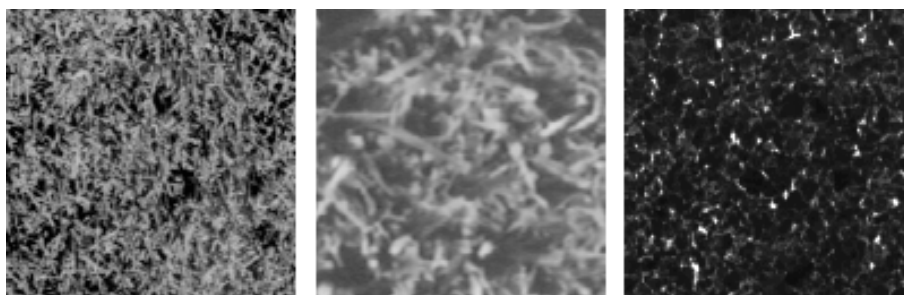
lon une orientation et une période précise. Ce type de texture, bien que facilement caractérisable, est très rare dans les images naturelles, c'est pourquoi les textures déterministes sont majoritairement artificielles.

2. Les textures stochastiques : cette catégorie comprend les textures composées de motifs qui ne peuvent être définis comme réguliers. Celles-ci peuvent être considérées comme des réalisations de champs aléatoires bi-dimensionnels.

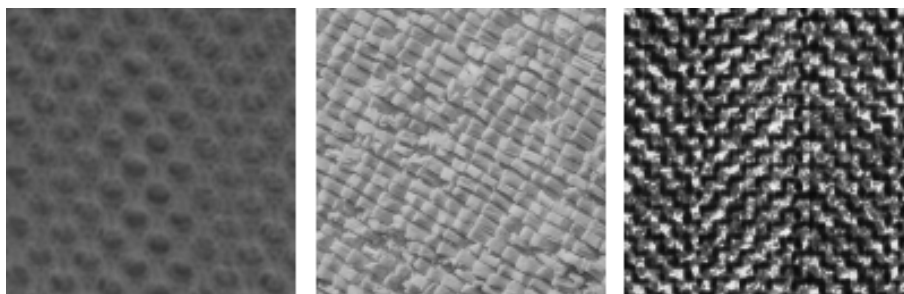
En plus des textures de types déterministes et stochastiques, une troisième catégorie peut être considérée : les textures quasi-déterministes. Ce sont des textures où un motif est visible car il est répété aléatoirement et de manière proche, mais rarement à l'identique, ce qui le rend difficile à isoler. Les textures naturelles font généralement partie de cette catégorie. La figure (2.2) illustre les trois types de textures.



(a) Textures de type déterministe



(b) Textures de type stochastique



(c) Textures de type quasi-déterministe

FIG. 2.2 – Illustration des trois principaux types de textures

Les multiples définitions de la texture expliquent les nombreuses méthodes développées

pour la caractériser. Dans la section suivante, nous présentons une revue des descripteurs de textures les plus couramment utilisés.

2.2 Les descripteurs de textures

Dans [Richards et Polit, 1974] les caractéristiques de textures sont séparées en quatre principaux types : les méthodes statistiques, les méthodes géométriques, les méthodes basées modèle et les méthodes inspirées par le traitement de signaux à une dimension.

2.2.1 Méthodes statistiques

Ces approches sont parmi les plus anciennes dans le domaine de l'analyse de textures. Elles s'appuient sur la répartition spatiale des niveaux de gris dans la texture considérée.

2.2.1.1 Analyse par matrice de cooccurrence

Introduite dans [Haralick *et al.*, 1973], l'analyse par matrice de cooccurrence (ou GLCM : Grey Level Cooccurrence Matrix) part du principe que l'information de texture d'un motif peut être représentée par les transitions au niveau local entre chaque niveau de gris le composant. Les analyses basées sur la GLCM utilisent les propriétés d'analyses statistiques d'ordre deux (donc plus adaptées aux textures car l'analyse est réalisée par paires de pixels), ce qui les a rendues très populaires. De nombreux travaux utilisent encore à l'heure actuelle des analyses basées sur la GLCM [Abdulhady *et al.*, 2002, Smutek *et al.*, 2003, Arvis *et al.*, 2004, Tesar *et al.*, 2007].

La matrice de cooccurrence permet d'obtenir des informations sur les transitions de niveaux de gris dans une image pour une distance et une direction donnée.

Soit I une image composée de X pixels $\mathbf{x}$ et de N niveaux de gris, et $U = (d, \theta)$ un vecteur représentant la distance d et l'angle θ des transitions analysées par la méthode (d et θ sont fixés par l'utilisateur). La taille de la GLCM dépend du nombre de niveaux de gris que l'on souhaite prendre en compte dans les calculs. Afin de la réduire, l'espace des N niveaux de gris de l'image Ω_I est généralement projeté dans un espace à N_g niveaux de gris tel que $N_g < N$. La valeur de N_g déterminera la taille de la matrice et donc sa précision. De grandes valeurs garantiront une grande précision mais entraîneront un coût plus élevé non seulement lors de la construction de la matrice, mais également lors du calcul ultérieur des coefficients d'Haralick. La plupart du temps, $N_g = 16$ ou 32 .

La matrice de cooccurrence P_U de taille $N_g \times N_g$ est donc définie comme l'histogramme 2D des transitions entre les N_g niveaux de gris de l'image Ω_I selon la transition $U(d, \theta)$:

$$P_U(i, j) = \text{Card} \{ \mathbf{x} | (I(\mathbf{x}) = i, I(\mathbf{x} + U(d, \theta)) = j) \} .$$

Ainsi, l'élément $P_U(i, j)$ représente le nombre de fois où la transition $U = (d, \theta)$ permet de passer d'un niveau de gris i vers un niveau de gris j .

P_U est généralement symétrique (les transitions $i \rightarrow j$ et $j \rightarrow i$ sont équivalentes), mais il est également possible d'utiliser une matrice non symétrique permettant de différencier les transitions $i \rightarrow j$ et $j \rightarrow i$.

La matrice de cooccurrence ainsi obtenue est une base pour le calcul de différentes caractéristiques de textures. Parmi les plus courantes peuvent être citées le contraste, permettant d'évaluer les fortes transitions de niveaux de gris, l'homogénéité, caractérisant le nombre de zones homogènes de la texture et la mesure des moments angulaires d'ordre deux qui peut être considérée comme une évaluation de sa granulosité. Les caractéristiques d'Haralick sont calculées à partir de la matrice de cooccurrence ; les plus utilisées sont disponibles en annexe A.

L'analyse de textures à l'aide de la matrice de cooccurrence est très populaire notamment grâce à sa facilité de mise en œuvre. En revanche, les différentes caractéristiques qu'elle fournit n'ont pas toujours le même pouvoir discriminant d'un type de texture à l'autre, ce qui rend généralement indispensable une étape de sélection de caractéristiques en amont. Le choix du vecteur de transition $U = (d, \theta)$ est également important. Dans [Haralick, 1979], il est suggéré de calculer une moyenne sur plusieurs matrices de cooccurrence avec différentes valeurs de U , rendant ainsi les caractéristiques indépendantes à la rotation (et au changement d'échelle dans une certaine mesure).

2.2.1.2 Mesure d'autocorrélation

Une texture est généralement composée de motifs plus ou moins identiques se répétant dans l'espace. La mesure d'autocorrélation permet de prendre en compte cette particularité en évaluant le caractère fin ou grossier de la texture, représenté par la taille de sa primitive.

La représentation à l'aide de la mesure d'autocorrélation d'une image texturée Ω_I de taille $N \times M$ est donnée par :

$$p(x, y) = \frac{M \cdot N}{(M - x) \cdot (N - y)} \frac{\sum_{i=1}^{N-x} \sum_{j=1}^{M-y} I(i, j) \cdot I(i + x, j + y)}{\sum_{i=1}^N \sum_{j=1}^M I^2(i, j)} . \quad (2.1)$$

La fonction d'autocorrélation est donc une mesure spatiale qui va décroître rapidement pour des textures fines, lentement pour des textures grossières, et qui sera périodique pour des textures régulières.

Trois principales approches basées sur la fonction d'autocorrélation permettent de caractériser une texture : l'utiliser directement comme signature de la texture, déterminer des informations basées sur sa périodicité ou encore en extraire des caractéristiques (basées sur le pic central). La figure (2.3) illustre la fonction d'autocorrélation pour différentes textures. La fonction d'autocorrélation peut être étendue à l'ordre N comme :

$$p(a_1, a_2, \dots, a_n) = \int f(r) f(r + a_1) f(r + a_2) \dots f(r + a_n) dr . \quad (2.2)$$

Se basant sur cette équation généralisée, une adaptation locale de la mesure d'autocorrélation d'ordre deux appelée HLAC (*High order Local AutoCorrelation features*) est proposée dans [Otsu et Kurita, 1988]. Plus récemment dans [Toyoda et Hasegawa, 2007], la méthode du HLAC a été étendue à l'ordre huit dans le cadre de la reconnaissance de visages.

2.2. LES DESCRIPTEURS DE TEXTURES

L'analyse de textures basée sur la mesure d'autocorrélation reste efficace et toujours d'actualité [Yamamoto *et al.*, 2006, Bergonnier *et al.*, 2007], mais elle nécessite des temps de calculs importants.

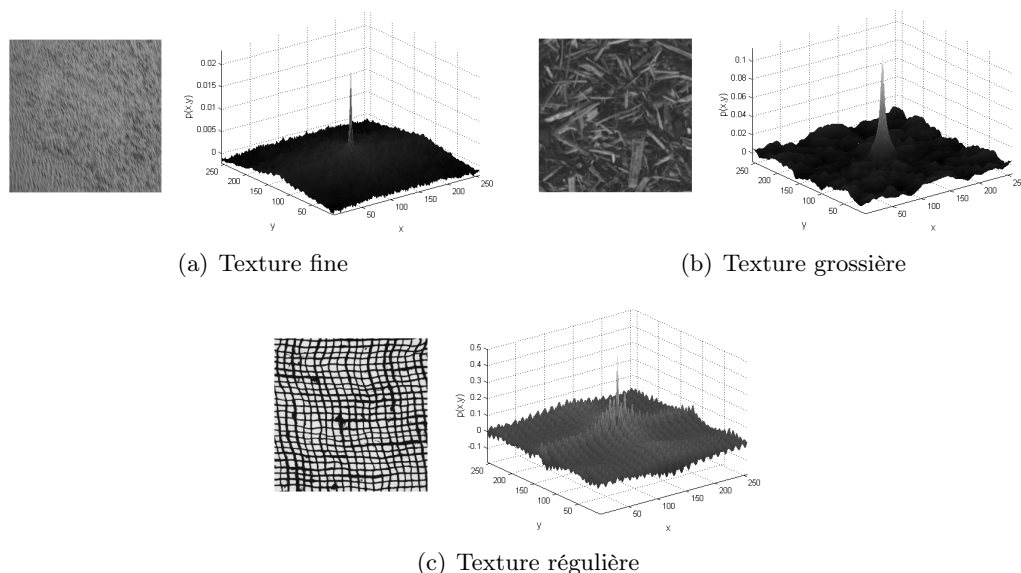


FIG. 2.3 – Mesure d'autocorrélation pour trois types de textures. a) Texture fine : la fonction décroît rapidement depuis le centre. b) Texture grossière : la fonction décroît lentement. c) Texture régulière : la fonction devient périodique et fait apparaître une forme de type "pics et vallées".

2.2.1.3 Modèle binaire local (*Local Binary Pattern*)

Initialement développée dans [Ojala *et al.*, 1996], la caractérisation de textures par modèle binaire local est inspirée du modèle de spectre de texture de Wang [Wang et He, 1990] et reste d'actualité dans de nombreuses applications [Nikam et Agarwal, 2008, Singh et Noore, 2008, Heikkila *et al.*, 2009]. Le principe de cette approche est de caractériser chaque pixel de l'image en fonction de la valeur du niveau de gris de ses voisins.

Soit $\mathbf{x}$ un pixel et $N(\mathbf{x}) = \{N_1, N_2, \dots, N_n\}$ l'ensemble composé de ces n voisins. Une valeur binaire B_i est affectée à chaque voisin $n_i \in N$ telle que :

$$B_i = \begin{cases} 1 & \text{si } I(n_i) > I(\mathbf{x}) \\ 0 & \text{sinon} \end{cases} . \quad (2.3)$$

Les B_i sont alors pondérées par une puissance de deux et sommées de manière à obtenir le code LBP du pixel $\mathbf{x}$.

$$LBP(\mathbf{x}) = \sum_{i=1}^n B_i 2^{i-1} . \quad (2.4)$$

2.2. LES DESCRIPTEURS DE TEXTURES

La figure (2.4) illustre le principe de la méthode et donne une valeur LBP du pixel traité égale à $0 \times 1 + 0 \times 2 + 1 \times 4 + 1 \times 8 + 1 \times 16 + 0 \times 32 + 0 \times 64 + 0 \times 128 = 28$.

Cette approche permet de caractériser chaque pixel de l'image de manière unique. Dans [Ojala *et al.*, 2002], il est suggéré d'utiliser un voisinage circulaire autour de chaque pixel, le rayon du cercle et le nombre de voisins considérés étant fixés. Une technique permettant de rendre la caractérisation par modèle binaire local invariante à la rotation est également proposée.

Cette méthode de caractérisation de textures, bien que limitée car basée sur un unique coefficient, bénéficie d'une très faible sensibilité aux variations de luminosité et permet, de part sa faible complexité, d'être intégrée dans des processus temps réel [Choi et Lam, 2008, Li *et al.*, 2008].

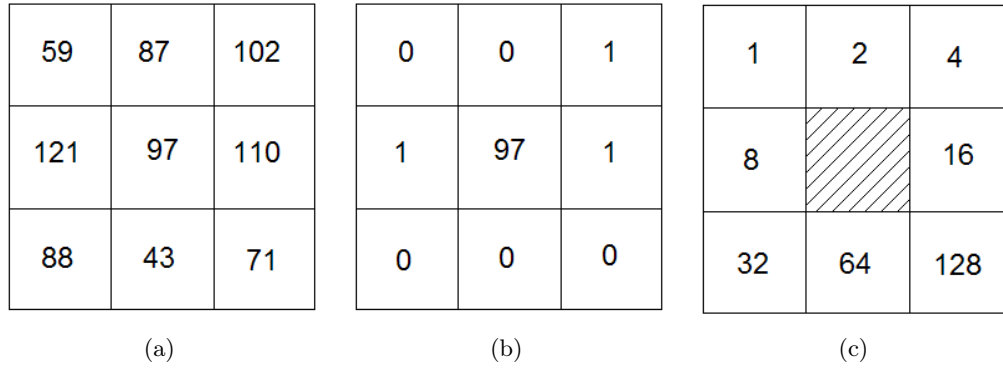


FIG. 2.4 – Illustration du principe de la caractérisation de textures par modèle binaire local pour un voisinage carré de largeur 3. a) Le pixel à caractériser (valeur de 97) est au centre de son voisinage. b) Chaque voisin de valeur supérieure à celle du pixel traité prend la valeur 1, les autres prennent la valeur 0. c) Répartition de la pondération par une puissance 2 de chaque voisin.

2.2.2 Méthodes géométriques

Ces méthodes s'appuient sur la définition de la texture comme la répétition dans l'espace d'un ou plusieurs motifs élémentaires et tentent de définir les caractéristiques de la primitive de la texture.

2.2.2.1 Analyse par diagramme de Voronoï

Le but de cette approche est d'isoler les différents motifs élémentaires de la texture et de les caractériser. L'extraction des motifs élémentaires est réalisée en utilisant le concept de diagramme de Voronoï [Shamos et Hoey, 1975, Chassery et Melkemi, 1991]. Afin d'en définir le principe, considérons S un ensemble de germes dans une image Ω_I et $P, Q \in S$ deux germes (un germe peut être défini comme un pixel isolé ou comme un regroupement de pixels voisins). La bissectrice du segment reliant P et Q définit deux demi-plans H_P^Q et H_Q^P , le premier contenant P et le deuxième Q . L'intersection de l'ensemble des demi-plans

contenant P est définie par :

$$\cap_{Q \in S, Q \neq P} H_P^Q. \quad (2.5)$$

Elle détermine une région polygonale appelée polygone de Voronoï, contenant les points de l'espace Euclidien plus proche de P que de n'importe quel autre point. L'ensemble des polygones de Voronoï définit $\forall Q \in S$ correspond alors à un partitionnement de l'image.

Dans [Tuceryan et Jain, 1990], le principe des polygones de Voronoï est appliqué à la segmentation d'images texturées synthétiques et une méthode en trois étapes est définie afin d'obtenir les germes d'une texture qui seront utilisés pour construire le pavage.

1. Appliquer un filtre *Laplacian of Gaussian* (LoG) sur l'image. Ce dernier peut être approximé par un filtre *Difference of Gaussians* (DoG) afin d'améliorer le temps d'exécution.
2. Détecter les maxima locaux dans l'image filtrée. Un pixel est considéré comme un maximum local si sa valeur est supérieure à au moins six de ses huit voisins. L'image est alors binarisée pour ne garder que les maxima locaux.
3. Réaliser une analyse des composantes connexes sur l'image binarisée en utilisant les huit plus proches voisins. Chaque composante connexe est alors définie comme élément structurant de la texture et devient un germe pour le pavage de Voronoï.

Les polygones de Voronoï possédant des caractéristiques similaires sont alors réunis pour former des régions texturées uniformes. Les principales caractéristiques utilisées sont les moments d'ordre 0 et diverses mesures sur la géométrie du polygone (voir [Tuceryan et Jain, 1990] pour plus de détails).

L'analyse de textures par diagramme de Voronoï peut être considérée comme un cas particulier car elle permet surtout de regrouper en régions uniformes différentes zones d'une image texturée. Le point fort de cette approche est la construction des différentes zones initiales, qui s'appuient chacune sur des polygones de Voronoï. En revanche, elle reste délicate à utiliser dans le cadre de la caractérisation de textures. En effet, bien que très intéressante, l'analyse de textures par diagramme de Voronoï est surtout une méthode originale de segmentation d'images texturées utilisant des descripteurs connus, plutôt qu'une réelle méthode de caractérisation de textures. Il pourrait toutefois être intéressant d'étudier si le nombre et la disposition des zones homogènes obtenues par cette approche peuvent aboutir à un descripteur de textures efficace.

2.2.2.2 Méthodes structurelles

Si les méthodes structurelles reprennent l'idée de base des méthodes géométriques (une texture est composée de la répétition dans l'espace d'un motif élémentaire), elles cherchent également à identifier les règles de placement des primitives qui ont permis de générer la texture. L'analyse d'une texture à l'aide d'une méthode structurelle est généralement composée de deux étapes : d'abord le motif élémentaire de la texture est identifié, puis les règles de placement ayant conduit à la texture étudiée sont déterminées.

Bien qu'il existe plusieurs méthodes structurelles intéressantes [Zucker, 1976, Voorhees et Poggio, 1987, Blostein et Ahuja, 1989], cette approche reste limitée car elle n'est réellement efficace que pour le traitement de textures très régulières. C'est pourquoi cette famille de méthodes de caractérisation de textures ne sera pas développée.

2.2.3 Méthodes basées modèle

2.2.3.1 Champs de Markov aléatoires

Les champs de Markov font partie de la théorie probabiliste d'analyse des dépendances spatiales et contextuelles des phénomènes physiques. La modélisation d'une image comme un champ de Markov aléatoire est très populaire dans le domaine de la vision par ordinateur et de l'analyse de textures, celles-ci étant généralement régies pas des règles de dépendances spatiales [Cross et Jain, 1983, Ruan *et al.*, 2002, Fjortoft *et al.*, 2003, Jedynak *et al.*, 2005, Stewart *et al.*, 2008]. Le lecteur pourra se référer à trois ouvrages très complets sur la théorie des champs de Markov et leur application à l'analyse d'images [Li, 1995, Li, 2001, Pieczynski, 2003].

Le principe de base des champs de Markov est de considérer que l'intensité de chaque pixel d'une image peut être déterminée uniquement par l'étude de ses voisins.

Avant de décrire plus en détails les champs de Markov, la notion de voisinage doit être précisée. Une image de taille N peut être représentée comme un ensemble de sites $S = \{s_1...s_N\}$ muni d'un système de voisinage $V_s, s \in S$ tel que :

$$s \notin V_s \quad (2.6)$$

$$\forall s, t \in S, s \in V_t \Leftrightarrow t \in V_s. \quad (2.7)$$

La figure (2.5) illustre les configurations de voisinage les plus souvent utilisées.

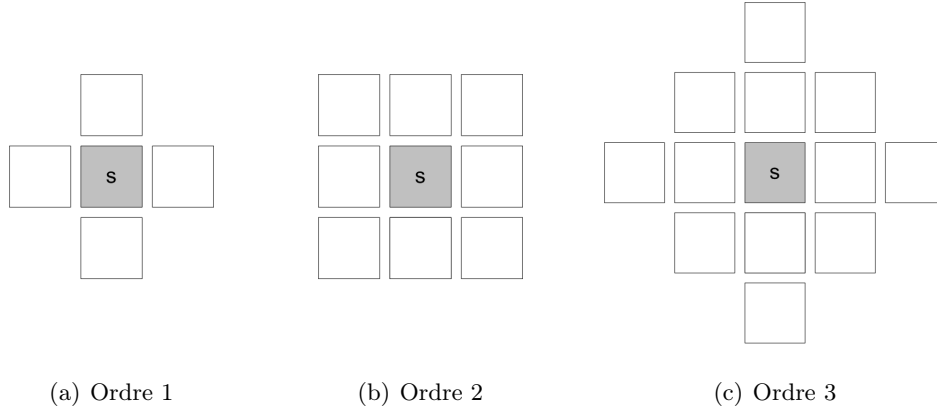


FIG. 2.5 – Trois premiers ordres de voisinage pour un site $s \in S$.

La réalisation des niveaux de gris d'une image peut être perçue comme un ensemble de variables aléatoires $X = \{X_1, ..., X_N\}, X_s \in S$. Chaque variable X_s prend ses valeurs x_s dans un ensemble d'états Ω avec une probabilité $P(X_s = x_s)$. Une réalisation complète de l'ensemble de variables aléatoires d'un champ sera notée comme l'évènement $X = x$, $x \in \Omega^N$, et aura une probabilité $P(X = x)$ d'être réalisée. Dans le cas d'une image en niveaux de gris, $Card(\Omega) = 256$.

Le champ (X, Ω^N, P) est considéré comme un champ de Markov par rapport au système

de voisinage V_s s'il satisfait :

$$\forall x \in \Omega^N, \forall s \in S : \quad (2.8)$$

$$P(X = x) > 0$$

$$P(X_s = x_s | X_r = x_r, r \in S, r \neq s) = P(X_s = x_s | X_r = x_r, r \in V_s) . \quad (2.9)$$

La première propriété entraîne une notion de positivité sur les variables aléatoires. La deuxième propriété (propriété Markovienne) spécifie que la valeur d'une variable aléatoire X_s n'est dépendante que des valeurs des variables appartenant à son voisinage V_s .

Il existe deux approches pour définir un modèle de champ de Markov : déterminer la probabilité conditionnelle $P(X_s = x_s | X_r = x_r, r \in V_s)$ ou la probabilité jointe $P(X = x)$ (probabilité de réalisation complète du champ X). D'après [Besag, 1974], exprimer un champ de Markov en terme de probabilité jointe est plus intéressant. Malheureusement, il n'existe pas de méthode simple pour la déduire de $P(X_s = x_s | X_r = x_r, r \in V_s)$. Il est néanmoins possible de déterminer la probabilité jointe d'un champ de Markov en utilisant les propriétés des champs de Gibbs.

Définissons dans un premier temps la notion de champ de Gibbs en utilisant les notations précédentes. Un champ aléatoire (X, Ω^N, P) peut être considéré comme un champ de Gibbs si sa probabilité jointe est de la forme :

$$P(X) = \frac{\exp(-\mathcal{U}(X))}{Z} , \quad (2.10)$$

avec $Z = \sum_{X \in \Omega^N} \exp(-\mathcal{U}(X))$ une constante de normalisation appelée fonction de partition.

L'énergie globale $\mathcal{U}(x)$ peut être décomposée en la somme des potentiels locaux :

$$\mathcal{U}(x) = \sum_{c \in C} V_c(x) . \quad (2.11)$$

C représente l'ensemble des cliques du système de voisinage. Les cliques d'ordre un sont les sites singletons, les cliques d'ordre deux sont les sites voisins deux à deux, les cliques d'ordre trois sont les sites voisins trois à trois, etc. La fonction de potentiel V_c est propre au modèle de champ choisi.

Le théorème d'Hammersley-Clifford fait un parallèle entre champs de Markov et champs de Gibbs, permettant ainsi aux champs de Markov de bénéficier des propriétés de décomposition de l'énergie globale [Hammersley et Clifford, 1971, Besag, 1974].

Théorème 1 *Un champ aléatoire X est un champ de Markov par rapport au système de voisinage V si et seulement si X est un champ de Gibbs par rapport à V .*

Cela signifie que la probabilité jointe d'un champ de Markov X s'écrit sous la forme d'une distribution de Gibbs similaire à l'équation (2.10).

Pour comprendre de quelle manière les champs de Markov s'utilisent dans une problématique d'analyse d'images, il est plus aisé de s'appuyer sur leurs différentes applications. Il existe deux principaux types d'application des champs de Markov en traitement d'images et plus particulièrement dans le domaine de l'analyse de textures : l'analyse ou la synthèse de textures, et la segmentation ou la restauration d'images texturées.

Caractérisation et synthèse de textures

Dans ce premier domaine d'application des champs de Markov, la texture est considérée comme la réalisation d'un champ de Markov. Dans le cas de la caractérisation d'une texture, les paramètres du champ seront estimés et considérés comme caractéristiques de la texture étudiée. Dans le cas de la synthèse d'une texture, on cherchera à générer une texture artificielle correspondant à un champ de Markov de paramètres connus.

Un modèle de champ de Markov populaire pour caractériser et synthétiser une texture est le modèle autobinominal [Cross et Jain, 1983], très adapté pour décrire des variables discrètes telles que les niveaux de gris d'une image. La probabilité de réalisation de X suit alors une loi binomiale telle que :

$$P(X_s = x_s | X_r = x_r, r \in V_s) = C_{F-1}^{x_s} \theta^{x_s} (1 - \theta)^{(F-1)-x_s} . \quad (2.12)$$

La probabilité recherchée $P(X_s = x_s)$ correspond donc à une loi binomiale où l'on cherche le nombre de fois où la variable X_s prendra la valeur x_s sur $(F - 1)$ tirages (dans le cas d'une image en niveaux de gris, $F = 256$), sachant que la probabilité θ de réalisation de l'évènement $X_s = x_s$ pour chaque tirage s'exprime comme :

$$\theta = \frac{e^\psi}{1 + e^\psi} , \quad (2.13)$$

où ψ une fonction relative au voisinage V_s .

Dans le cas d'un voisinage d'ordre 1 : $\psi = a + b_1(x_{i,j-1} + x_{i,j+1}) + b_2(x_{i-1,j} + x_{i+1,j})$. Le coefficient a représente alors la moyenne des intensités de niveaux de gris dans le voisinage, les coefficients b_1 et b_2 les homogénéités respectivement verticales et horizontales.

Dans le cas d'un voisinage d'ordre 2 : $\psi = a + b_1(x_{i,j-1} + x_{i,j+1}) + b_2(x_{i-1,j} + x_{i+1,j}) + b_3(x_{i-1,j-1} + x_{i+1,j+1}) + b_4(x_{i+1,j-1} + x_{i-1,j+1})$. Les coefficients b_3 et b_4 représentent alors les homogénéités suivant les deux directions diagonales.

Dans une problématique de caractérisation de textures, les paramètres du champ X sont généralement estimés en utilisant le principe du maximum de vraisemblance ou les moindres carrés. Dans le cadre de la synthèse de textures, le but du modèle est de générer artificiellement une texture dont les paramètres sont connus. Une fois le modèle de champ, le type de voisinage et les paramètres fixés, un algorithme d'échantillonnage est utilisé pour générer un champ aléatoire respectant la loi de probabilité du modèle choisi. L'algorithme de Metropolis [Metropolis *et al.*, 1953] et l'échantillonneur de Gibbs [Geman et Geman, 1984] sont les deux algorithmes les plus utilisés.

Segmentation, reconstruction d'images

Ici, nous nous plaçons dans le cas où l'on considère les données comme manquantes. Si les données sont partiellement manquantes, il s'agit de reconstruire le modèle original. Si les données sont complètement manquantes, il s'agit d'une segmentation (les données sont alors des étiquettes de segmentation).

Les données (i.e. l'image d'origine) sont considérées comme la réalisation d'un premier champ de Markov Y prenant ses valeurs dans Ω . Est alors recherchée une réalisation d'un champ de Markov caché X , correspondant à l'image segmentée ou restaurée. X est le champ des étiquettes dans le cas d'une segmentation (chaque étiquette représentant alors un type de texture recherché), et le champ des intensités dans la cas de la restauration. Y est alors considéré comme une version dégradée du champ X et le but de la restauration ou de la segmentation va être de définir le processus inverse permettant de déterminer X sachant Y .

Dans un cadre Bayésien, le processus est généralement traité comme un problème d'optimisation de type Maximum *A Posteriori* (MAP). La réalisation du champ X maximisant l'expression de la probabilité *a posteriori* $P(X|Y)$ est donc recherchée. Pour cela, en se basant sur l'expression de la probabilité *a priori* $P(X)$ et la propriété conditionnelle $P(Y|X)$ (appelée également vraisemblance), l'expression de $P(X|Y)$ est définie comme :

$$P(X|Y) \approx P(Y|X)P(X) . \quad (2.14)$$

La réalisation du champ X (i.e. l'image restaurée ou segmentée) permettant de maximiser la probabilité $P(Y|X)P(X)$ doit donc être déterminée. L'estimation MAP de X est donnée par :

$$\hat{X}_{MAP} = \arg \max_X \{P(X|Y)\} \quad (2.15)$$

$$= \arg \max_X \{P(Y|X)P(X)\} . \quad (2.16)$$

La réalisation du champ X peut également être estimée en terme d'énergies associées aux champs :

$$\hat{X}_{MAP} = \arg \min_X \{U(X|Y)\} \quad (2.17)$$

$$= \arg \min_X \{U(Y|X) + U(X)\} . \quad (2.18)$$

Afin de déterminer la réalisation de X qui maximisera la probabilité *a posteriori*, il est donc nécessaire de définir l'énergie *a priori* $U(X)$ et l'énergie de vraisemblance $U(Y|X)$.

Les propriétés des champs de Gibbs permettent de définir la probabilité *a priori* $P(X)$ du modèle telle que :

$$P(X) = \frac{\exp(-U(X))}{Z} . \quad (2.19)$$

Un modèle d'*a priori* se caractérise par la définition d'un potentiel de clique et d'un voisinage. Trois modèles sont principalement utilisés.

- Le modèle d'Ising [Ising, 1925] est défini pour un espace des étiquettes binaires ($L = \{-1, 1\}$) avec un voisinage d'ordre un ou deux et des cliques d'ordre un et deux également :

$$U(X) = -\beta \sum_{s,t \in C_2} x_s x_t - B \sum_{s \in S} x_s ,$$

avec C_2 l'ensemble des cliques d'ordre deux, B une constante représentant un champ

magnétique externe et β un paramètre d'interaction (β positif favorisera les énergies faibles, donc les voisins de même signe). De part ses limitations sur l'ensemble des étiquettes, le modèle d'Ising n'est applicable que dans des problématiques de segmentation binaire.

- Le modèle de Potts [Besag, 1986] est une extension du modèle d'Ising. Il définit l'énergie *a priori* pour un ensemble d'étiquettes $L = \{0, \dots, M - 1\}$, un voisinage d'ordre un ou deux et des cliques uniquement d'ordre deux :

$$U(X) = - \sum_{s,t \in C_2} \beta \delta(x(s), x(t))$$

$$\text{avec } \delta(i, j) = \begin{cases} -1 & \text{si } i = j \\ +1 & \text{si } i \neq j \end{cases}, \beta > 0.$$

Plus β est grand et plus les configurations possédant deux étiquettes voisines différentes seront pénalisées.

- Le modèle gaussien est défini pour un ensemble d'étiquettes $L = \{0, \dots, 255\}$, un voisinage d'ordre un ou deux et des cliques d'ordre un et deux :

$$U(X) = \beta \sum_{s,t \in C_2} (x(s) - x(t))^2 + \alpha \sum_{s \in S} (x(s) - \mu_s)^2.$$

Le premier terme est un terme de régularisation sur les cliques d'ordre deux, le deuxième terme est considéré comme un terme d'attache aux données. α et β servent à pondérer l'influence de chaque terme. De part ses limitations sur l'ensemble des étiquettes, le modèle gaussien n'est applicable que dans une problématique de restauration d'images.

Une fois le modèle d'*a priori* choisi, l'expression de la vraisemblance doit être déterminée. Cette dernière est généralement définie sous l'hypothèse d'indépendance conditionnelle :

$$P(Y|X) = \prod P(y_s|x_s). \quad (2.20)$$

Généralement, les classes contenues dans l'image segmentée sont considérées comme gaussiennes de paramètres (μ_m, σ_m) , avec $m \in \{0, \dots, M - 1\}$, ce qui définit :

$$P(y_s|x_s = m) = \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp^{-U(Y|X)}$$

$$U(Y|X) = \frac{(y_s - \mu_m)^2}{2\sigma_m^2}. \quad (2.21)$$

Une fois le modèle d'*a priori* et l'énergie de vraisemblance déterminés, la configuration maximisant la probabilité *a posteriori* $P(X|Y)$ peut donc être recherchée. Un algorithme de type recuit simulé permettra de réaliser une optimisation globale, et un algorithme de type mode conditionnel itératif (ICM : *Iterative Conditional Mode*) [Bouman et Liu, 1991] déterminera un optimum local. Des algorithmes issus de la théorie des méthodes Monte Carlo par chaînes de Markov (MCMC) devront alors être utilisés pour simuler la loi *a posteriori*, tel que l'algorithme de Metropolis-Hasting [Metropolis *et al.*, 1953] ou encore l'échantillonneur de Gibbs [Geman et Geman, 1984].

2.2.3.2 Approche par fractales

La dimension fractale introduite dans [Mandelbrot, 1977, Mandelbrot, 1983] peut être considérée comme un indicateur permettant de déterminer dans quelle mesure un objet occupe l'espace Euclidien. Cette mesure est définie pour des formes artificielles (voir l'exemple de la courbe de Koch dans [Mandelbrot, 1977]) mais ne peut être qu'estimée pour des textures naturelles. Les nombreuses méthodes développées dans ce but peuvent être divisées en deux principales familles : les méthodes géométriques et les méthodes stochastiques.

Parmi les principales méthodes géométriques peuvent être citées les approches *Planar triangles* [Clarke, 1986], *Covering blanket* [Peleg et al., 1984], *Flat structuring element* [Dubuc et al., 1989a] ou encore la méthode de *Box counting* [Keller et al., 1989].

Les méthodes stochastiques s'appuient sur la théorie définissant la plupart des fractales présentes dans les phénomènes physiques comme des fonctions Browniennes fractales (*fractal Brownian functions*, fBf) [Pentland, 1984]. L'estimation de la dimension fractale à l'aide de méthodes stochastiques se fait principalement avec les variogrammes [Dubuc et al., 1989b] ou l'analyse par spectre de puissance basée sur la transformée de Fourier de l'image [Pentland, 1984].

Un état de l'art très complet, ainsi qu'une comparaison des différentes méthodes d'estimation peuvent être trouvés dans [Soille et Rivest, 1996].

La technique de *Box-Counting* est l'une des méthodes géométriques les plus simples à mettre en œuvre. L'image est transformée en un quadrillage de carrés élémentaires (*boxes*) de taille ϵ . N_ϵ , le nombre de carrés nécessaires à l'approximation de la texture, c'est-à-dire recouvrant au moins un élément de la texture, est alors déterminé. La zone recouverte par les N_ϵ devient plus précise à mesure que ϵ diminue. Ainsi, la *box-dimension*, estimateur de la dimension fractale de la zone considérée X est donnée par :

$$D(X) = \lim_{\epsilon \rightarrow 0} \frac{\log(N_\epsilon)(X)}{\log \epsilon} . \quad (2.22)$$

Afin de déterminer cette limite, plusieurs estimations sont réalisées pour différentes valeurs de ϵ . La dimension fractale est alors équivalente à la pente d'une régression linéaire de la fonction $\log(D(X_\epsilon)) = f(\log(\epsilon))$.

L'estimation de la dimension fractale ne permet pas une discrimination précise. En effet, deux textures différentes peuvent avoir la même valeur de $D(X)$. Ainsi, une estimation de la mesure de lacunarité, permettant une discrimination plus efficace, est proposée dans [Keller et al., 1989]. La lacunarité est définie comme :

$$\Lambda = \frac{E[M^2(\epsilon)] - E^2[M(\epsilon)]}{E^2[M(\epsilon)]} . \quad (2.23)$$

E est l'espérance mathématique, M est la masse fractale définie par $M(\epsilon) = \sum_{m=1}^{\epsilon^2} m PR_{G(f)}(m, \epsilon)$, avec $PR_{G(f)}(m, \epsilon)$ la probabilité estimée de trouver un point de la fractale $G(f)$ à l'intérieur d'un carré de taille ϵ .

2.2.4 Méthodes empruntées au traitement du signal

Les méthodes d'analyse fréquentielle considèrent qu'une texture peut être caractérisée par l'analyse des répétitions d'un ou plusieurs motifs selon différentes fréquences spatiales. Ainsi, de nombreuses méthodes initialement développées pour l'analyse fréquentielle des signaux à une dimension ont été adaptées aux deux dimensions de l'image.

2.2.4.1 Filtrage de Fourier

Lorsque les variations d'un signal sont analysées au niveau fréquentiel, il est plus efficace de passer dans l'espace de Fourier. En effet, l'intensité du signal n'y est plus représentée en fonction du temps, mais en fonction des différentes fréquences le composant. Ainsi, il devient plus aisé de déterminer les fréquences les plus utiles et celles ne détenant que peu d'information.

Dans le cadre du traitement des signaux à une dimension (comme du son par exemple), l'expression d'un signal $f(x)$ dans le domaine de Fourier est donnée par sa transformée $F(u)$ telle que :

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi ux} dx . \quad (2.24)$$

$F(u)$ donne alors l'intensité du signal f pour la fréquence u . La transformée de Fourier peut être étendue aux signaux à deux dimensions comme les images :

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)e^{-j2\pi(ux+vy)} dx dy . \quad (2.25)$$

L'analyse d'un signal dans le domaine de Fourier donne ainsi la contribution de chaque fréquence, mais cela pour la durée totale du signal. Il est parfois intéressant de ne connaître la contribution de chaque fréquence que de manière localisée dans l'espace (ou le temps). C'est le principe de la transformée de Fourier fenêtrée, donnée pour un signal à une dimension par :

$$F(u, \epsilon) = \int_{-\infty}^{\infty} f(x)w(x - \epsilon)e^{-j2\pi ux} dx , \quad (2.26)$$

avec $w(x - \epsilon)$ une fonction permettant de ne transformer le signal $f(x)$ que sur une durée relative à ϵ . Nous verrons par la suite que la transformée de Fourier fenêtrée est à la base de l'application des filtres de Gabor.

L'analyse de textures à l'aide de la transformée de Fourier a donné lieu à de nombreux travaux en classification [Azencott *et al.*, 1997, Luo et Mirchandani, 2000], en segmentation [Hsu *et al.*, 2000] ou en caractérisation de textures [Hong *et al.*, 1999].

2.2.4.2 Filtrage de Gabor

Initialement développés dans [Gabor, 1946], Daugman identifia la capacité des filtres de Gabor à approximer les cellules du cortex de certains mammifères dans [Daugman, 1980]. Leur application à l'analyse de textures fut initiée dans [Turner, 1986, Clark *et al.*, 1987]. L'analyse de textures par filtre de Gabor reste à l'heure actuelle un domaine de recherche

2.2. LES DESCRIPTEURS DE TEXTURES

riche en contributions [Autio et Elomaa, 2003, Rousseau *et al.*, 2003, Samir *et al.*, 2007, Li et Staunton, 2008].

Le filtrage de Gabor est un cas particulier de la transformée de Fourier fenêtrée, où l'on cherche à observer un signal à travers une fenêtre gaussienne afin de mettre en évidence certaines fréquences de sa transformée de Fourier.

Dans le domaine spatial, la réponse d'un filtre de Gabor est donc similaire à celle des filtres utilisés pour la transformée de Fourier fenêtrée :

$$g(x, y) = s(x, y)w(x, y) , \quad (2.27)$$

où $s(x, y)$ est une sinusoïde complexe et $w(x, y)$ l'enveloppe d'une gaussienne. $s(x, y)$ est définie par :

$$s(x, y) = \exp(j(2\pi(u_0x + v_0y) + P)) . \quad (2.28)$$

$s(x, y)$ peut également être représentée par ses parties réelles et imaginaires :

$$\begin{aligned} \operatorname{Re}(s(x, y)) &= \cos(2\pi(u_0x + v_0y) + P) \\ \operatorname{Im}(s(x, y)) &= \sin(2\pi(u_0x + v_0y) + P) . \end{aligned} \quad (2.29)$$

(u_0, v_0) sont les paramètres fréquentiels de la sinusoïde en coordonnées cartésiennes et P définit sa phase. $s(x, y)$ peut-être exprimée en coordonnées polaires en définissant une magnitude F_0 et un angle w_0 tels que $u_0 = F_0 \cos w_0$ et $v_0 = F_0 \sin w_0$. On obtient :

$$s(x, y) = \exp(j(2\pi F_0(x \cos w_0 + y \sin w_0) + P)) . \quad (2.30)$$

L'enveloppe gaussienne $w_r(x, y)$ est définie par :

$$w(x, y) = K \exp\left(-\pi \frac{((x - x_0)\cos\theta + (y - y_0)\sin\theta)^2}{\sigma_x^2} + \frac{((y - y_0)\cos\theta - (x - x_0)\sin\theta)^2}{\sigma_y^2}\right) . \quad (2.31)$$

K détermine l'amplitude de la gaussienne, (x_0, y_0) les coordonnées de son pic, (σ_x, σ_y) la valeur de ses écarts types (longueur et largeur de la gaussienne) et θ l'angle de rotation qui lui est appliqué.

L'expression complète du filtre de Gabor, en coordonnées polaires, dans le domaine spatial est donc :

$$\begin{aligned} g(x, y) &= \exp(j(2\pi F_0(x \cos w_0 + y \sin w_0) + P)) \\ &\cdot \\ &K \exp\left(-\pi \frac{((x - x_0)\cos\theta + (y - y_0)\sin\theta)^2}{\sigma_x^2} + \frac{((y - y_0)\cos\theta - (x - x_0)\sin\theta)^2}{\sigma_y^2}\right) . \end{aligned} \quad (2.32)$$

La manière dont les nombreux paramètres du filtre déterminent le domaine fréquentiel qui sera observé devient plus intuitive en représentant $g(x, y)$ dans le domaine de Fourier (figure (2.6)). Les paramètres de la sinusoïde déterminent la position du centre de la gaussienne (voir équation (2.30)). La forme et la position de cette dernière sont alors définies par ses nombreux paramètres (voir équation (2.31)). Généralement, la sinusoïde et l'enveloppe gaussienne sont orientées dans la même direction ($\theta = w$).

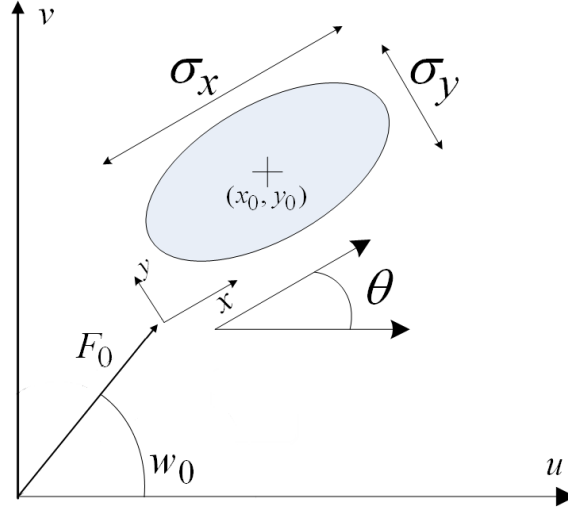


FIG. 2.6 – Illustration de l’influence des différents paramètres du filtre de Gabor sur le domaine fréquentiel observé.

En termes d’application à l’analyse de textures, un filtre de Gabor sera représenté par un masque de taille $N \times M$. La réponse de ce filtre sera alors déterminée par sa convolution avec chaque pixel de l’image. La gaussienne étant généralement définie centrée (i.e. $x_0 = y_0 = 0$), les valeurs du masque sont déterminées à l’aide de l’équation (2.32) telles que $x \in [-N/2, N/2]$ et $y \in [-M/2, M/2]$. La taille du masque doit être choisie relativement à la taille de la gaussienne, de manière à refléter au mieux son comportement.

Malgré une efficacité reconnue, l’analyse de textures à l’aide des filtres de Gabor souffre du nombre élevé de paramètres nécessaires à la création du filtre. En effet, un banc de filtres de Gabor avec différents paramètres est généralement créé de manière à caractériser au mieux la texture. Cela rend nécessaire une phase de sélection de caractéristiques, afin de déterminer les filtres les plus discriminants. Plusieurs travaux traitent du meilleur choix de paramètres pour les filtres de Gabor. Ainsi dans [Clausi et Jernigan, 2000], plusieurs stratégies sont proposées pour extraire les caractéristiques des réponses de différents filtres de Gabor. Une évaluation de l’influence des paramètres sur le pouvoir discriminant de chaque filtre est également proposée dans [Bianconi et Fernández, 2007].

2.2.4.3 Transformée en ondelettes

L’analyse d’images à l’aide des coefficients d’ondelettes d’un signal est devenue très populaire [Waku et Chassery, 1993, Guarget-Duport *et al.*, 1996, Núñez, 2000, Bashar *et al.*, 2003, Kokare *et al.*, 2004, Kim et Kang, 2007, Qiao *et al.*, 2007]. Basés sur l’étude sismologique, les premiers travaux aboutissant à la transformée en ondelettes continue furent initiés dans [Grossman et Morlet, 1984]. Ils furent ensuite développés dans [Mallat, 1989].

Le principe de la théorie des ondelettes est qu’un signal $f(x)$ peut être décomposé en un ensemble de sous-signaux de fréquences inférieures et de leurs coefficients attachés.

2.2. LES DESCRIPTEURS DE TEXTURES

Une ondelette est définie comme une fonction $\Psi(x)$ (ondelette mère), de moyenne nulle, permettant de générer une famille de sous-fonctions dilatées et translatées $\Psi(a, b)(t)$ telles que :

$$\Psi(a, b)(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{x-b}{a}\right), \quad (2.33)$$

avec a et b les facteurs respectivement de dilatation et de translation par rapport à l'ondelette mère $\Psi(x)$. Dans le domaine continu, la transformée en ondelettes d'un signal à une dimension $f(x)$ est donnée par :

$$Wf(a, b) = \int_{-\infty}^{\infty} f(x) \frac{1}{\sqrt{a}} \bar{\Psi}_{a,b}(t) dx, \quad (2.34)$$

avec $\bar{\Psi}(a, b)(x)$ le complexe conjugué de $\Psi(a, b)(t)$.

La famille de coefficients $Wf(a, b)$ fournit une représentation alternative de $f(x)$. Les coefficients d'ondelettes peuvent alors être utilisés pour reconstruire le signal $f(x)$ en suivant :

$$f(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{\sqrt{a}} Wf(a, b) \Psi\left(\frac{x-b}{a}\right) da db. \quad (2.35)$$

La représentation d'un signal en ondelettes continues reste redondante et la reconstruction du signal est irréalisable en pratique à cause des intégrales sur les domaines infinis. Une manière de pallier ces inconvénients est de réaliser la transformée en ondelettes dans le domaine discret (*Discrete Wavelet Transform*, DWT) : a et b deviennent des variables d'échelle et de temps discrètes et les différentes intégrales deviennent alors des sommes discrètes. L'algorithme DWT est appliqué à l'analyse multi-résolution dans [Mallat, 1989].

Soient une ondelette mère $\Psi(a, b)(x)$ et sa fonction de changement d'échelle $\phi(x)$. $\Psi(a, b)(x)$ et $\phi(x)$ sont utilisées pour créer deux filtres qui seront appliqués sur le signal : un filtre passe-bas H et un filtre passe-haut G , son filtre miroir (i.e. $g(n) = -1^{(1-n)}h(1-n)$). Considérant $a_0[n]$ comme le signal d'origine discret, à chaque étape j le signal $a_{j-1}[n]$ est décomposé en deux sous-signaux de résolution deux fois inférieure $a_j[n]$ et $d_j[n]$, obtenus par convolution de $a_{j-1}[n]$ avec les filtres H et G tels que :

$$\begin{aligned} a_j[n] &= a_{j-1}[n] * h[2n] \\ d_j[n] &= a_{j-1}[n] * g[2n]. \end{aligned} \quad (2.36)$$

$a_j[n]$ et $d_j[n]$ sont appelés respectivement signal d'approximation et signal de détail de $a_{j-1}[n]$. La décomposition est ensuite appliquée de nouveau en partant de $a_j[n]$. La figure (2.7) illustre ce procédé de décomposition.

Le cheminement inverse peut être réalisé. Partant d'un niveau de décomposition j , connaissant l'élément d'approximation $a_j[n]$ et de détails $d_j[n]$, il est possible de déterminer $a_{j-1}[n]$ en suivant :

$$a_{j-1}[n] = z(a_j[n]) * h[n] + z(d_j[n]) * g[n]. \quad (2.37)$$

$z(a_j[n])$ et $z(d_j[n])$ représente un rééchantillonnage de $a_j[n]$ et $d_j[n]$ avec un zéro ajouté entre chaque valeur. Ce rééchantillonnage permet de multiplier par deux la résolution des deux signaux.

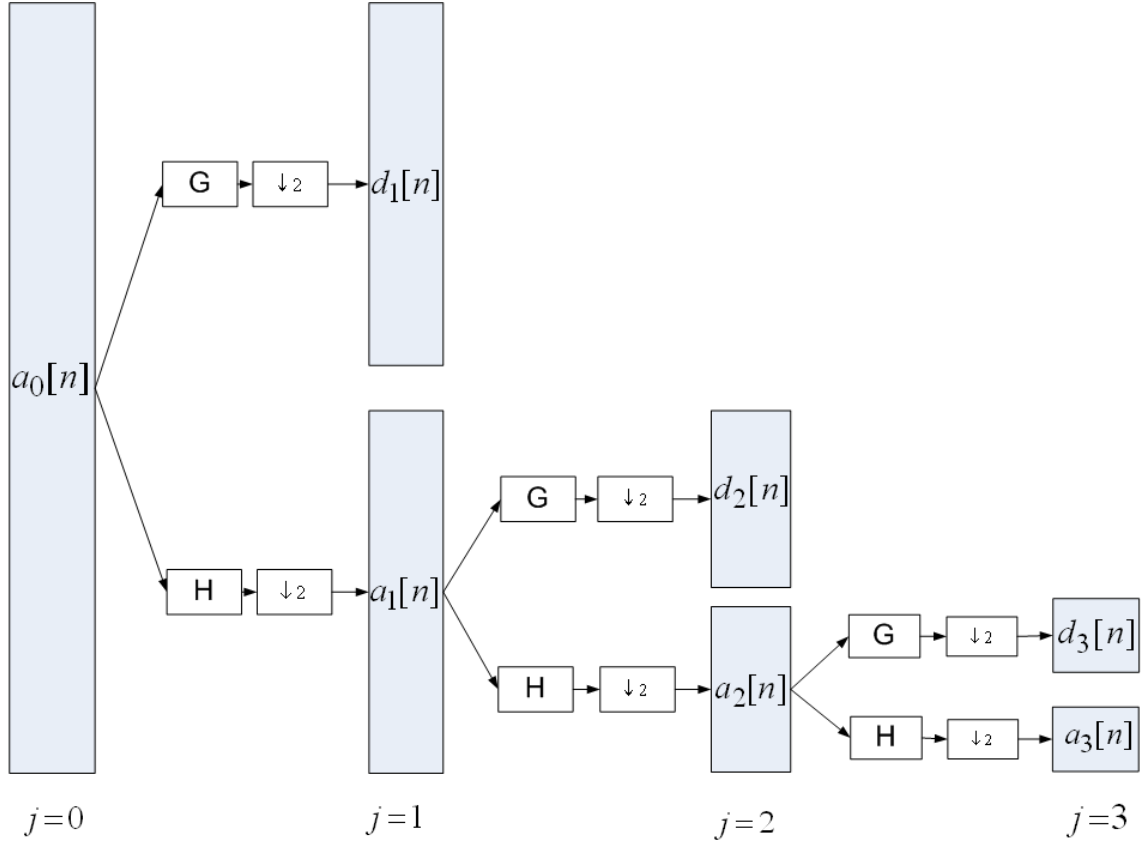


FIG. 2.7 – Décomposition en ondelettes discrète d'un signal 1D : chaque étape j de la décomposition fournit un signal d'approximation $a_j[n]$ et un signal de détail $d_j[n]$, chacun de résolution deux fois inférieure au signal d'approximation précédent (" $\downarrow 2$ " exprime la division de la résolution par deux).

Le principe de la transformée en ondelettes discrète multi-résolution s'étend aisément aux signaux à deux dimensions. A chaque phase de décomposition, deux séries de filtrage seront réalisées. Les filtres H et G vont être une première fois appliqués sur les lignes de l'image, créant deux décompositions intermédiaires. Sur chacune de ces dernières sera alors appliquée une nouvelle décomposition, mais cette fois-ci sur les colonnes de l'image. Le résultat d'une étape de décomposition complète d'un signal en deux dimensions permettra donc d'obtenir quatre sous-signaux. L'application successive du filtre passe-bas sur les lignes puis sur les colonnes déterminera le signal d'approximation $A(x, y)$ qui sera utilisé lors de la prochaine étape. Les trois autres combinaisons de filtres détermineront les trois signaux de détails $D^1(x, y)$, $D^2(x, y)$ et $D^3(x, y)$. Les figures (2.8) et (2.9) illustrent la décomposition discrète des signaux à deux dimensions.

De par sa capacité à recréer un signal à partir de ces différentes décompositions, la transformée en ondelettes est très utilisée dans le domaine de la compression d'image [Lin et Ling, 2003, Perng et Lin, 2006, Wang et Meng, 2008, Muzaffar et Choi, 2008]. Le principe est alors de décomposer l'image sur plusieurs niveaux et de supprimer les sous-

2.3. CARACTÉRISER UNE TEXTURE : DANS QUEL BUT ?

signaux de détails à partir d'un niveau fixé, ceux-ci ne contenant qu'une très faible information (cela induit une perte d'information qui peut être considérée comme négligeable). L'image d'origine peut ensuite être reconstruite à l'aide de la transformée inverse (en extrapolant les sous-signaux supprimés).

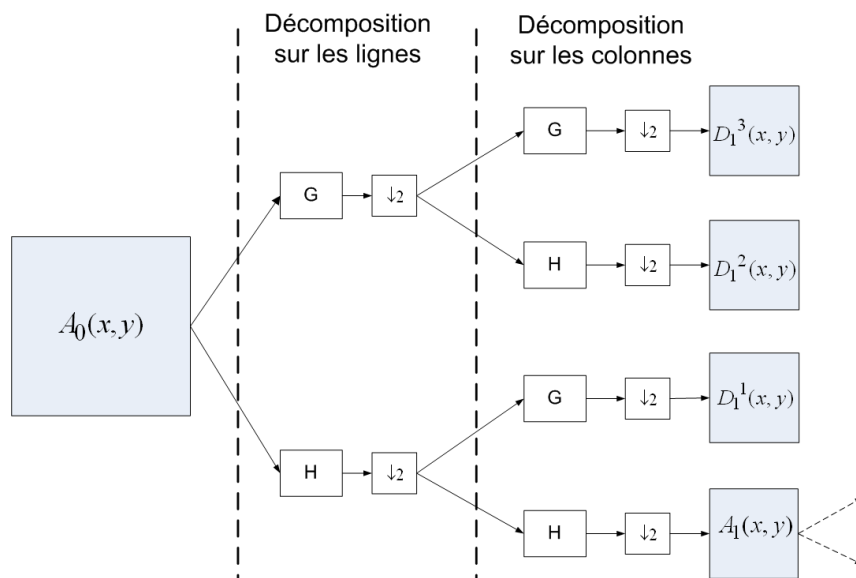


FIG. 2.8 – Principe de la décomposition en ondelettes discrète en deux dimensions. Les filtres H et G sont convolués une première fois dans le sens des lignes. Leurs sorties sont alors convoluées une nouvelle fois avec les deux filtres dans le sens des colonnes. Une étape de décomposition donnera donc trois images de détails et une image d'approximation.

Dans le domaine de l'analyse de textures, la transformée en ondelettes est généralement utilisée comme un pré-traitement à l'application d'autres descripteurs de textures [Portilla et Simoncelli, 2000, Arivazhagan et Ganesan, 2003, Ramakrishnan et Selvan, 2006, Hiremath *et al.*, 2008, Raju *et al.*, 2008]. Des mesures sont généralement réalisées à partir de l'information contenue dans les imagerie obtenues lors de la décomposition (moyenne, variance, histogramme, matrice de cooccurrence ...). La difficulté est alors de savoir à quel niveau de décomposition l'information est la plus pertinente.

2.3 Caractériser une texture : dans quel but ?

Il existe plusieurs applications à l'analyse de textures. Dans cette partie nous allons décrire les principales : la classification de textures, la synthèse de textures, la segmentation et enfin l'interprétation des formes.

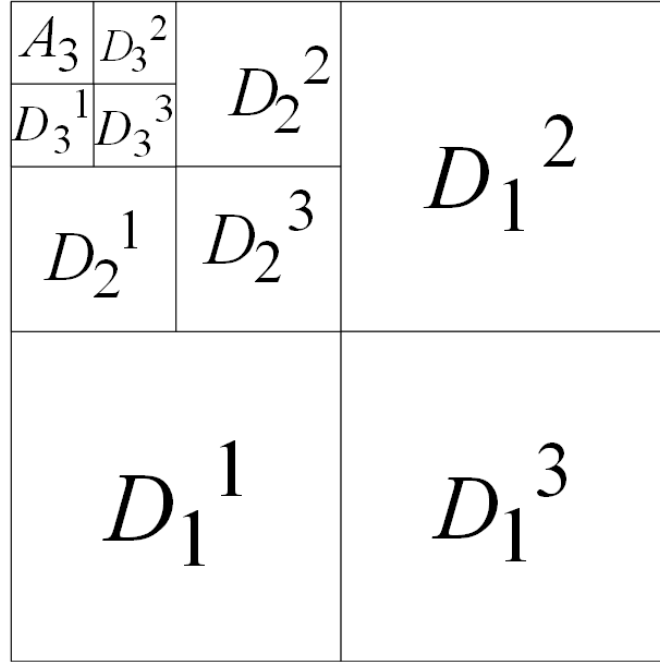


FIG. 2.9 – Trois premières étapes de la décomposition en ondelettes discrète d’une image A_0 . Chaque étape détermine quatre sous-images de résolution deux fois inférieure : trois images de détails (D^1 , D^2 et D^3) et une image d’approximation (A). Cette dernière est alors utilisée comme base de la décomposition suivante.

2.3.1 Classification de textures

Le principe de la classification de textures est de déterminer la classe associée à un type de texture particulier. Il existe deux approches de classification : supervisée et non-supervisée.

Dans le cadre de la classification supervisée, une information sur les données à traiter est disponible et est utilisée pour entraîner le processus de classification, cela constitue la phase d’apprentissage du modèle. Cette information appelée ensemble d’apprentissage est généralement constituée d’un ensemble d’individus {caractéristiques, classe associée}.

Dans le cas de la classification de textures, l’ensemble d’apprentissage est constitué d’un panel de textures "types". Chaque individu est donc composé du couple (caractéristiques de textures, texture associée). Cet ensemble est alors appris par un algorithme de classification supervisée classique (machines à vecteurs supports, réseaux de neurones, K-plus-proches-voisins ...). Une fois la phase d’apprentissage réalisée, l’algorithme de classification est alors utilisé afin de déterminer la classification d’un ensemble d’individus tests composé d’un grand nombre d’échantillons de textures. La classification de textures supervisée reste un domaine de recherche très actif [Talbar *et al.*, 1998, Kim *et al.*, 2001, Ashour *et al.*, 2008]. Plusieurs classificateurs supervisés sont détaillés au 3.2.2.3.

Dans le cadre de la classification non supervisée, aucune information *a priori* sur la texture n’est connue. On cherche alors à regrouper les différents exemples de textures

à traiter en fonction de la valeur de leurs descripteurs de manière à créer des classes homogènes. Ce procédé nécessite généralement de fixer au préalable le nombre de classes désirées, que ce soit de manière empirique ou automatique. La classification de textures non supervisée est un domaine très riche en contributions depuis une quinzaine d'années [He et Wang, 1992, Raghu *et al.*, 1997, Luo et Savakis, 2001, Idrissa et Acheroy, 2002] et reste encore d'actualité [Qin *et al.*, 2008, Kim et Kang, 2007].

2.3.2 Synthèse de textures

Le principe de la synthèse de textures est de recréer, à partir d'un échantillon d'une texture donnée, une image texturée artificielle composée du même motif mais de plus grande taille. Il existe deux principaux types d'applications dans ce domaine : le remplissage par textures de zones d'une image, où l'on va chercher à masquer une zone précise en la recouvrant d'une texture présente dans son voisinage, ou encore l'agrandissement de textures où un certain type de texture va être étendu afin d'obtenir un échantillon de taille plus importante, tout en conservant la résolution d'origine (à l'opposé d'un simple agrandissement de l'image).

Un algorithme de synthèse de textures possède généralement deux étapes : dans un premier temps les caractéristiques de l'échantillon de texture étudié sont estimées, puis ces dernières sont utilisées pour synthétiser la texture.

Trois principales familles de méthodes d'analyse sont utilisées dans ce domaine : les champs de Markov aléatoires [Paget et Longstaff, 1998, Liang *et al.*, 2001], les méthodes fréquentielles [Van Nevel, 1998, Portilla et Simoncelli, 2000] et les approches fractales [Kaplan et Jay Kuo, 1995].

2.3.3 Segmentation basée sur la texture

La segmentation basée sur la texture consiste à séparer les différentes parties présentes dans une image en fonction de leurs caractéristiques de textures. En d'autres termes, le processus de segmentation cherchera à regrouper les zones de l'image présentant des caractéristiques de textures communes.

Il existe deux manières de réaliser la segmentation d'une image texturée. La première consiste à utiliser un algorithme de segmentation classique basé sur les niveaux de gris de l'image et à le faire évoluer en prenant en compte les caractéristiques de textures des pixels présents dans l'image. La deuxième approche consiste à considérer la segmentation comme un processus de classification de textures au niveau des pixels de l'image.

Dans la première approche, en plus de la difficulté liée au choix des descripteurs de textures utilisés, débattue au 2.4, le choix de l'algorithme de segmentation reste important. Intuitivement, tout algorithme de segmentation basé sur l'étude des niveaux de gris de l'image peut être utilisé afin de déterminer les différentes régions texturées. Il suffit qu'il base sa réalisation sur les caractéristiques de textures en lieu et place des niveaux de gris. En pratique, pour être efficace sur les images texturées, un algorithme de segmentation doit être capable de prendre en compte un vecteur de caractéristiques pour chaque pixel de l'image.

2.3. CARACTÉRISER UNE TEXTURE : DANS QUEL BUT ?

Certains algorithmes de segmentation ne pouvant prendre de vecteurs de caractéristiques en entrée vont donc devoir réaliser une extraction de caractéristiques et déterminer ainsi un unique descripteur qui permettra d'obtenir le plus d'information sur la texture étudiée. Même si une telle approche peut paraître moins efficace en raison de la restriction faite au niveau du nombre de caractéristiques utilisées, certains travaux demeurent très intéressants [Schneuders, 2001, Houhou *et al.*, 2008].

Un choix plus classique consiste à adopter des algorithmes permettant de gérer des vecteurs de caractéristiques en modifiant des méthodes existantes [Malpica *et al.*, 2003, Chan *et al.*, 2000], en développant des modèles originaux [Paragios et Deriche, 1999] ou encore en utilisant des algorithmes déjà capables de gérer les vecteurs de caractéristiques [Doretto *et al.*, 2003, Ozden et Polat, 2005]. Deux courants peuvent être distingués dans ce type d'approche : les méthodes non-supervisées [Rousson *et al.*, 2003, Lianantonakis et Petillot, 2005], ne nécessitant pas d'information sur les données traitées, et les méthodes supervisées [Paragios et Deriche, 1999, Malpica *et al.*, 2003], ayant recours à une phase d'apprentissage afin de déterminer les caractéristiques discriminantes pour le modèle.

Dans l'approche par classification, le problème de la segmentation est traité comme un processus de classification de pixel ou de *labelling*. Chaque pixel est considéré comme un individu à classer et la réalisation de la classification de tous les pixels détermine les différentes régions de l'image (une région correspondant à une classe).

Cette classification peut être supervisée ou non. La segmentation supervisée nécessite la création d'une base d'apprentissage contenant des exemples de caractéristiques de pixels appartenant à chaque classe recherchée. Un classificateur classique est alors appris sur cet ensemble d'apprentissage (réseaux de neurones artificiels, machines à vecteurs supports, K -plus-proches-voisins... [Ng et Bouzerdoun, 2000, Zhan et Shen, 2006, Luo *et al.*, 2007, Kurnaz *et al.*, 2007, Li et Staunton, 2008]). Les images de test sont ensuite segmentées en réalisant la classification de tous leurs pixels.

Les méthodes non supervisées sont généralement basées sur des approches par regroupement (*clustering*) de type K-means [Hartigan et Wong, 1979, Wei, 2002], K-median [Supot *et al.*, 2007], CLARA (Clustering LARge Applications) [Kaufman et Rousseeuw, 1990] ou encore l'algorithme BIRCH [Raghavan et Birchard, 1979]. Les pixels possédant des caractéristiques proches sont alors regroupés pour former des classes de textures. La seule information *a priori* nécessaire est généralement le nombre de classes recherchées, bien que certaines méthodes le déterminent de manière automatique [Lorette *et al.*, 2000, Cariou et Chehdi, 2008]. Il peut être intéressant de noter que, même si son schéma s'éloigne de la classification "classique", la segmentation d'images texturées par champ de Markov décrite au 2.2.3.1 fait partie des méthodes de segmentation basées sur une classification non supervisée des pixels.

Les deux types d'approches (supervisée et non supervisée) sont efficaces, mais en pratique, les approches non supervisées sont généralement privilégiées car elles sont moins contraignantes et permettent d'obtenir de bonnes segmentations avec aucune ou très peu d'information *a priori*. En revanche, ce sont souvent des approches dédiées à un type de problème donné. Une segmentation supervisée permettra de développer des méthodes certes plus lourdes, mais possédant également une plus grande adaptabilité, tant au niveau du

type de texture considéré que des descripteurs utilisés.

Choisir entre une approche par segmentation ou une approche par classification dépendra des critères de segmentation estimés comme les plus importants dans le problème traité.

Une approche basée sur un algorithme de segmentation classique privilégiera l'homogénéité des régions. L'algorithme pourra laisser certains pixels dans une région même si leurs caractéristiques sont différentes afin de répondre à ce critère d'homogénéité. Au contraire, une approche par classification permettra un étiquetage plus précis des pixels d'une image. Malheureusement, cela ne pourra se faire qu'au détriment de l'homogénéité des régions créées.

Il reste néanmoins possible de combiner approches par classification et par segmentation, comme nous le détaillerons au 3.2.2.

2.3.4 Interprétation des formes à partir de l'information de texture (*Shape from Texture*)

La texture d'un objet permet parfois d'acquérir de l'information sur sa position et son orientation dans un plan à trois dimensions. Plusieurs travaux ont apporté les bases de ce domaine d'application. Ainsi dans [Bajcsy et Lieberman, 1976], une mesure de gradient sur la taille de l'élément structurant de la texture est utilisée afin de calculer ses variations et évaluer ainsi la distance des zones de l'image par rapport à la caméra. Dans [Stevens, 1980], la distance et l'orientation de surfaces sont déterminées à partir de l'étude du changement d'échelle de leurs textures. Dans [Witkin, 1981], des mesures d'inclinaison et de pente de surfaces texturées sont obtenues en utilisant l'orientation de leurs contours. Enfin dans [Blostein et Ahuja, 1989], l'orientation de surfaces texturées est déterminée à l'aide de l'extraction de leur élément structurant.

Ce domaine d'application est à l'heure actuelle encore très actif [Garding, 1995, Super et Bovik, 1995, Ribeiro et Hancock, 2002, Clerc et Mallat, 2002, Sweet et Ware, 2004].

2.4 Choisir un descripteur de textures

La littérature abondante et parfois contradictoire en ce qui concerne l'évaluation des performances des caractéristiques de textures illustre le problème que représente le choix d'un descripteur.

Ainsi, l'approche fractale, les champs de Markov, la matrice de cooccurrence et les filtres de Gabor sont comparés dans [Ohanian et Dubes, 1992] dans le cadre de la classification de textures. L'analyse par matrice de cooccurrence est identifiée comme la plus efficace. Une comparaison de plusieurs méthodes statistiques est également réalisée dans [Ojala *et al.*, 1996], et l'approche par *local binary pattern* apparaît comme la plus précise malgré une faible robustesse aux changements d'intensités (problème que l'auteur résout en combinant la méthode LBP avec une mesure de contraste). Dans [Pichler *et al.*, 1996], les ondelettes et les filtres de Gabor sont évalués pour la segmentation de textures. Les filtres

de Gabor se montrent les plus performants, mais au détriment d'un temps de calcul plus élevé. Le comportement de l'ensemble des méthodes fréquentielles est également étudié pour la classification de textures dans [Chen et Chen, 1999]. Les auteurs considèrent alors l'approche par ondelette et les filtres de Gabor comme les caractéristiques les plus efficaces.

De nombreuses autres études ont ainsi été menées et, étant donnés les différents résultats obtenus, il est réaliste de penser qu'il n'existe pas de descripteur de textures universel performant sur tout type de texture, comme il est annoncé dans [Randen et Husoy, 1999]. La conclusion de [Sharma et Singh, 2001] est similaire : toute méthode d'analyse de textures permet de caractériser une texture de manière différente et il serait intéressant de combiner les critères. De la même manière et plus récemment dans [Stachowiak *et al.*, 2005], différents descripteurs sont comparés (analyse par matrice de cooccurrence, transformée en ondelettes discrète, champs de Markov aléatoires et LBP) sur différents types de textures et même si chaque descripteur est performant sur un type de texture donné, la combinaison de la transformée en ondelettes discrète et de l'analyse par matrice de cooccurrence est performante sur chaque type. Pour Unser [Unser, 1995], il existe deux familles de caractéristiques, et chacune est plus efficace pour un type de texture : les méthodes fréquentielles (analyses dans le domaine de Fourier, filtres de Gabor, ondelettes ...) permettent d'analyser plus efficacement les textures à motifs larges (macro textures) et les méthodes à faible voisinage (matrices de cooccurrence, spectres de texture, LBP ...) réalisent une analyse plus locale et sont plus efficaces sur les textures à motifs fins (micro textures).

Choisir le ou les descripteurs de textures adéquats pour une application donnée reste donc un enjeu important dans le développement d'une méthode d'analyse de textures. Un de nos objectifs était de réaliser une méthode d'analyse d'images texturées tirant avantage des capacités de segmentation avérées des contours actifs. L'évolution d'un contour actif est guidée par les pixels présents dans le voisinage de la courbe et, dans le cas des approches régions, par des descripteurs calculés sur les régions déterminées par la position de la courbe (régions qui évoluent à mesure que le contour se déplace). Bien que notre approche ne soit pas dépendante du descripteur utilisé, nous avons cherché celui qui s'intégrait le plus naturellement dans notre processus de segmentation. Il nous fallait donc un descripteur de textures qui, en plus d'être performant, devait être applicable aussi bien pour des caractérisations locales de pixels que pour la caractérisation de régions de formes variables. Il fallait également que le descripteur reste rapide à évaluer pour ne pas trop ralentir l'évolution du contour, et calculable aussi bien en 2D qu'en 3D.

Les caractéristiques d'Haralick obtenues à l'aide de la matrice de cooccurrence possèdent ces propriétés. Leur intérêt a été démontré et ce sont des descripteurs rapides à calculer qui, quand ils sont déterminés à l'aide d'une fenêtre de voisinage, deviennent des descripteurs de textures locaux permettant de caractériser un unique pixel de l'image. De plus, ils sont applicables en tant que descripteurs de régions quelle que soit la forme de la région en 2D et en 3D.

En revanche, comme il a été décrit au 2.2.1.1, le principal inconvénient de l'analyse par matrice de cooccurrence est la forte différence de pouvoir discriminant des caractéristiques d'Haralick en fonction du type de texture étudié. Notre but était de développer une méthode de segmentation la plus générique possible. Nous ne voulions pas être dépendant d'un type de texture. Il a donc naturellement été décidé d'introduire le contour actif dans un processus

2.4. CHOISIR UN DESCRIPTEUR DE TEXTURES

de segmentation supervisé. Ainsi, nous garantissons une adaptabilité de notre méthode à de nombreuses applications en segmentation d'images texturées.

Chapitre 3

Améliorations apportées aux propriétés de segmentation des contours actifs

3.1. ACCÉLÉRATION DE CONTOURS ACTIFS PARAMÉTRIQUES PAR CALCUL DE VOISINAGES DYNAMIQUES

Ce chapitre présente les différentes approches développées dans le but d'améliorer les contours actifs. La première partie décrit les méthodes d'accélération par calcul de voisinages dynamiques intégrées dans l'algorithme d'évolution gloutonne des modèles paramétriques. La deuxième partie détaille les modèles de contours actifs implicites supervisés développés pour la segmentation d'images texturées.

3.1 Accélération de contours actifs paramétriques par calcul de voisinages dynamiques

De part leur faible complexité, les contours actifs implémentés de manière paramétrique sont des modèles rapides encore fréquemment utilisés (voir au 1.1.2.1 pour plus de détails). Dans le cadre d'une démarche cherchant à améliorer cette famille de contours actifs, nous avons jugé, en l'état actuel de nos connaissances, qu'il n'était pas forcément pertinent de tenter d'améliorer la précision des modèles paramétriques. En effet, la précision de segmentation reste le domaine de prédilection des modèles implicites. Il semble donc illusoire d'espérer un niveau de performances équivalent de la part d'un modèle paramétrique, en particulier en raison de la gestion de la courbe à l'aide de points de contrôle. Dès lors, il nous a semblé judicieux de concentrer nos travaux sur la vitesse de segmentation, caractéristique principale des modèles paramétriques.

En raison de sa grande rapidité, l'algorithme de déformation le plus utilisé pour les modèles paramétriques est l'évolution gloutonne. La particularité de cette approche est l'utilisation d'une fenêtre de voisinage fixe afin de déterminer, pour chaque point de contrôle, la position voisine possédant la plus grande influence sur la diminution de l'énergie du contour actif.

Dans le but d'accélérer l'évolution gloutonne, il nous a paru intéressant d'insérer un soupçon d'adaptabilité dans la gestion de la grille de voisinage. En tirant profit de l'information contenue dans la direction suivie par la courbe lors de l'évolution passée, le voisinage est modifié dynamiquement afin de privilégier, par anticipation, la direction semblant la plus efficace. Trois types de voisinages dynamiques ont été développés : un voisinage linéaire, un voisinage décalé et un voisinage déformé.

Ces trois méthodes ont été implémentées pour des modèles paramétriques 2D et 3D, et également comparées, dans le cadre de la segmentation 3D, à l'implémentation implicite rapide de Shi et Karl (décrite au 1.1.2.2), afin de prouver que la supériorité des modèles paramétriques, en terme de rapidité, reste d'actualité. Tous les tests réalisés sont détaillés au chapitre 4.

3.1.1 Voisinage à anticipation linéaire

Lors de l'évolution d'un contour actif paramétrique implémenté par évolution gloutonne, à chaque itération t , la nouvelle position de chaque point de contrôle $\mathbf{v}_i^{(t+1)}$ est déterminée à l'aide d'une grille de voisinage $N_i^{(t)}$ définie autour de $\mathbf{v}_i^{(t)}$ telle que :

$$\mathbf{v}_i^{(t+1)} = \arg \min_{\mathbf{v}'_k \in N_i^{(t)}} E(\mathbf{v}'_k) . \quad (3.1)$$

3.1. ACCÉLÉRATION DE CONTOURS ACTIFS PARAMÉTRIQUES PAR CALCUL DE VOISINAGES DYNAMIQUES

La grille de voisinage $N_i^{(t)}$ est généralement considérée comme un carré de largeur L_N , centrée autour de $\mathbf{v}_i^{(t)}$. Le point $\mathbf{v}_i^{(t+1)}$ représente alors le pixel possédant l'énergie minimale dans le voisinage de $\mathbf{v}_i^{(t)}$ (dont ce dernier fait partie).

Le principe du voisinage dynamique à anticipation linéaire est d'alterner, au sein de chaque itération de l'algorithme d'évolution gloutonne, une évolution avec un voisinage classique, et une évolution utilisant un voisinage linéaire.

A l'itération t , chaque déplacement réalisé avec la grille de voisinage classique permet de déterminer la direction suivie par la courbe en chaque point de contrôle $\mathbf{v}_i^{(t)}$. Cette direction peut être considérée comme celle entraînant la meilleure minimisation de la fonctionnelle d'énergie localement au niveau de chaque point de contrôle.

La vitesse du contour actif dépend directement de la rapidité de cette minimisation. Aussi, afin d'accélérer la segmentation, la méthode de voisinage par anticipation linéaire va forcer la courbe à rechercher plus en profondeur dans la direction courante. Une nouvelle itération va donc être réalisée en utilisant, cette fois-ci, un voisinage linéaire de taille fixe orienté dans cette direction.

Dans notre approche, le résultat de l'équation (3.1) constitue la première étape de déplacement de chaque point de contrôle et détermine un premier point temporaire $\mathbf{v}_i'^t$. Le mouvement $\mathbf{v}_i^t \rightarrow \mathbf{v}_i'^t$ est effectué afin de déterminer, pour chaque point de contrôle $\mathbf{v}_i$, un vecteur unitaire $\mathbf{d}_i^t$ représentant l'orientation du déplacement subit :

$$\mathbf{d}_i^{(t)} = \frac{\mathbf{v}_i'^{(t)} - \mathbf{v}_i^{(t)}}{\|\mathbf{v}_i'^{(t)} - \mathbf{v}_i^{(t)}\|} . \quad (3.2)$$

Utilisant $\mathbf{d}_i^{(t)}$, un voisinage linéaire $N_{L_i}^{(t)}$ est déterminé. Celui-ci se compose des l pixels les plus proches de $\mathbf{v}_i'^{(t)}$ dans la direction donnée par $\mathbf{d}_i^{(t)}$ et est défini par :

$$N_{L_i}^{(t)} = \left\{ \mathbf{v}_i'^{(t)} + \left\lfloor k \cdot \mathbf{d}_i^{(t)} \right\rfloor, \forall k \in [1, l] \right\} . \quad (3.3)$$

$\lfloor \cdot \rfloor$ est l'opérateur de partie entière permettant d'atteindre les l pixels appartenant au voisinage linéaire et disposés sur la grille discrète des pixels de l'image. Il est admis que l'opérateur de partie entière appliqué sur un vecteur est équivalent à la partie entière de chacune de ces composantes.

Chaque pixel composant $N_{L_i}^{(t)}$ est alors exploré et celui permettant de minimiser la fonctionnelle d'énergie $E(\mathbf{v}_i'^{(t)})$ du $i^{\text{ème}}$ point de contrôle est choisi comme nouvelle position $\mathbf{v}_i^{(t+1)}$ en suivant :

$$m = \underset{k \in [0, l]}{\operatorname{argmin}} \left\{ E(\mathbf{v}_i'^{(t)} + \left\lfloor k \cdot \mathbf{d}_i^{(t)} \right\rfloor) \right\} . \quad (3.4)$$

La figure (3.1) illustre la manière dont la direction suivie par chaque point de contrôle est déterminée puis utilisée afin de définir le voisinage linéaire pour une grille de voisinage de largeur 5 et un voisinage linéaire de taille 6. L'algorithme (1) décrit la méthode complète pour un contour actif de n points de contrôle, T itérations de l'algorithme d'évolution gloutonne et un voisinage linéaire de taille l .

3.1. ACCÉLÉRATION DE CONTOURS ACTIFS PARAMÉTRIQUES PAR CALCUL DE VOISINAGES DYNAMIQUES

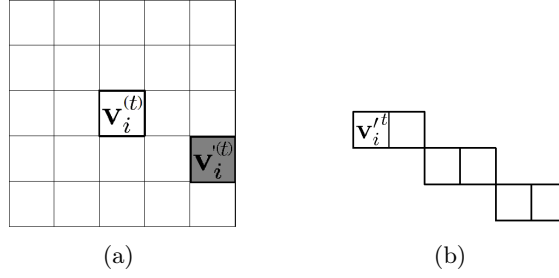


FIG. 3.1 – Illustration de la création du voisinage linéaire. a) Itération classique de l'évolution gloutonne pour un point $\mathbf{v}_i^{(t)}$. La nouvelle position choisie pour $\mathbf{v}_i^{(t)}$ est $\mathbf{v}_i'^{(t)}$. Le déplacement effectué étant $(2, 1)$, le vecteur unitaire $\mathbf{d}_i^{(t)}$ défini par l'équation (3.2) vaut $(0, 9; 0, 45)$. b) Le voisinage linéaire est créé selon l'équation (3.3) $\forall k \in [1, 6]$, donnant l'ensemble des coordonnées des pixels dans le voisinage linéaire relativement à $\mathbf{v}_i'^{(t)}$: $\{(0; 0), (1; 0), (2; 1), (3; 1), (4; 2), (5; 2)\}$.

Algorithme 1 : Accélération de contour actif par voisinage à anticipation linéaire

```

1 début
2   pour  $t \leftarrow 1$  à  $T$  faire
3     pour  $i \leftarrow 1$  à  $n$  faire
4       // Premier déplacement avec une grille de voisinage classique :
        $\mathbf{v}_i'^{(t)} = \arg \min_{\mathbf{v}_k' \in \mathcal{N}_i^{(t)}} E(\mathbf{v}_k')$ ;
       // Deuxième déplacement avec un voisinage linéaire :
       Calcul de la direction suivie  $\mathbf{d}_i^{(t)} = \frac{\mathbf{v}_i'^{(t)} - \mathbf{v}_i^{(t)}}{\|\mathbf{v}_i'^{(t)} - \mathbf{v}_i^{(t)}\|}$ ;
5       Evolution par voisinage linéaire :  $m = \operatorname{argmin}_{k \in [0, l]} \{E(\mathbf{v}_i^{(t)} + k\mathbf{d}_i^{(t)})\}$ ;
6       Mise à jour :  $\mathbf{v}_i^{(t+1)} \leftarrow \mathbf{v}_i^{(t)} + m\mathbf{d}_i^{(t)}$ ;
7     fin
8 fin

```

3.1. ACCÉLÉRATION DE CONTOURS ACTIFS PARAMÉTRIQUES PAR CALCUL DE VOISINAGES DYNAMIQUES

La figure (3.2) illustre une itération complète de notre méthode intégrée dans l'évolution gloutonne pour une grille de voisinage de 5 pixels de large et un voisinage linéaire de taille 6. Dans la figure (3.2.a), le pixel noir au centre de la courbe représente le point de contrôle $\mathbf{v}_i^{(t)}$ en cours de déplacement à l'itération t . L'objet recherché se situe en bas à droite de l'image. Les figures (3.2.b) et (3.2.c) illustrent l'exploration du voisinage de $\mathbf{v}_i^{(t)}$ (pixels gris foncés) et la réalisation d'un premier déplacement de $\mathbf{v}_i^{(t)}$ vers le pixel possédant l'énergie minimale $\mathbf{v}'_i^{(t)}$ (pixel gris clair). Dans la figure (3.2.d), le mouvement $\mathbf{v}_i^{(t)} \rightarrow \mathbf{v}'_i^{(t)}$ permet de déterminer le vecteur de déplacement $\mathbf{d}_i^{(t)}$ et le voisinage linéaire $N_{L_i}^{(t)}$ (ligne de pixels gris foncés). Celui-ci est ensuite exploré et le pixel possédant l'énergie minimum (pixel gris clair placé sur la frontière de l'objet recherché) est choisi comme nouveau point de contrôle $\mathbf{v}_i^{(t+1)}$. Les points de contrôle suivants sont alors traités.

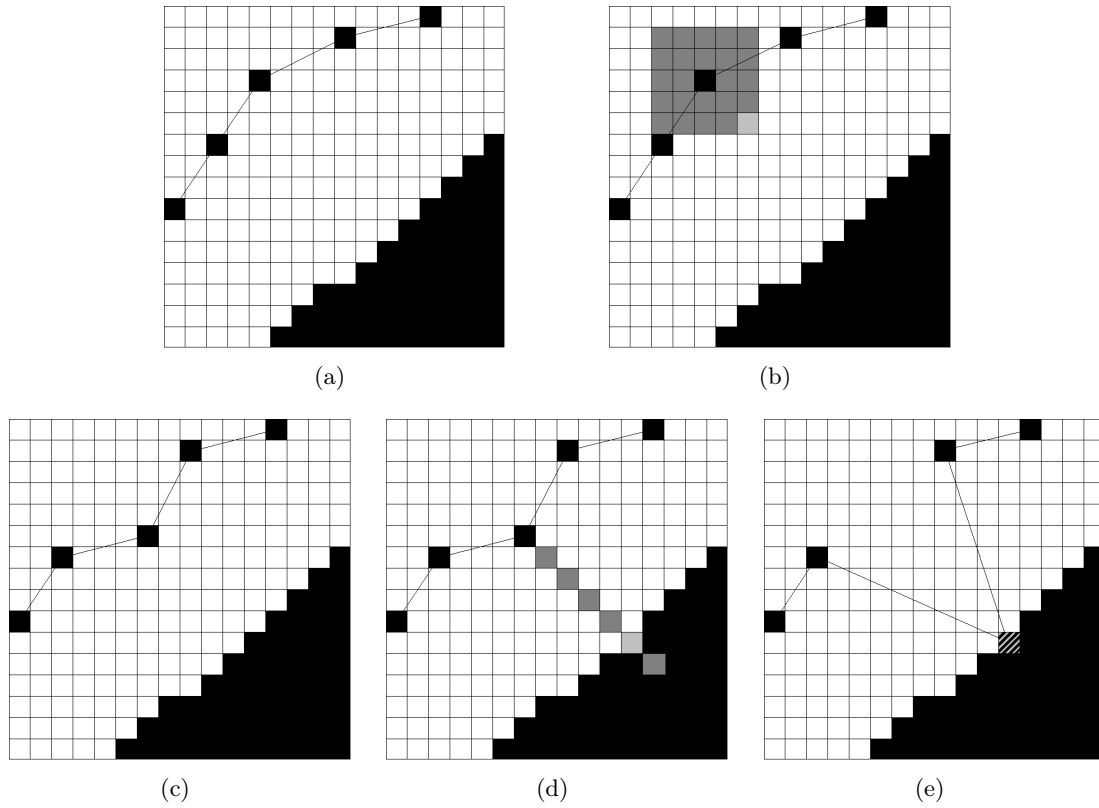


FIG. 3.2 – Illustration d'une itération complète de l'évolution gloutonne avec anticipation par voisinage linéaire.

3.1.2 Voisinage à anticipation par décalage

L'intérêt d'insérer une notion d'anticipation dans l'évolution gloutonne d'un modèle paramétrique a été introduit par l'approche par voisinage linéaire. Toutefois, il peut parfois être trop restrictif de forcer l'évolution des points de contrôle de la courbe dans une unique direction, le risque étant que ces derniers ne se retrouvent trop contraints et passent alors à côté de zones intéressantes (voir l'illustration de ce phénomène en figure (3.3)). Une solution est de ne plus "forcer" une unique direction, mais au contraire d'orienter simplement l'évolution, tout en laissant à chaque point de contrôle la possibilité de réaliser un changement de direction si cela s'avère nécessaire.

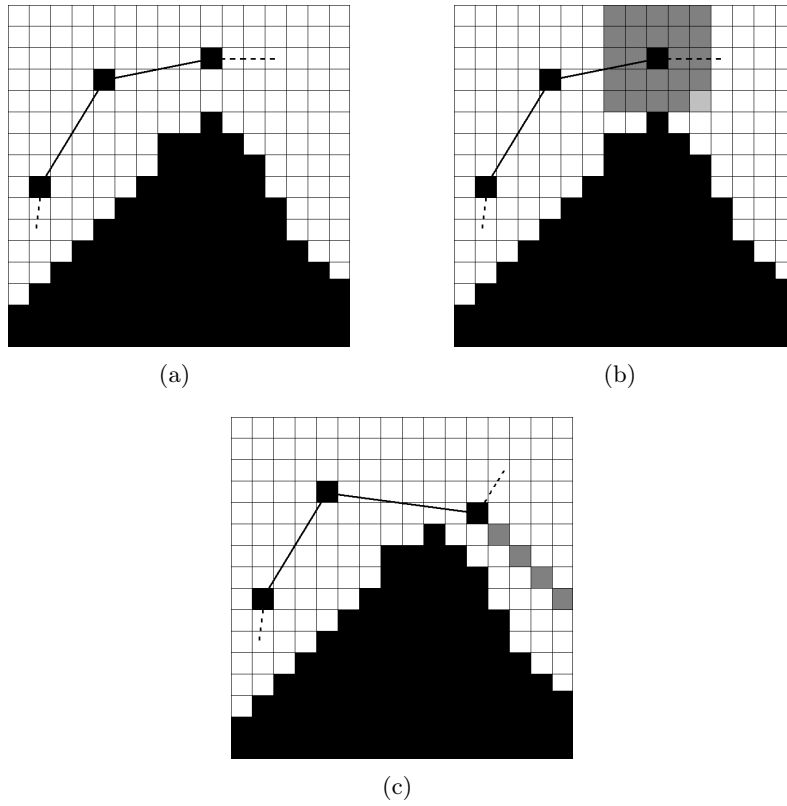


FIG. 3.3 – Illustration d'un effet non désiré de l'utilisation du voisinage à anticipation linéaire. a) Le pixel à l'extrémité droite de la courbe représente le point de contrôle $\mathbf{v}_i^{(t)}$ actuellement en cours d'étude. b) Premier déplacement de $\mathbf{v}_i^{(t)}$ vers $\mathbf{v}_i'^{(t)}$. c) Le voisinage linéaire $N_{L_i}^{(t)}$ déterminé est trop restrictif et guide le contour à côté de l'objet à segmenter. Celui-ci sera atteint quelques itérations plus tard, mais cela influencera la rapidité de segmentation du modèle.

Le voisinage à anticipation par décalage a été développé afin de prendre en compte ce degré de liberté. L'anticipation n'est plus réalisée lors d'une phase d'évolution particulière, mais directement intégrée dans la gestion de la grille de voisinage. Celle-ci est ainsi autorisée à se décaler vers la direction semblant la plus intéressante.

3.1. ACCÉLÉRATION DE CONTOURS ACTIFS PARAMÉTRIQUES PAR CALCUL DE VOISINAGES DYNAMIQUES

Les grilles de voisinages n'étant alors plus obligatoirement centrées sur leurs points de contrôle respectifs, une nouvelle notation doit être introduite. Si l'on considère un repère dont l'origine est le pixel à l'extrémité supérieure gauche de la grille de voisinage, la position de chaque point de contrôle à l'intérieur de sa grille est alors définie par un vecteur de décalage $\mathbf{s}_i^{(t)}$. Ainsi, dans le cas d'un voisinage classique (i.e. $\mathbf{v}_i^{(t)}$ est au centre de sa grille), le vecteur $\mathbf{s}_i^{(t)}$ s'exprime comme :

$$\mathbf{s}_i^{(t)} = \begin{pmatrix} \lfloor L_N/2 \rfloor \\ \lfloor L_N/2 \rfloor \end{pmatrix}. \quad (3.5)$$

De manière similaire à la méthode du voisinage à anticipation linéaire, à chaque itération de l'algorithme d'évolution gloutonne, la direction suivie par chaque point de contrôle est conservée. Le but de cette approche étant d'orienter la grille de voisinage par pas de un pixel, le déplacement de chaque point de contrôle est approximé sur une grille de voisinage 3×3 en utilisant une fonction de décalage limite $D_L(L_{min}, L_{max}, \mathbf{v})$ définie par :

$$D_L(L_{min}, L_{max}, \mathbf{v}) = \begin{pmatrix} \max(L_{min}, \min(L_{max}, v_x)) \\ \max(L_{min}, \min(L_{max}, v_y)) \end{pmatrix} \quad (3.6)$$

$$\forall L_{min}, L_{max} \in \mathbb{N}, L_{max} > L_{min}, \mathbf{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} \in \mathbb{N}^2.$$

La direction suivie par chaque point de contrôle $\mathbf{v}_i^{(t)}$ à l'itération t est ainsi approximée par le déplacement $\mathbf{d}_i^{(t)}$ défini comme :

$$\mathbf{d}_i^{(t)} = D_L(-1, 1, \mathbf{v}_i^{(t+1)} - \mathbf{v}_i^{(t)}). \quad (3.7)$$

$\mathbf{d}_i^{(t)}$ ne peut prendre que 8 valeurs de déplacement différentes dans une grille 3×3 , comme l'illustre la figure (3.4).

Une fois la direction $\mathbf{d}_i^{(t)}$ déterminée, la translation de la grille de voisinage lors de l'itération $t + 1$ est obtenue en calculant le vecteur de décalage $\mathbf{s}_i^{(t+1)}$ représentant la nouvelle position du point de contrôle $\mathbf{v}_i^{(t+1)}$ dans sa grille de voisinage :

$$\mathbf{s}_i^{(t+1)} = \mathcal{D}_L(1, L_N - 2, \mathbf{s}_i^{(t)} - \mathbf{d}_i^{(t)}). \quad (3.8)$$

L'expression $\mathbf{s}_i^{(t)} - \mathbf{d}_i^{(t)}$ permet de calculer les nouvelles coordonnées de $\mathbf{v}_i^{(t+1)}$ dans sa grille de voisinage. La fonction de décalage limite est une nouvelle fois utilisée afin de s'assurer qu'un point de contrôle ne se retrouvera jamais sur la périphérie de sa grille de voisinage. En effet, si ce cas de figure se présentait, de nombreuses directions ne pourraient plus être explorées par le point de contrôle car elle correspondraient à des pixels à l'extérieur de la grille de voisinage. Cette dernière n'aurait alors plus aucune possibilité d'être à nouveau décalée dans la direction opposée, et l'évolution du modèle se retrouverait bloquée sans aucun moyen de revenir en arrière.

Une fois les coordonnées du décalage de $\mathbf{v}_i^{(t+1)}$ déterminées, la définition du voisinage à l'itération $t + 1$ est donnée par :

$$N_i^{(t+1)} = \left\{ \mathbf{v}_i^{(t+1)} + \mathbf{r} - \mathbf{s}_i^{(t+1)} \mid \mathbf{r} \in [0, L_N - 1]^2 \right\}. \quad (3.9)$$

3.1. ACCÉLÉRATION DE CONTOURS ACTIFS PARAMÉTRIQUES PAR CALCUL DE VOISINAGES DYNAMIQUES

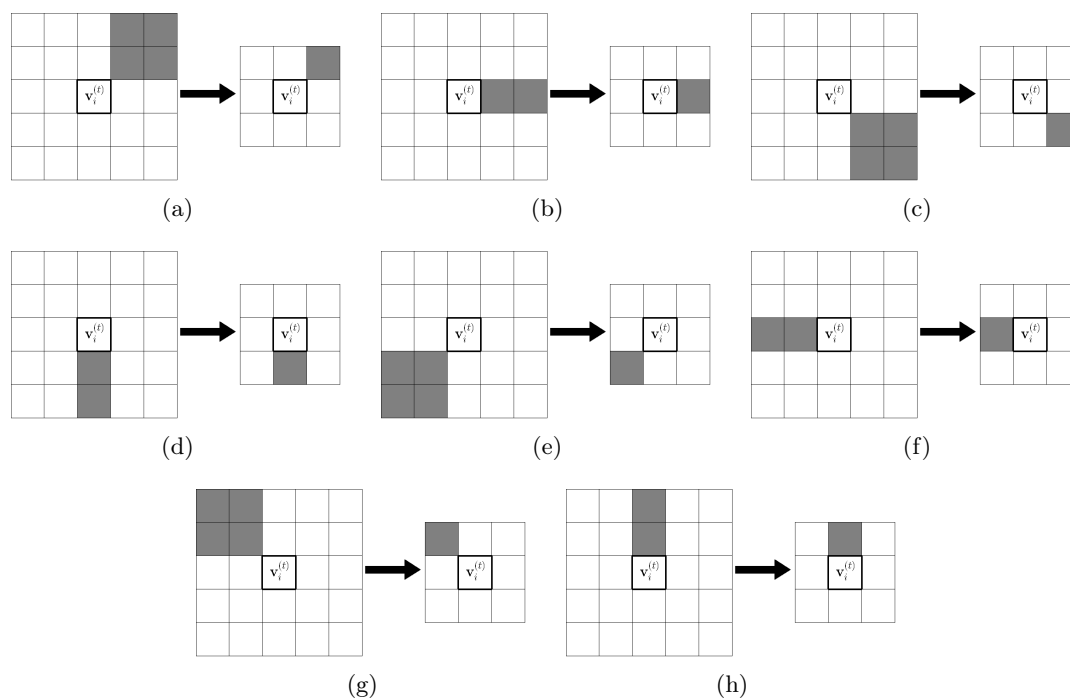


FIG. 3.4 – Différentes valeurs prises par $\mathbf{d}_i^{(t)}$. Sur chaque imagerie, la figure de gauche correspond aux différents déplacements $\mathbf{v}_i^{(t)} \rightarrow \mathbf{v}_i^{(t+1)}$ (représentés par les pixels gris) pouvant être réalisés lors d'une itération de l'évolution gloutonne avec un voisinage classique. La figure de droite représente le déplacement $\mathbf{d}_i^{(t)}$ approximé.

L'algorithme (2) présente la méthode du voisinage à anticipation par décalage introduite dans l'évolution gloutonne. La figure (3.5) illustre le comportement de la méthode sur un point de contrôle et une itération pour un voisinage de largeur $L_N = 5$.

3.1. ACCÉLÉRATION DE CONTOURS ACTIFS PARAMÉTRIQUES PAR CALCUL DE VOISINAGES DYNAMIQUES

Algorithme 2 : Accélération de contour actif par voisinage à anticipation par décalage

```

1 début
2   pour  $t \leftarrow 1$  to  $T$  faire
3     pour  $i \leftarrow 1$  to  $n$  faire
4       // Déplacement du point  $\mathbf{v}_i^{(t)}$  en utilisant la grille de voisinage  $N_i^{(t)}$  :
        $\mathbf{v}_i^{(t+1)} = \arg \min_{\mathbf{v}'_k \in N_i^{(t)}} E(\mathbf{v}'_k)$ ;
       // Calcul du décalage de la nouvelle grille de voisinage pour l'itération
        $t + 1$  :
5        $\mathbf{d}_i^{(t)} = \mathcal{D}_L(-1, 1, \mathbf{v}_i^{(t+1)} - \mathbf{v}_i^{(t)})$ ;
6        $\mathbf{s}_i^{(t+1)} = \mathcal{D}_L(1, L_N - 2, \mathbf{s}_i^{(t)} - \mathbf{d}_i^{(t)})$ ;
7        $N_i^{(t+1)} = \left\{ \mathbf{v}_i^{(t+1)} + \mathbf{r} - \mathbf{s}_i^{(t+1)} \mid \mathbf{r} \in [0, L_N - 1]^2 \right\}$ ;
8 fin

```

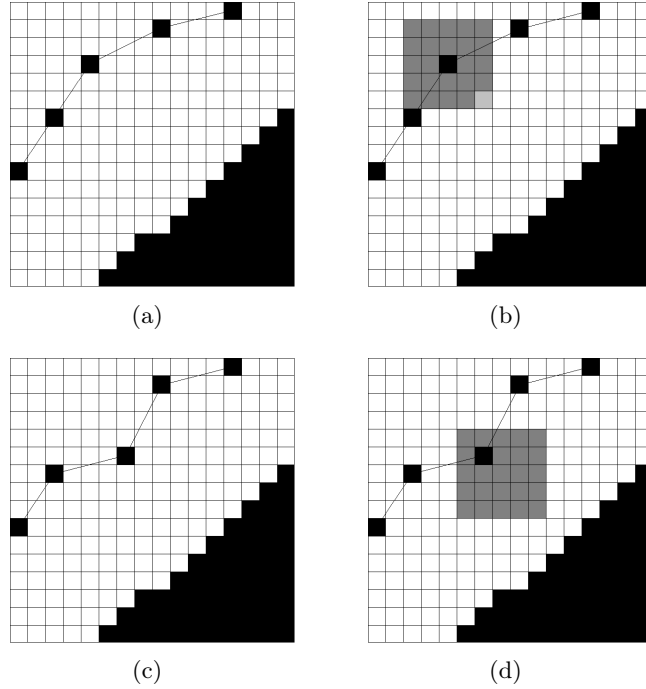


FIG. 3.5 – Illustration de l'utilisation du voisinage à anticipation par décalage pour un point de contrôle. a-b-c) Déroulement de l'itération classique de l'évolution gloutonne. d) Le déplacement du point de contrôle entraîne le décalage de la grille de voisinage dans la direction suivie. Ainsi, lors de la prochaine itération, un plus grand nombre de pixels sera exploré dans la direction semblant la plus intéressante.

3.1.3 Voisinage à anticipation par déformation

Les deux méthodes présentées dans les sections précédentes ont permis de mettre en avant l'intérêt d'intégrer une notion d'anticipation dans l'évolution d'un contour actif. Cependant, une information pouvant se révéler intéressante n'est pas prise en compte lors des déplacements : la notion de fiabilité de la direction choisie par l'évolution gloutonne. En effet, lorsque l'algorithme d'évolution gloutonne détermine un pixel dans le voisinage de $\mathbf{v}_i^{(t)}$, il peut s'avérer utile de savoir si le modèle a une forte probabilité de poursuivre dans la même direction lors de l'itération suivante, ou si au contraire ce choix n'a que peu de chances de se réitérer. En d'autres termes, il s'agit de déterminer une mesure de fiabilité sur la direction suivie.

C'est dans le but d'intégrer une telle notion que le voisinage à anticipation par déformation a été développé. La fiabilité peut être évaluée lors du déplacement de chaque point de contrôle. Deux cas sont alors considérés.

1. Le pixel choisi est placé sur la périphérie de la grille de voisinage : la fiabilité de la direction est considérée comme forte car par son biais, l'algorithme cherche à s'éloigner le plus possible de la position actuelle. Il est donc logique de penser que si la grille avait été plus large, un pixel plus lointain dans cette direction aurait certainement été choisi.
2. Le pixel choisi n'est pas placé sur la périphérie de la grille de voisinage : la fiabilité de la direction est considérée comme faible car l'algorithme a préféré garder le point de contrôle proche de sa position et cela même si cette direction est considérée comme la plus intéressante.

Dans le cadre de la méthode du voisinage à anticipation par déformation, ces deux cas de figure vont entraîner une modification différente de la grille de voisinage. Une fois la direction suivie $\mathbf{d}_i^{(t)}$ déterminée en utilisant l'équation (3.7), si celle-ci est considérée fiable, elle va être fortement favorisée. La grille de voisinage va alors être modifiée et des pixels vont être à la fois ajoutés selon $\mathbf{d}_i^{(t)}$ et enlevés sur le ou les côtés opposés (dans ce cas, cela revient à effectuer un simple décalage de la grille équivalent à celui du voisinage à anticipation par décalage). Si $\mathbf{d}_i^{(t)}$ est considérée peu fiable, alors la grille de voisinage va être modifiée de manière à favoriser cette direction, mais sans pour autant défavoriser les directions opposées. Ainsi, des pixels vont être ajoutés sur les côtés correspondants à $\mathbf{d}_i^{(t)}$ mais aucun ne sera supprimé. La grille de voisinage ne sera alors plus nécessairement une fenêtre carrée, mais pourra désormais être rectangulaire.

Etant donnée la forme variable pouvant être prise par la grille de voisinage, les notions de largeur et de hauteur de grille sont introduites. Ainsi, $L_i^{(t)}$ correspondra à la largeur de la grille de voisinage $N_i^{(t)}$ à l'itération t , et $H_i^{(t)}$ à sa hauteur. Par soucis de simplicité, et toujours en raison de la forme variable prise par la grille de voisinage, le vecteur de décalage est décomposé selon ses deux coordonnées tel que $\mathbf{s}_i^{(t)} = \begin{pmatrix} s_{ix}^{(t)} \\ s_{iy}^{(t)} \end{pmatrix}$

A chaque itération t et après le déplacement de chaque point de contrôle $\mathbf{v}_i^{(t)}$, la méthode du voisinage à anticipation par déformation doit déterminer la nouvelle grille de voisinage $N_i^{(t+1)}$, définie par ses nouvelles dimensions $L_i^{(t+1)}$, $H_i^{(t+1)}$ et $\mathbf{s}_i^{(t+1)}$, la nouvelle position

3.1. ACCÉLÉRATION DE CONTOURS ACTIFS PARAMÉTRIQUES PAR CALCUL DE VOISINAGES DYNAMIQUES

relative de $\mathbf{v}_i^{(t+1)}$ dans sa grille.

Afin de calculer ces paramètres, le déplacement suivi par chaque point est dans un premier temps déterminé en utilisant l'équation (3.7) du voisinage à anticipation par décalage. Le déplacement $\mathbf{d}_i^{(t)} = \begin{pmatrix} d_{ix}^{(t)} \\ d_{iy}^{(t)} \end{pmatrix}$ est alors décomposé selon chacune de ses deux coordonnées.

Si le pixel choisi par l'algorithme d'évolution gloutonne se situe sur la périphérie de $N_i^{(t)}$, les nouvelles dimensions de $N_i^{(t+1)}$ ne sont pas modifiées car la grille est simplement décalée. Seul le vecteur de décalage $\mathbf{s}_i^{(t+1)}$ sera modifié, et déterminé par :

$$s_{ix}^{(t+1)} = D_L(1, L_i^{(t+1)} - 2, s_{ix}^{(t)} - d_{ix}^{(t)}) \quad (3.10)$$

$$s_{iy}^{(t+1)} = D_L(1, H_i^{(t+1)} - 2, s_{iy}^{(t)} - d_{iy}^{(t)}) . \quad (3.11)$$

De manière similaire à l'équation (3.8), la fonction de décalage limite est utilisée afin de s'assurer que l'évolution ne se retrouve pas bloquée dans une unique direction.

Si le pixel choisi n'est pas placé sur la périphérie de la grille, les nouvelles dimensions seront :

$$L_{N_i^{(t+1)}} = L_i^{(t)} + d_{ix}^{(t)} \quad (3.12)$$

$$H_{N_i^{(t+1)}} = H_i^{(t)} + d_{iy}^{(t)} . \quad (3.13)$$

Les coordonnées $s_{ix}^{(t+1)}$ et $s_{iy}^{(t+1)}$ ne seront différentes de $s_{ix}^{(t)}$ et $s_{iy}^{(t)}$ que si le déplacement réalisé selon ces composantes est négatif. En effet, dans le cas d'un déplacement $d_{ix}^{(t)}$ positif, une colonne est ajoutée à l'extrémité droite de la grille de voisinage. La largeur de la grille est donc modifiée, mais le vecteur de décalage reste identique. En revanche, si $d_{ix}^{(t)} < 0$, une colonne est ajoutée à l'extrémité gauche de la grille. Cette opération déplace l'origine du repère du vecteur de décalage. Celui-ci doit donc être modifié en conséquence afin de conserver la position du point de contrôle dans l'image. Ainsi, les coordonnées de $\mathbf{s}_i^{(t+1)}$ sont déterminées par :

$$s_{ix}^{(t+1)} = s_{ix}^{(t)} - \min(0, d_{ix}^{(t)}) \quad (3.14)$$

$$s_{iy}^{(t+1)} = s_{iy}^{(t)} - \min(0, d_{iy}^{(t)}) . \quad (3.15)$$

La méthode du voisinage à anticipation par décalage est introduite directement dans l'algorithme d'évolution gloutonne, comme l'illustre l'algorithme (3).

3.1. ACCÉLÉRATION DE CONTOURS ACTIFS PARAMÉTRIQUES PAR CALCUL DE VOISINAGES DYNAMIQUES

Algorithme 3 : Accélération de contour actif par voisinage à anticipation par déformation

```

1  début
2  |   pour  $t \leftarrow 1$  to  $T$  faire
3  |   |   pour  $i \leftarrow 1$  to  $n$  faire
4  |   |   |   // Déplacement du point  $\mathbf{v}_i^{(t)}$  en utilisant la grille de voisinage  $N_i^{(t)}$  :
4  |   |   |    $\mathbf{v}_i^{(t+1)} = \arg \min_{\mathbf{v}'_k \in N_i^{(t)}} E(\mathbf{v}'_k);$ 
5  |   |   |   // Calcul de la forme de la nouvelle grille pour l'itération  $t + 1$  :
5  |   |   |    $\mathbf{d}_i^{(t)} = \mathcal{D}_L(-1, 1, \mathbf{v}_i^{(t+1)} - \mathbf{v}_i^{(t)});$ 
6  |   |   |   si  $\mathbf{v}_i^{(t+1)}$  est sur la périphérie de  $N_i^{(t)}$  alors
7  |   |   |   |    $L_i^{(t+1)} = L_i^{(t)};$ 
8  |   |   |   |    $H_i^{(t+1)} = H_i^{(t)};$ 
9  |   |   |   |    $s_{ix}^{(t+1)} = D_L(1, L_i^{(t+1)} - 2, s_{ix}^{(t)} - d_{ix}^{(t)}) ;$ 
10  |   |   |   |    $s_{iy}^{(t+1)} = D_L(1, H_i^{(t+1)} - 2, s_{iy}^{(t)} - d_{iy}^{(t)});$ 
11  |   |   |   sinon
12  |   |   |   |    $L_i^{(t+1)} = L_i^{(t)} + d_{ix}^{(t)};$ 
13  |   |   |   |    $H_i^{(t+1)} = H_i^{(t)} + d_{iy}^{(t)};$ 
14  |   |   |   |    $s_{ix}^{(t+1)} = s_{ix}^{(t)} - \min(0, d_{ix}^{(t)});$ 
15  |   |   |   |    $s_{iy}^{(t+1)} = s_{iy}^{(t)} - \min(0, d_{iy}^{(t)});$ 
16  |   |   |    $N_i^{(t+1)} = \left\{ \mathbf{v}_i^{(t+1)} + \mathbf{r} - \mathbf{s}_i^{(t+1)} \mid \mathbf{r} \in [0, L_i^{(t+1)} - 1] \times [0, H_i^{(t+1)} - 1] \right\};$ 
17 fin

```

3.1. ACCÉLÉRATION DE CONTOURS ACTIFS PARAMÉTRIQUES PAR CALCUL DE VOISINAGES DYNAMIQUES

La figure (3.6) présente le déroulement de l'algorithme pour deux points de contrôle $\mathbf{v}_i$ et $\mathbf{v}_j$ donnés, possédant un voisinage de taille $H_i^0 = H_j^0 = L_i^0 = L_j^0 = 5$. La figure (3.6.a) indique les positions initiales de $\mathbf{v}_i^{(t)}$ (à gauche), $\mathbf{v}_j^{(t)}$ (à droite) et de leurs grilles de voisinage respectives $N_i^{(t)}$ et $N_j^{(t)}$. Les deux voisinages étant centrés, $\mathbf{s}_i^{(t)} = \mathbf{s}_j^{(t)} = (2, 2)$. La figure (3.6.b) illustre la réalisation du déplacement de $\mathbf{v}_i^{(t)}$. Le nouveau point de contrôle $\mathbf{v}_i^{(t+1)}$ est choisi et l'équation (3.7) détermine le déplacement $\mathbf{d}_i^{(t)} = (1, 1)$. Dans la figure (3.6.c), le pixel choisi étant sur la périphérie de $N_i^{(t)}$, la nouvelle grille de voisinage $N_i^{(t+1)}$ est déterminée et correspond à une version décalée de $N_i^{(t)}$ de dimensions inchangées, avec $\mathbf{s}_i^{(t+1)} = (1, 1)$. La figure (3.6.d) illustre le déplacement de $\mathbf{v}_j^{(t)}$, déterminant $\mathbf{d}_j^{(t)} = (1, 1)$. Dans la figure (3.6.e), le pixel choisi n'est pas sur la périphérie de $N_j^{(t)}$, les nouvelles dimensions de la grille deviennent alors $L_j^{(t+1)} = 6$ et $H_j^{(t+1)} = 6$. Le vecteur de décalage devient $\mathbf{s}_j^{(t+1)} = (2, 2)$. $N_j^{(t+1)}$ correspond alors à une version déformée de $N_j^{(t)}$.

Ainsi, le déplacement de $\mathbf{v}_i^{(t)}$ ayant été considéré fiable, $N_i^{(t+1)}$ favorise fortement les pixels dans cette direction. En revanche, le déplacement de $\mathbf{v}_j^{(t)}$ étant moins fiable, $N_j^{(t+1)}$ se déforme dans cette direction, mais sans pour autant défavoriser les directions opposées.

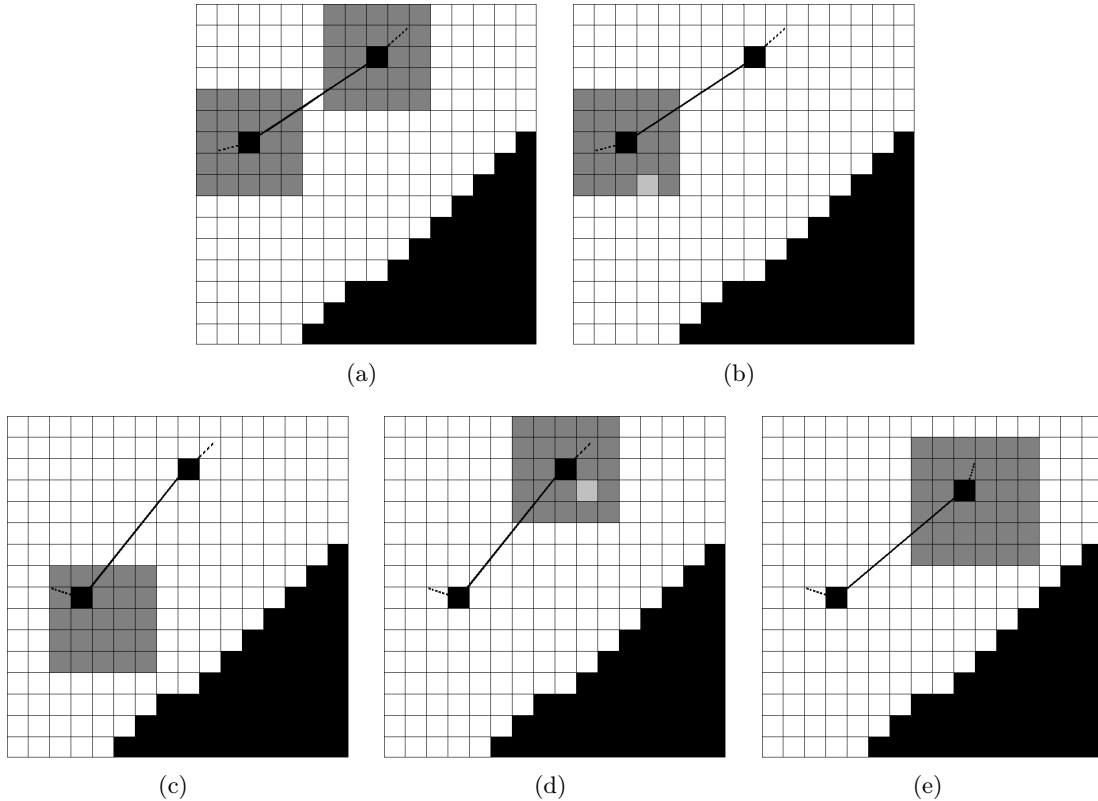


FIG. 3.6 – Illustration de la méthode d'anticipation par déformation pour deux points de contrôle.

3.2 Contours actifs supervisés pour la segmentation d'images texturées par ensembles de niveaux

Alors que la première partie de ce chapitre était consacrée aux méthodes permettant d'accélérer les contours actifs représentés de manière paramétrique, cette seconde partie va s'intéresser aux méthodes développées pour améliorer la précision des modèles implicites.

En effet, conservant le raisonnement décrit au 3.1, il nous a paru intéressant de développer des approches permettant d'améliorer les propriétés des modèles implicites dans le domaine où ils sont les plus efficaces : la précision de segmentation.

Parmi les différents types d'images existants, les images texturées font partie des plus complexes à appréhender. Leur segmentation nécessite souvent une approche de haut niveau s'appuyant sur des caractéristiques appelées descripteurs de textures.

Comme établi au chapitre 2, il existe un grand nombre de descripteurs, mais afin de permettre aux différents modèles de contours actifs proposés ici d'être efficaces, il a été nécessaire de déterminer quel type de descripteur de textures allait être utilisé. Plusieurs contraintes se sont alors imposées.

- La plupart des modèles de contours actifs évolue en fonction des pixels présents dans le voisinage de la courbe, ou dans le cas des approches région, en fonction de descripteurs calculés sur les régions déterminées par la position de la courbe (régions qui évoluent à mesure que le contour se déplace). Il est donc nécessaire d'utiliser des descripteurs calculables aussi bien localement au niveau du pixel que sur une région entière.
- Même si l'utilisation d'un descripteur entraîne nécessairement une augmentation de la complexité du modèle, la segmentation doit s'effectuer en un temps raisonnable. L'algorithme de calcul du descripteur doit donc être de faible complexité.
- La segmentation d'images texturées est un domaine très actif aussi bien pour les images 2D que 3D. En conséquence, le descripteur choisi doit être adaptable à la 3D sans pour autant voir sa complexité augmenter de manière significative.

Après une étude des principaux descripteurs présents dans la littérature, il a été décidé de développer des modèles de contours actifs basés sur les coefficients d'Haralick calculés à partir de la matrice de cooccurrence. En effet, comme il est détaillé au 2.2.1.1, cette famille de descripteurs permet de satisfaire toutes les contraintes présentées ci-dessus. L'intérêt des coefficients d'Haralick a été démontrée, ce sont des descripteurs rapides à calculer qui, quand ils sont déterminés à l'aide d'une fenêtre de voisinage, deviennent des descripteurs de textures locaux permettant de caractériser chaque pixel de l'image. De plus, ils sont applicables en tant que descripteurs de régions quelle que soit la forme de la région.

Il existe néanmoins un inconvénient à l'utilisation des descripteurs de textures : ils ne possèdent généralement pas le même pouvoir discriminant d'un type d'image à un autre. Cela s'applique également aux coefficients d'Haralick.

En conséquence, afin de permettre aux modèles proposés ici d'être peu sensibles au changement de type d'image texturée, ceux-ci ont été incorporés dans un processus de segmentation supervisé.

Dans cette partie, deux approches différentes et innovantes pour la segmentation

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

d'images texturées par contour actif supervisé sont présentées. Leur principe général est d'utiliser une image d'apprentissage possédant une segmentation experte afin de réaliser une étape d'apprentissage du modèle. Le premier modèle, inspiré du contour actif de Chan et Vese [Chan et Vese, 2001], utilise une approche par programmation linéaire afin de déterminer, à l'aide d'une image d'apprentissage, le pouvoir discriminant de chaque coefficient d'Haralick. Cette étape d'apprentissage est alors utilisée pour calculer un poids pour chaque coefficient. Le deuxième modèle présenté possède la particularité d'être entièrement guidé par un classificateur binaire supervisé lors de la segmentation, le classificateur étant introduit dans l'équation d'évolution en utilisant l'implémentation sans résolution des EDP développée dans [Shi et Karl, 2005a]. Le but de l'étape d'apprentissage est alors de déterminer les différents paramètres du classificateur binaire.

3.2.1 Contour actif supervisé basé sur la programmation linéaire pour la segmentation d'images texturées

Le contour actif pour la segmentation de textures développé est inspiré du modèle de Chan et Vese introduit dans [Chan et Vese, 2001, Chan *et al.*, 2000] (voir au 1.2.2.3 pour plus de détails). L'utilisation des coefficients d'Haralick entraînant une dépendance au type d'image traité, le modèle est introduit dans un processus de segmentation supervisé s'appuyant sur la programmation linéaire.

Dans un premier temps, nous allons présenter le modèle de contour actif basé texture. Dans un deuxième temps, le principe général de la programmation linéaire sera détaillé et enfin le modèle de programmation linéaire développé dans le but de pondérer les différents coefficients d'Haralick guidant le contour actif ainsi que le processus complet de segmentation seront décrits.

3.2.1.1 Modèle de contour actif implicite pour la segmentation d'images texturées

Le modèle présenté ici a été développé afin de résoudre le problème de segmentation d'images texturées par contour actif. L'évolution de la courbe dans l'image se base sur la fonctionnelle d'énergie du modèle de Chan et Vese mais au lieu de s'appuyer sur les niveaux de gris présents dans l'image, celle-ci va utiliser m coefficients d'Haralick pour chaque pixel de l'image. Afin d'être en mesure de déterminer l'ensemble des coefficients pour un pixel donné, une fenêtre de voisinage de 11 pixels de large est construite autour de celui-ci. La matrice de cooccurrence le caractérisant peut alors être déterminée puis les différents coefficients d'Haralick calculés. La valeur de la transition U a été choisie telle que $d = 1$ et $\theta = 0$.

Le modèle de contour actif pour la segmentation de textures basé sur m coefficients

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

d'Haralick va donc se déformer de manière à minimiser la fonctionnelle d'énergie :

$$\begin{aligned}
E(\mathbf{C}, \overline{k_j^{in}}, \overline{k_j^{out}}) = & \mu \cdot \text{Longueur}(\mathbf{C}) \\
& + \int_{\text{Intérieur}(\mathbf{C})} \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{in}} \right|^2 d\mathbf{x} \\
& + \int_{\text{Extérieur}(\mathbf{C})} \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{out}} \right|^2 d\mathbf{x} . \quad (3.16)
\end{aligned}$$

$\overline{k_j^{in}}$ et $\overline{k_j^{out}}$ sont les valeurs moyennes du $j^{\text{ème}}$ coefficient d'Haralick pour les régions respectivement à l'intérieur et à l'extérieur de $\mathbf{C}$. $k_j(\mathbf{x})$ est la valeur du $j^{\text{ème}}$ coefficient d'Haralick pour le pixel $\mathbf{x}$. w_j est le poids associé au $j^{\text{ème}}$ coefficient d'Haralick, permettant de pondérer son influence lors de la segmentation. $\text{Longueur}(\mathbf{C})$ est un terme de régularisation similaire à celui introduit dans [Chan et Vese, 2001]. Les deux intégrales correspondent au terme d'attache aux données et sont utilisées afin d'évaluer la position de la courbe dans l'image. Elles sont déterminées par les m coefficients de textures d'Haralick. Leur comportement est équivalent à celui des fonctions E_1 et E_2 basées sur les moyennes d'intensités décrites dans l'équation (1.67).

Afin de représenter le modèle de manière implicite, la fonction d'ensembles de niveaux est introduite dans la fonctionnelle d'énergie en utilisant les versions lissées des fonctions de Heaviside et Dirac, $H_\epsilon(\phi(\mathbf{x}))$ et $\delta_\epsilon(\phi(\mathbf{x}))$:

$$\begin{aligned}
E_\epsilon(\phi, \overline{k_j^{in}}, \overline{k_j^{out}}) = & \mu \int_{\Omega} \delta_\epsilon(\phi(\mathbf{x})) |\nabla \phi(\mathbf{x})| d\mathbf{x} \\
& + \int_{\Omega} \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{in}} \right|^2 H_\epsilon(\phi(\mathbf{x})) d\mathbf{x} \\
& + \int_{\Omega} \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{out}} \right|^2 H_\epsilon(\phi(\mathbf{x})) d\mathbf{x} . \quad (3.17)
\end{aligned}$$

Ainsi, minimiser l'équation (3.17) par rapport à ϕ en utilisant les équations d'Euler-Lagrange permet d'obtenir l'équation aux dérivées partielles (EDP) régissant l'évolution du modèle représenté avec les ensembles de niveaux (voir l'annexe B pour les détails de la démonstration) :

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \delta_\epsilon(\phi) \left[\mu \cdot \text{div} \frac{\nabla \phi}{|\nabla \phi|} - \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{in}} \right|^2 + \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{out}} \right|^2 \right] . \quad (3.18)$$

L'évolution de la courbe $\mathbf{C}$ est alors déterminée par l'observation du niveau zéro de $\phi(\mathbf{x}, t)$ à chaque étape de sa déformation.

L'équation d'évolution (3.18) peut également être déterminée en intégrant le modèle dans le schéma général des contours actifs basés région défini dans [Jehan-Besson *et al.*, 2003] et présenté au 1.2.2.5. Les différents paramètres du formalisme

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

proposé dans [Jehan-Besson *et al.*, 2003] deviennent alors :

$$\begin{cases} k^{(in)} &= - \sum_{j=1}^m w_j \left| k_j(\mathbf{s}) - \overline{k_j^{in}} \right|^2 \\ k^{(out)} &= - \sum_{j=1}^m w_j \left| k_j(\mathbf{s}) - \overline{k_j^{out}} \right|^2 \\ k^{(b)} &= \mu , \end{cases} \quad (3.19)$$

avec s l'abscisse curviligne de $\mathbf{C}$.

Le descripteur de contour restant constant, sa dérivée selon $\mathcal{C}$ devient nulle et l'équation d'évolution du modèle peut alors s'exprimer comme :

$$\frac{\partial C(s, t)}{\partial t} = \left[- \sum_{j=1}^m w_j \left| k_j(\mathbf{s}) - \overline{k_j^{in}} \right|^2 + \sum_{j=1}^m w_j \left| k_j(\mathbf{s}) - \overline{k_j^{out}} \right|^2 + \mu \cdot k \right] \mathbf{N} . \quad (3.20)$$

Il est alors possible de déterminer l'équation d'évolution du modèle représenté à l'aide des ensembles de niveaux. Sa seule différence avec l'équation (3.18) est que la courbe se déforme selon $|\nabla \phi|$ au lieu de $\delta_\epsilon(\phi)$:

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \left[\mu \cdot \text{div} \frac{\nabla \phi}{|\nabla \phi|} - \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{in}} \right|^2 + \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{out}} \right|^2 \right] \cdot |\nabla \phi| . \quad (3.21)$$

Le modèle ainsi défini est efficace mais souffre d'une trop grande sensibilité au réglage des poids w_j , car ceux-ci vont devoir varier d'un type d'image à un autre pour garantir une segmentation correcte. Pour permettre au modèle de s'affranchir de leur réglage, il a été choisi de déterminer les valeurs des w_j les plus efficaces en utilisant l'information présente dans une image d'apprentissage. L'utilisation d'un modèle de programmation linéaire nous est apparu appropriée car celui-ci possède la capacité de déterminer la valeur optimale de plusieurs variables tout en respectant un objectif fixé. Cette approche a donc été utilisée pour modéliser le comportement du contour actif basé texture sur l'image d'apprentissage et définir ainsi les valeurs optimales des w_j .

La section suivante présente le principe général de la programmation linéaire.

3.2.1.2 Programmation linéaire

Les problèmes de programmation linéaire simple (PLS), introduit dans [Dantzig, 1963], peuvent être considérés comme une manière spécifique de formuler des problèmes d'optimisation à variables réelles. Modéliser un tel problème comme un PLS permet d'avoir accès à certaines méthodes de résolution optimales (i.e. méthodes permettant d'obtenir une solution optimale du problème). Actuellement, cette approche est principalement employée pour déterminer des bornes inférieures de problèmes linéaires en nombres entiers (PLNE) [Neumaier et Shcherbina, 2004], ou pour obtenir des solutions approchées à des problèmes de taille très importante [Wagner *et al.*, 2004, Babonneau *et al.*, 2006]. A l'origine, la programmation linéaire était principalement utilisée dans le domaine de la

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

recherche opérationnelle, mais à l'heure actuelle cette méthode de résolution est appliquée dans de nombreuses autres thématiques [Applegate *et al.*, 2007, Yanover *et al.*, 2006, Feldman *et al.*, 2005, Kingsford *et al.*, 2004].

Dans cette section, nous allons dans un premier temps décrire la formulation habituelle de programmation linéaire puis nous réaliserons un bref état de l'art des méthodes de résolutions connues.

Le principe de la programmation linéaire consiste à utiliser des *données* afin de déterminer un certain nombre de *variables* soumises à un ensemble de *contraintes* linéaires d'égalité ou d'inégalité de manière à optimiser une *fonction objectif* linéaire. La spécification de l'ensemble {données, variables, contraintes, fonction objectif} définit le problème à résoudre.

Ainsi, soit $\mathbf{x}$ le vecteur composé de l'ensemble des n variables du problème, $\mathbf{x} \in \mathbb{R}^n$, et $\mathbf{c} \in \mathbb{R}^n$ le vecteur des coefficients de la fonction objectif $f(\mathbf{x})$ définie telle que :

$$f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} . \quad (3.22)$$

Chacune des m contraintes du problème est répartie dans deux matrices différentes. Pour la $i^{\text{ème}}$ contrainte du problème, la partie gauche de l'inégalité (ou de l'égalité) correspond à la ligne i dans la matrice des contraintes notée $\mathbf{A} \in \mathbb{R}^m \times \mathbb{R}^n$, alors que la partie droite détermine la $i^{\text{ème}}$ valeur dans le vecteur $\mathbf{b} \in \mathbb{R}^m$.

Ainsi, l'ensemble des contraintes du problème est défini par :

$$\mathbf{Ax} \leq \mathbf{b} . \quad (3.23)$$

Le problème complet peut alors être exprimé sous sa forme canonique :

$$\begin{aligned} \max f(\mathbf{x}) &= \mathbf{c}^T \mathbf{x} \\ \text{sous les contraintes } &\mathbf{Ax} \leq \mathbf{b} . \end{aligned}$$

Résoudre un tel problème revient donc à déterminer les valeurs de $\mathbf{x}$ maximisant $f(\mathbf{x})$, tout en respectant l'ensemble des contraintes définies par $\mathbf{Ax} \leq \mathbf{b}$.

D'un point de vue géométrique, l'ensemble des contraintes linéaires définit l'ensemble des solutions réalisables comme un polytope convexe (un polytope est un polygone de dimension supérieure à 3). Deux conditions sont nécessaires afin d'assurer une solution : les contraintes ne doivent pas être en opposition et le polytope doit être borné dans la direction de la fonction objectif. Etant donnée la linéarité de la fonction objectif, la solution optimale correspond à un des sommets du polytope.

L'algorithme du simplexe, développé dans [Dantzig, 1963], a été le premier capable de résoudre les problèmes de programmation linéaire. Cet algorithme construit une première solution réalisable correspondant à un sommet du polytope, puis se déplace de sommet en sommet en passant par les différentes arêtes accessibles. A chaque sommet (correspondant à la solution courante), l'algorithme choisit, comme destination, un sommet accessible permettant d'augmenter la valeur de la fonction objectif (ou de la diminuer, selon que l'on cherche à maximiser ou minimiser $f(\mathbf{x})$). Cet algorithme est efficace en pratique et permet de déterminer la solution optimale du problème à condition qu'aucun cycle ne soit créé dans le choix des sommets parcourus. Il a néanmoins été prouvé dans [Klee et Minty, 1972]

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

qu'il était possible de construire un problème de programmation linéaire nécessitant un nombre exponentiel d'étapes de l'algorithme du simplexe pour être résolu.

Le premier algorithme permettant une résolution de problèmes de programmation linéaire en un temps polynomial est apparu dans [Khachiyan, 1979]. Cette approche est basée sur la méthode des ellipses, une méthode de résolution non linéaire proposée dans [Shor, 1970]. La méthode des ellipses a été généralisée au domaine de l'optimisation convexe dans [Yudin et Nemirovskii, 1976]. La complexité dans le pire des cas est de $O(n^6 L)$, avec n le nombre de variables et L le nombre de bits en entrée, mais en pratique l'algorithme du simplexe reste plus efficace.

Le premier algorithme polynomial réellement efficace a été proposé dans [Karmarkar, 1984], introduisant la famille des approches par méthode du point intérieur. Au lieu de parcourir les différentes arêtes du polytope, l'idée principale est d'accéder au sommet représentant la solution optimale en construisant un point à l'intérieur du polytope à chaque itération. Cette méthode possède une complexité dans le pire des cas en $O(n^{3.5} L)$ et reste en pratique au moins aussi efficace que l'algorithme du simplexe.

De par leur capacité à déterminer la valeur optimale de plusieurs variables, les modèles de programmation linéaire nous sont apparus comme une excellente solution pour déterminer les coefficients w_j du modèle de contour actif basé texture. La section suivante présente le modèle de programmation linéaire construit dans cet objectif.

3.2.1.3 Modèle de programmation linéaire pour contour actif supervisé

Afin d'être en mesure de déterminer les poids w_j du contour actif de manière supervisée, un modèle de programmation linéaire a été développé. Ce dernier modélise le comportement du contour actif basé texture décrit au 3.2.1.1. Avant de détailler le modèle mathématique, il est nécessaire d'expliquer de quelle manière ses différentes données sont obtenues.

Extraction de l'ensemble d'apprentissage

Le caractère supervisé du processus de segmentation rend nécessaire une connaissance *a priori* sur les données traitées. Dans notre approche, celle-ci se compose d'une image d'apprentissage Ω^* dans laquelle une courbe C^* représentant une segmentation optimale est présente. Le but du modèle de programmation linéaire va donc être de déterminer les poids w_j permettant à un contour actif, défini par la fonctionnelle d'énergie (3.16), de réaliser la même segmentation que C^* (et de généraliser à des images semblables).

Tel qu'il a été présenté au 3.2.1.1, le modèle de contour actif basé texture proposé cherche à réaliser la partition de l'image Ω_I en deux régions possédant des caractéristiques de textures les plus homogènes possible. En conséquence, les données nécessaires au modèle de programmation linéaire sont les m coefficients d'Haralick caractérisant les deux régions créées par C^* , notées C^{in} et C^{out} . Ces données vont être utilisées pour déterminer les coefficients moyens $\overline{k_j^{in}}$ et $\overline{k_j^{out}}$ de l'équation (3.18) et également dans la principale contrainte du modèle de programmation linéaire.

Intuitivement, et afin de modéliser au mieux le comportement du contour actif, C^{in} devrait être définie comme la région à l'intérieur de C^* et C^{out} comme le reste de l'image,

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

tel que $\Omega^* = C^{in} \cup C^{out}$. Mais comme il a été énoncé dans [Mille *et al.*, 2007], la région à l'extérieur de l'objet segmenté n'est que très rarement homogène et contient même parfois des textures très similaires à celle de l'objet recherché. De plus, même si ce phénomène n'apparaît pas dans l'image d'apprentissage Ω^* , il peut encore être présent dans les autres images de la série. Cela entraînera alors une grande variabilité de la texture de C^{out} sur toute la série d'images, rendant difficile le travail de classification réalisé par le modèle de programmation linéaire.

Aussi, afin de maximiser la distance entre les textures de C^{in} et de C^{out} , garantissant un plus grand pouvoir discriminant aux caractéristiques d'Haralick, et de conserver des propriétés de textures les plus constantes possibles pour chacune des deux régions, C^{in} est définie comme la région intérieure à C^* , et C^{out} comme une bande étroite fixée par l'expert autour de C^{in} (ainsi $C^* \in C^{out}$).

Une fois cette étape réalisée, les m coefficients d'Haralick des n pixels composant C^{in} et C^{out} sont calculés pour créer un ensemble d'apprentissage $\mathbf{X}$ composé de $n = n_1 + n_2$ individus, tel que n_1 pixels appartiennent à C^{in} et n_2 à C^{out} (les paramètres de la matrice de cooccurrence sont analogues à ceux détaillés au 3.2.1.1). Chaque individu de cet ensemble est alors défini comme :

$$\begin{aligned} s(i) &= \{k_1(i), k_2(i) \dots k_m(i), l_i\} \\ l_i &= \begin{cases} -1 & \text{si } i \text{ appartient à } C^{in} \\ +1 & \text{si } i \text{ appartient à } C^{out} \end{cases} \end{aligned} \quad (3.24)$$

où $k_1(i), k_2(i) \dots k_m(i)$ sont les m coefficients d'Haralick de l'individu i et l_i une étiquette représentant son appartenance à C^{in} ou C^{out} . La création de C^{in} , C^{out} et de l'ensemble d'apprentissage est illustrée figure (3.7).

Calcul des w_j par programmation linéaire

Toutes les données nécessaires au modèle de programmation linéaire ne sont pas contenues dans l'ensemble d'apprentissage $\mathbf{X}$ et certaines doivent être calculées.

Tout d'abord, il est nécessaire de déterminer les valeurs moyennes de chaque coefficient d'Haralick pour les régions C^{in} et C^{out} , notées $\overline{k_j^{in}}$ et $\overline{k_j^{out}}$.

Le but du modèle de programmation linéaire est de modéliser le comportement du contour actif basé texture proposé. Ainsi, en s'inspirant du terme d'attache aux données de l'équation (3.16), les données du modèle peuvent être complétées en définissant $D_j(i, C^{in})$ et $D_j(i, C^{out})$, les distances de l'individu i à chacune des deux classes C^{in} et C^{out} relativement au $j^{\text{ème}}$ coefficient d'Haralick :

$$\begin{aligned} D_j(i, C^{in}) &= (k_j(i) - \overline{k_j^{in}})^2 \\ D_j(i, C^{out}) &= (k_j(i) - \overline{k_j^{out}})^2 \end{aligned} \quad (3.25)$$

Les variables à déterminer sont les poids w_j contrôlant l'influence de chaque coefficient d'Haralick. Si seul le terme d'attache aux données de l'équation (3.18) est pris en compte, alors modéliser l'affectation d'un pixel à C^{in} ou C^{out} revient à comparer ses distances à

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

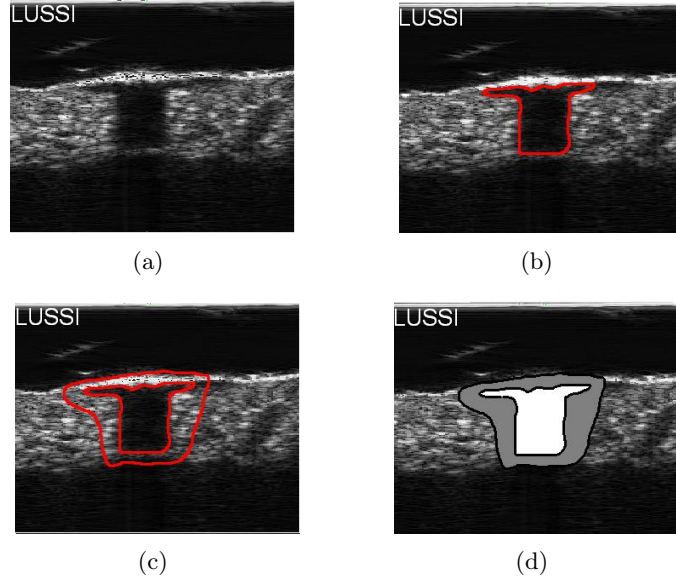


FIG. 3.7 – Image d'apprentissage pour une série d'images échographiques. a) Image d'origine b) Définition de C^{in} , la région intérieure de l'objet. c) Définition de C^{out} , la région extérieure de l'objet (région frontière). d) Les coefficients d'Haralick des pixels définissant C^{in} (en blanc) et C^{out} (en gris) sont déterminés afin de composer l'ensemble d'apprentissage $\mathbf{X}$.

chacune de ces deux classes. Ainsi, en exprimant cette distance en fonction des coefficients d'Haralick, la règle de classification se traduit dans le modèle de programmation linéaire par l'ensemble d'inégalités :

$$\begin{cases} \sum_{j=1}^m (w_j (D(i, C^{out}) - D(i, C^{in}))) \geq 0 & si \quad i \in C^{in} \\ \sum_{j=1}^m (w_j (D(i, C^{out}) - D(i, C^{in}))) < 0 & si \quad i \in C^{out} . \end{cases} \quad (3.26)$$

Le modèle doit être capable d'évaluer la qualité de sa décision sur les w_j . Cette évaluation peut être réalisée en définissant une mesure d'estimation du nombre d'individus bien classés. En utilisant la règle de classification (3.26), il est possible de définir pour chaque individu i la fonction de décision d_i suivante, correspondant alors à l'unique contrainte du modèle de programmation linéaire :

$$d_i = \sum_{j=1}^m w_j (D(i, C^{in}) - D(i, C^{out})) \cdot l_i . \quad (3.27)$$

d_i sera positive si l'individu i a été affecté correctement et négative sinon. En effet, si l'individu i appartient à la région C^{in} alors $l_i = -1$. Si le modèle de programmation linéaire prend la même décision alors la distance de l'individu i à C^{in} (pondérée par les w_j déterminés par le modèle) est inférieure à sa distance à C^{out} , entraînant alors un d_i positif.

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

A l'opposé, si le modèle prend une décision différente et affecte i à la région C^{out} , alors sa distance à C^{in} sera supérieure à sa distance à C^{out} et d_i sera négatif.

Il est désormais possible d'exprimer la fonction objectif du modèle. Le but est de déterminer les w_j permettant d'obtenir une affectation des différents individus de l'ensemble d'apprentissage la plus juste possible. Un individu bien classé se traduisant par une fonction d'évaluation d_i positive, réaliser la meilleure affectation globale revient à maximiser la somme des d_i . Ainsi, le modèle va chercher les w_j maximisant le nombre d'individus bien classés.

Finalement, le modèle de programmation linéaire complet est donné par :

Données

- m : le nombre de coefficients d'Haralick
- n : le nombre d'individus dans l'ensemble d'apprentissage
- $\bar{k}_j^{in}, \forall j = 1 \dots m$: la valeur moyenne du $j^{ème}$ coefficient d'Haralick pour la classe C^{in}
- $\bar{k}_j^{out}, \forall j = 1 \dots m$: la valeur moyenne du $j^{ème}$ coefficient d'Haralick pour la classe C^{out}
- $k_j(i), \forall i = 1 \dots n, \forall j = 1 \dots m$: la valeur du $j^{ème}$ coefficient d'Haralick de l'individu i
- $l_i, \forall i = 1 \dots n$: l'affectation de l'individu i dans l'ensemble d'apprentissage
- $D_j(i, C^{in}), \forall i = 1 \dots n, \forall j = 1 \dots m$: la distance entre l'individu i et la classe C^{in} relativement au $j^{ème}$ coefficient d'Haralick
- $D_j(i, C^{out}), \forall i = 1 \dots n, \forall j = 1 \dots m$: la distance entre l'individu i et la classe C^{out} relativement au $j^{ème}$ coefficient d'Haralick

Variables

- $w_j \in \mathbb{R}^+, \forall j = 1 \dots m$: le poids du $j^{ème}$ coefficient d'Haralick
- $d_i \in [-1, 1] \subset \mathbb{R}, \forall i = 1 \dots n$: la variable de décision pour l'individu i . Si $d_i > 0$ alors les w_i permettent l'affectation de i à la bonne région

Contrainte

$$\forall i = 1 \dots n, d_i = \sum_{j=1}^m w_j (D(i, C^{out}) - D(i, C^{in})) \cdot l_i$$

Fonction objectif

$$\max \sum_{i=1}^n d_i$$

Le modèle mathématique de programmation linéaire ainsi exprimé est fourni à un programme de résolution qui construit la forme canonique du problème puis utilise une méthode d'optimisation afin de déterminer la solution optimale (correspondant aux w_j optimaux).

Processus complet de segmentation

Le but de l'approche supervisée proposée ici est de déterminer un jeu de paramètres permettant à un unique modèle de contour actif basé texture de segmenter une série complète

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

d'images, chacune possédant des propriétés texturales proches.

Dans un premier temps, une étape interactive est réalisée afin de déterminer dans une image d'apprentissage une courbe C^* représentant une segmentation experte et les deux régions C^{in} et C^{out} la caractérisant (idéalement cette étape doit être réalisée par un spécialiste du type d'images traitées). Chaque pixel appartenant à ces deux régions est alors caractérisé par m coefficients d'Haralick, permettant ainsi la création de l'ensemble d'apprentissage.

Dans un deuxième temps, le modèle mathématique de programmation linéaire est résolu. Celui-ci nous permet d'obtenir, pour chaque coefficient d'Haralick, le poids w_j à utiliser dans l'équation d'évolution du contour actif.

Enfin, une fois tous ses paramètres déterminés, le contour actif peut être utilisé afin de segmenter la série d'images possédant des propriétés de textures similaires à celles de l'image d'apprentissage.

La figure (3.8) illustre les différentes étapes de la segmentation.

3.2.1.4 Analyse du comportement du modèle de programmation linéaire et de son influence sur la segmentation obtenue par le contour actif

Le modèle de contour actif pour la segmentation de textures présenté dans cette partie est intégré dans un processus de segmentation supervisé s'appuyant sur la programmation dynamique afin de déterminer l'influence de chaque coefficient d'Haralick. Le modèle de programmation dynamique peut donc être considéré comme un processus d'extraction de caractéristiques car les w_j déterminés sont utilisés dans le contour actif pour déterminer une unique caractéristique par pixel en réalisant une combinaison linéaire des coefficients d'Haralick. En pratique, de nombreux w_j sont fixés à 0 par le modèle (seuls deux à trois coefficients d'Haralick sont généralement considérés discriminants et voient leurs w_j non nuls). Ce comportement s'apparente alors à un processus de sélection de caractéristiques.

En outre, le modèle utilise un ensemble d'apprentissage afin de déterminer les poids optimaux des coefficients d'Haralick permettant au maximum d'individus (i.e. de pixels) d'être affectés à leur région d'origine (C^{in} ou C^{out}). Ce comportement est similaire à celui d'un classificateur linéaire binaire supervisé.

L'utilisation de la programmation linéaire pour régler les w_j modifie le comportement du contour actif. En effet, tel qu'il est décrit au 3.2.1.1, celui-ci cherche à créer une partition de l'image en deux régions possédant des propriétés de textures homogènes. Une fois intégré dans le processus supervisé, le contour se déforme alors de manière à créer une région intérieure à la courbe possédant des caractéristiques proches de la région C^{in} et utilise l'information de la texture de C^{out} pour évaluer la position de la frontière de C^{in} . La segmentation réalisée sur l'image d'apprentissage prend alors une grande importance car elle détermine l'ensemble d'apprentissage sur lequel seront basées les segmentations automatiques réalisées sur les images suivantes de la série.

Le modèle proposé a été testé sur des images échographiques 2D et ses performances ont été comparées à celles du modèle sur lequel il est basé, le contour actif de Chan et Vese. Ces tests sont détaillés au 4.5.

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

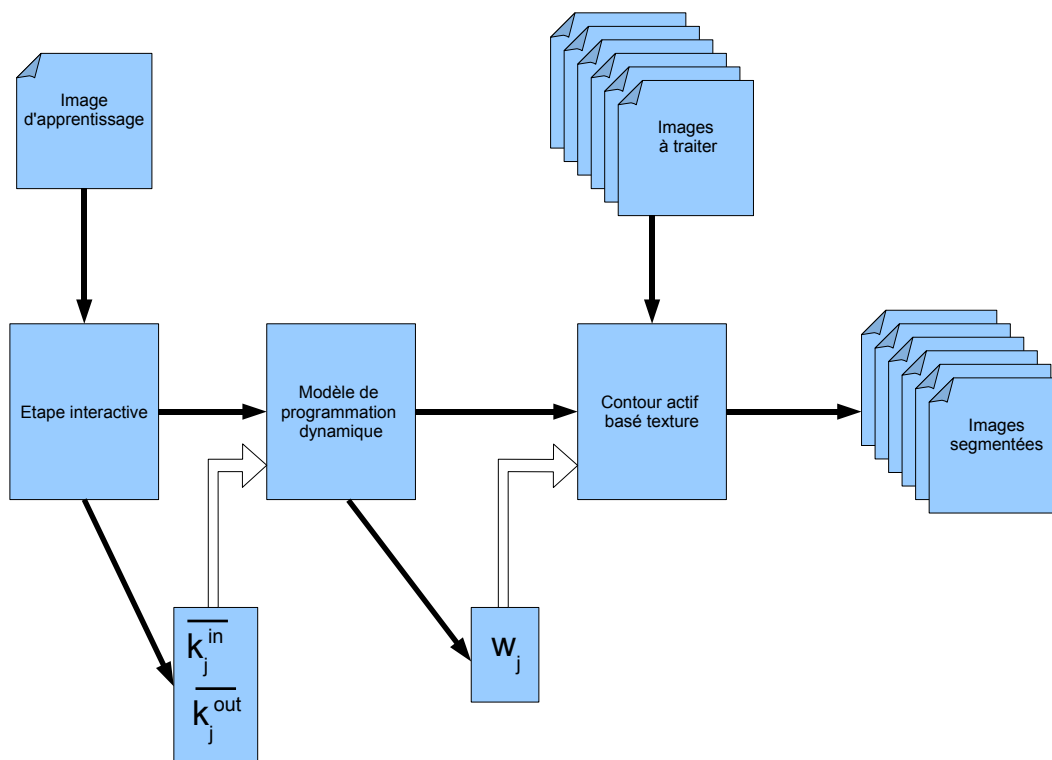


FIG. 3.8 – Processus complet de segmentation : les coefficients d'Haralick $\overline{k_j^{in}}$ et $\overline{k_j^{out}}$ sont extraits des régions C^{in} et C^{out} de l'image d'apprentissage puis réinjectés dans le modèle de programmation linéaire. Celui-ci détermine alors les poids optimaux w_j et les fournit au modèle de contour actif qui sera utilisé pour segmenter toute la série d'images.

3.2.2 Contour actif guidé par classificateur binaire supervisé pour la segmentation de textures

L'originalité du contour actif supervisé présenté dans la section précédente est d'utiliser un modèle de programmation linéaire et de transformer celui-ci en classificateur linéaire supervisé afin de déterminer la région d'appartenance de chaque pixel de l'image. En effet, le problème d'optimisation s'appuie sur un ensemble d'apprentissage composé d'individus (les pixels), chacun défini par des caractéristiques (les coefficients d'Haralick) et une classe d'appartenance (C^{in} ou C^{out}). La résolution du modèle de programmation linéaire est équivalente à un algorithme d'apprentissage, son but étant de déterminer les différents paramètres du classificateur (les w_j dans notre cas). Enfin, une fois le modèle appris, une règle de classification permet l'affectation de n'importe quel individu à une classe de manière déterministe (l'équation (3.26)).

Un tel comportement est celui d'un classificateur supervisé. Le modèle proposé met

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

ainsi en évidence la viabilité d'une approche par classificateur pour les contours actifs. Néanmoins, le caractère linéaire de la programmation dynamique peut apparaître comme un frein à l'affectation correcte des pixels de l'image, et donc à la précision du résultat de segmentation, car les caractéristiques de textures des deux régions composant l'image d'apprentissage ne les rendent pas nécessairement linéairement séparables.

Dès lors, il devient naturel de penser à remplacer le modèle de programmation linéaire par un véritable classificateur binaire, en particulier un classificateur non-linéaire. Ce principe est à la base du deuxième modèle de contour actif supervisé développé. Ce dernier est en effet entièrement guidé par un classificateur binaire supervisé lors de l'étape de segmentation.

Trois classificateurs différents ont ainsi été introduits dans le contour actif et comparés. Le premier classificateur est l'algorithme des k-PPV, choisi en raison de sa simplicité d'implémentation. Le deuxième est un modèle de machines à vecteurs supports (SVM) qui possède une grande capacité à classer des données non-linéairement séparables dans l'espace des caractéristiques. Enfin le troisième classificateur utilisé est un réseau de neurones, connu comme approximateur universel de fonctions.

La section suivante décrit le principe de la classification binaire supervisée ainsi que la création de l'ensemble d'apprentissage utilisé par le modèle. La section 3.2.2.2 présente le processus permettant d'introduire la réponse d'un classificateur directement dans l'équation d'évolution d'un contour actif implicite, la section 3.2.2.3 détaille le principe des trois classificateurs précédemment cités et enfin la section 3.2.2.4 décrit le processus complet de segmentation.

3.2.2.1 Classification binaire supervisée

Le principe de la classification binaire supervisée est basé sur l'observation d'un ensemble d'apprentissage $\mathbf{X}$ composé de n individus tels que $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$. Les m caractéristiques des individus $\mathbf{x}_i \in \mathbb{R}^m$, $i \in \{1, \dots, n\}$ ainsi que l'ensemble de leurs classes d'appartenance $\mathbf{L} = \{l_1, \dots, l_n\}$, $l_i \in \{-1, 1\}$ sont connus et utilisés afin de définir une règle de classification $\hat{l} = f(\mathbf{x})$ capable de déterminer la classe $\hat{l}$ de tout individu $\mathbf{x} \in \mathbb{R}^m$.

Pour notre approche, l'ensemble d'apprentissage est constitué de manière identique à celle décrite au 3.2.1.3 : deux régions C^{in} et C^{out} sont extraites d'une image d'apprentissage, chaque pixel les composant est alors caractérisé par ses coefficients d'Haralick (d'une manière identique à celle décrite au 3.2.1.1). Ainsi, $\mathbf{X}$ reste défini par l'équation (3.24).

En revanche, l'algorithme d'apprentissage ne sera pas appliqué sur tout l'ensemble $\mathbf{X}$. En effet, un classificateur nécessite généralement le réglage de plusieurs paramètres, ces derniers influençant fortement le résultat de la classification. Aussi, il a été décidé de séparer $\mathbf{X}$ en deux sous-ensembles $\mathbf{X}_l$ et $\mathbf{X}_e$ tels que $\mathbf{X} = \mathbf{X}_l \cup \mathbf{X}_e$. Chaque classificateur est appris selon différentes configurations de paramètres variant de manière régulière sur l'ensemble $\mathbf{X}_l$, puis évalué sur l'ensemble d'évaluation $\mathbf{X}_e$. La configuration de paramètres ayant obtenu le meilleur taux de classification sur $\mathbf{X}_e$ permet alors de déterminer le classificateur introduit dans le modèle supervisé de contour actif. Afin de conserver un grand nombre d'individus pour l'apprentissage, la proportion d'individus dans chaque sous-ensemble est telle que $Card(\mathbf{X}_l) = \frac{2}{3}Card(\mathbf{X})$ et $Card(\mathbf{X}_e) = \frac{1}{3}Card(\mathbf{X})$ (l'ensemble $\mathbf{X}$ possédant en

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

pratique environ 10000 individus, une proportion de 1/3 est suffisante pour évaluer les classificateurs). En général, une fois les paramètres déterminés, il est recommandé de relancer un apprentissage sur l'ensemble complet $\mathbf{X}$, afin d'apprendre le maximum de données. En pratique, l'ensemble $\mathbf{X}$ est suffisamment grand pour conserver le classificateur appris uniquement avec $\mathbf{X}_l$ (correspondant à plus de 6600 individus).

3.2.2.2 Introduction d'un classificateur binaire dans l'équation d'évolution d'un contour actif

Afin d'introduire la réponse d'un classificateur dans l'équation d'évolution d'un contour actif représenté avec les ensembles de niveaux, l'implémentation rapide de Shi et Karl a été retenue [Shi et Karl, 2005a, Shi et Karl, 2005b]. Le principe de cette implémentation, détaillé au 1.1.2.2, est de faire évoluer la courbe en ne considérant que le signe de sa fonction vitesse $F(\mathbf{x})$ et deux listes de points frontière L_{in} et L_{out} . Le processus d'évolution peut alors être résumé ainsi : si $F(\mathbf{x}) > 0$ pour un point de L_{out} , la courbe est déplacée vers l'extérieur et si $F(\mathbf{x}) < 0$ pour un point de L_{in} , la courbe est déplacée vers l'intérieur. La déformation se poursuit par mises à jour successives des deux listes jusqu'à atteindre un critère d'optimalité.

L'idée principale de cette approche est donc de faire évoluer la courbe sans résoudre l'EDP complète du modèle mais en ne calculant que la valeur de la fonction vitesse $F(\mathbf{x})$ afin de déterminer son signe. En d'autres termes, le rôle de la fonction vitesse est uniquement de déterminer si un point de la courbe appartient à la région intérieure ou extérieure au contour, la gestion de la courbe par une double liste garantissant alors sa continuité et décidant s'il est nécessaire de la faire évoluer. Ainsi, le comportement de la fonction vitesse est équivalent à celui d'un classificateur binaire décidant à quelle classe (objet ou fond) doit être affecté un individu (le pixel).

Il a donc été décidé de remplacer entièrement la fonction vitesse du contour actif par la réponse d'un classificateur non-linéaire binaire supervisé, appris grâce à une image possédant une segmentation experte. Cette approche permet de combiner la précision des classificateurs non-linéaires avec les propriétés de lissage des contours actifs, car même si la fonction vitesse ne possède désormais plus de termes de courbure, une étape de lissage additionnelle est réalisée par l'implémentation de Shi et Karl, garantissant une courbure régulière de la frontière détectée.

3.2.2.3 Classificateurs utilisés

Trois classificateurs supervisés couramment utilisés ont été introduits dans l'équation d'évolution d'un contour actif et leurs performances ont été comparées sur plusieurs séries d'images.

k-Plus-Proches-Voisins

Le premier classificateur utilisé pour guider la courbe est l'algorithme des k-PPV, un des classificateurs supervisés les plus simples dans le domaine de l'apprentissage par or-

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

dinateur (*Machine Learning*). Cette approche n'utilise pas d'algorithme d'apprentissage à proprement parlé. Pour réaliser une classification, les individus de l'ensemble d'apprentissage sont simplement triés en fonction de leur distance par rapport à l'exemple à classer. La classification recherchée correspond alors à la classe la plus représentée parmi les k individus de l'ensemble d'apprentissage les plus proches de l'exemple traité, suivant le principe du vote majoritaire. Les individus de l'ensemble d'apprentissage sont généralement triés en utilisant la distance Euclidienne par rapport à l'exemple, bien que l'utilisation des distances de Manhattan, Mahalanobis ou Hamming ait déjà été observée.

Le choix du paramètre k joue un rôle important dans le processus de classification. Choisir une grande valeur pour k permettra de réduire l'influence du bruit sur la classification mais ajoutera en contrepartie un flou au niveau de la frontière entre les classes. Concernant la classification binaire, il est recommandé de choisir k impair, de manière à éviter toute égalité dans les votes. Les sous-ensembles $\mathbf{X}_l$ et $\mathbf{X}_e$ sont utilisés ici pour déterminer la meilleure valeur pour k .

Ainsi, la fonction vitesse du contour actif guidé par l'algorithme k-PPV est donnée par :

$$F(\mathbf{x}) = l_k , \quad (3.28)$$

avec l_k la classe la plus représentée parmi les k plus proches voisins de l'exemple composé des m coefficients d'Haralick du pixel $\mathbf{x}$.

Machines à vecteurs supports (SVM)

Le deuxième classificateur introduit dans l'équation d'évolution de la courbe est un modèle de machines à vecteurs supports (*Support Vector Machines*, SVM), introduites dans [Vapnik, 1995]. Les SVM font partie des classificateurs les plus récents dans le domaine de l'apprentissage par ordinateur. Leur efficacité est due à de très bonnes performances en généralisation (i.e. quand les exemples de test sont légèrement éloignés en terme de caractéristiques des individus de l'ensemble d'apprentissage) et à leur capacité à résoudre des problèmes où l'ensemble d'apprentissage $\mathbf{X}$ ne possède pas nécessairement de séparation linéaire dans l'espace des caractéristiques.

Le principe des SVM est de classer les exemples $\mathbf{x} \in \mathbb{R}^m$ en définissant un hyper-plan séparateur $h(\mathbf{x})$ déterminé par des individus particuliers de l'ensemble d'apprentissage $\mathbf{X}$ appelés vecteurs supports, représentant les individus les plus proches de la frontière. Une marge est alors définie comme la distance entre l'hyper-plan et les vecteurs supports. Lorsque les individus ne sont pas linéairement séparables dans l'espace des caractéristiques, les SVM reconsidèrent le problème en le projetant, par le biais d'une fonction noyau, dans un espace de dimension supérieure où une séparation linéaire apparaîtra (la méthode peut implicitement projeter dans un espace de dimension infinie). Dans le cas de SVM à marge souple, la méthode autorise certains individus à être à l'intérieur de la marge ou même mal classés. Afin de maximiser la marge sur l'ensemble d'apprentissage, le problème de classification est reformulé comme un problème d'optimisation quadratique, pour lequel plusieurs méthodes de résolution existent. Nous allons maintenant détailler le fonctionnement des SVM.

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

Considérons $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ un ensemble d'apprentissage et $\mathbf{L} = \{l_1, \dots, l_n\}$ l'ensemble des classes des individus de $\mathbf{X}$ tel que $l_k = -1$ si $h(\mathbf{x}_k) \leq 0$ et $+1$ sinon.

Si les différents exemples de $\mathbf{X}$ sont linéairement séparables, il existe une infinité d'hyper-plans séparateurs $h(\mathbf{x})$. Le modèle de SVM va alors rechercher l'hyper-plan optimal permettant de maximiser la marge sur l'ensemble $(\mathbf{X}, \mathbf{L})$. Tout individu $\mathbf{x} \in \mathbb{R}^m$ pourra alors être classé en déterminant le signe de sa projection sur l'hyper-plan optimal $h(\mathbf{x})$ tel que :

$$h(\mathbf{x}) = \sum_{k=1}^n \alpha_k^* l_k (\mathbf{x}^T \cdot \mathbf{x}_k) + w_0 . \quad (3.29)$$

Si l'on ne se place pas dans le cas de SVM à marge souple, les α_k^* correspondent aux multiplicateurs de Lagrange optimaux, tels que $\alpha_k^* \neq 0$ uniquement si l'individu $\mathbf{x}_k$ est situé sur l'hyper-plan de marge maximale (i.e. la somme n'est réalisée que sur les individus de l'ensemble d'apprentissage correspondant à des vecteurs supports). w_0 est la constante de l'hyper-plan.

Lorsque les données ne sont pas linéairement séparables dans l'espace des caractéristiques, les SVM utilisent une méthode de projection appelée "astuce du noyau" (*Kernel trick*) [Aizerman *et al.*, 1964]. Les individus sont projetés dans un espace de dimension supérieure où une séparation linéaire est plus évidente.

Considérant φ une transformation non-linéaire, la projection d'un individu $\mathbf{x} \in \mathbb{R}^m$ sur l'hyper-plan optimal est déterminée par :

$$h(\mathbf{x}) = \sum_{k=1}^n \alpha_k^* l_k (\varphi(\mathbf{x}_k)^T \cdot \varphi(\mathbf{x})) + w_0 . \quad (3.30)$$

Cette formulation utilise un produit scalaire entre deux vecteurs de grande dimension, nécessitant un temps de calcul significatif. Les SVM utilisent alors une fonction noyau définie par :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T \cdot \varphi(\mathbf{x}_j) . \quad (3.31)$$

L'équation (3.30) devient alors :

$$h(\mathbf{x}) = \sum_{k=1}^n \alpha_k^* l_k K(\mathbf{x}_k, \mathbf{x}) + w_0 . \quad (3.32)$$

Ainsi, tous les calculs sont réalisés dans l'espace d'origine des caractéristiques. La transformation φ n'est jamais appliquée, seule la fonction noyau est prise en compte. Les deux noyaux les plus utilisés sont le polynomial et le gaussien, définis par :

$$\text{Noyau polynomial : } K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \cdot \mathbf{x}_j)^d \quad (3.33)$$

$$\text{Noyau gaussien : } K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) . \quad (3.34)$$

Le modèle de SVM utilisé dans notre approche évolue avec des marges souples et autorise ainsi certains individus de l'ensemble d'apprentissage à être placés à l'intérieur de

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

la marge ou même mal classés. Cela correspond à définir une borne sur la valeur des α_k^* appelée constante de régularisation C telle que $\forall \mathbf{x} \in \mathbf{X}$:

$$\alpha_k^* = 0 \text{ si l'individu } \mathbf{x} \text{ est bien classé} \quad (3.35)$$

$$0 < \alpha_k^* < C \text{ si l'individu } \mathbf{x} \text{ est sur la marge} \quad (3.36)$$

$$\alpha_k^* = C \text{ si l'individu } \mathbf{x} \text{ est à l'intérieur de la marge ou mal classé.} \quad (3.37)$$

Les équations (3.35) à (3.37) sont appelées conditions KKT (Karush-Kuhn-Tucker) [Karush, 1939, Kuhn et Tucker, 1951].

Le noyau choisi est de type gaussien car il nécessite moins de paramètres que le noyau polynomial, ce qui réduit le nombre d'apprentissages nécessaires.

L'algorithme d'apprentissage est réalisé par la méthode SMO (*Sequential Minimal Optimisation*), qui détermine les vecteurs supports, leurs multiplicateurs de Lagrange α_k^* et la constante de l'hyper-plan w_0 (voir [Platt *et al.*, 1999] pour plus de détails). Les sous-ensembles $\mathbf{X}_l$ et $\mathbf{X}_e$ sont utilisés afin de déterminer la constante de régularisation C et le paramètre du noyau σ .

Une fois le modèle de SVM appris, l'équation (3.32) est introduite directement comme fonction vitesse du contour actif afin de déterminer la classe de tout pixel $\mathbf{x}$ en fonction de ses coefficients d'Haralick :

$$F(\mathbf{x}) = \sum_{k=1}^n \alpha_k^* l_k K(\mathbf{x}_k, \mathbf{x}) + w_0 . \quad (3.38)$$

Le signe de $F(\mathbf{x})$ permet alors d'établir si les caractéristiques de texture du pixel $\mathbf{x}$ correspondent à celles de la classe C^{in} ou de la classe C^{out} .

Réseau de neurones artificiels

Le troisième classificateur utilisé est un réseau de neurones artificiels. Introduits dans [Rosenblatt, 1958], les réseaux de neurones sont des modèles inspirés de la biologie utilisant une approche connexionniste afin d'interconnecter des neurones artificiels. L'un des modèles les plus populaires est le perceptron multicouche (*Multi Layer Perceptron*, MLP) développé dans [Rumelhart *et al.*, 1986], qui est connu comme approximateur universel de fonctions. Le MLP est un réseau multicouche à propagation avant composé d'une couche d'entrée, d'une ou plusieurs couches cachées et d'une couche de sortie. Le modèle utilisé pour guider le contour actif est composé de 12 neurones d'entrée (les coefficients d'Haralick), une seule couche cachée comprenant n neurones et 2 neurones de sortie. Les poids synaptiques sont initialisés aléatoirement et l'apprentissage est réalisé par l'algorithme de rétropropagation du gradient de l'erreur [Lippmann, 1987]. Les sous-ensembles $\mathbf{X}_l$ et $\mathbf{X}_e$ sont utilisés pour déterminer la meilleure valeur n correspondant au nombre de neurones cachés.

Une fois les différents poids synaptiques du réseau appris, la valeur du $k^{ème}$ neurone de sortie pour un exemple $\mathbf{x} = \{x_1, x_2, \dots, x_m\}$, $\mathbf{x} \in \mathbb{R}^m$ est donnée par :

$$S_k(\mathbf{x}) = f\left(\sum_{i=1}^n ow_{k,i} \cdot f\left(\sum_{j=1}^{12} hw_{i,j} x_j + hw_{i,13}\right) + ow_{1,n+1}\right) , \quad (3.39)$$

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

avec f la fonction de transfert log-sigmoïde, définie par :

$$f(x) = \frac{1}{1 + e^{(-x)}} . \quad (3.40)$$

La classification d'un exemple sera alors déterminée par le neurone de sortie possédant le plus fort potentiel (correspondant à la valeur de $S_k(\mathbf{x})$ maximum). Afin d'introduire cette règle de classification dans un contour actif, elle est exprimée comme une fonction vitesse telle que :

$$F(\mathbf{x}) = S_1(\mathbf{x}) - S_2(\mathbf{x}) . \quad (3.41)$$

Le neurone de sortie possédant le plus haut potentiel déterminera alors le signe de F et guidera ainsi l'évolution de la courbe. La figure (3.9) décrit le perceptron multicouche utilisé.

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

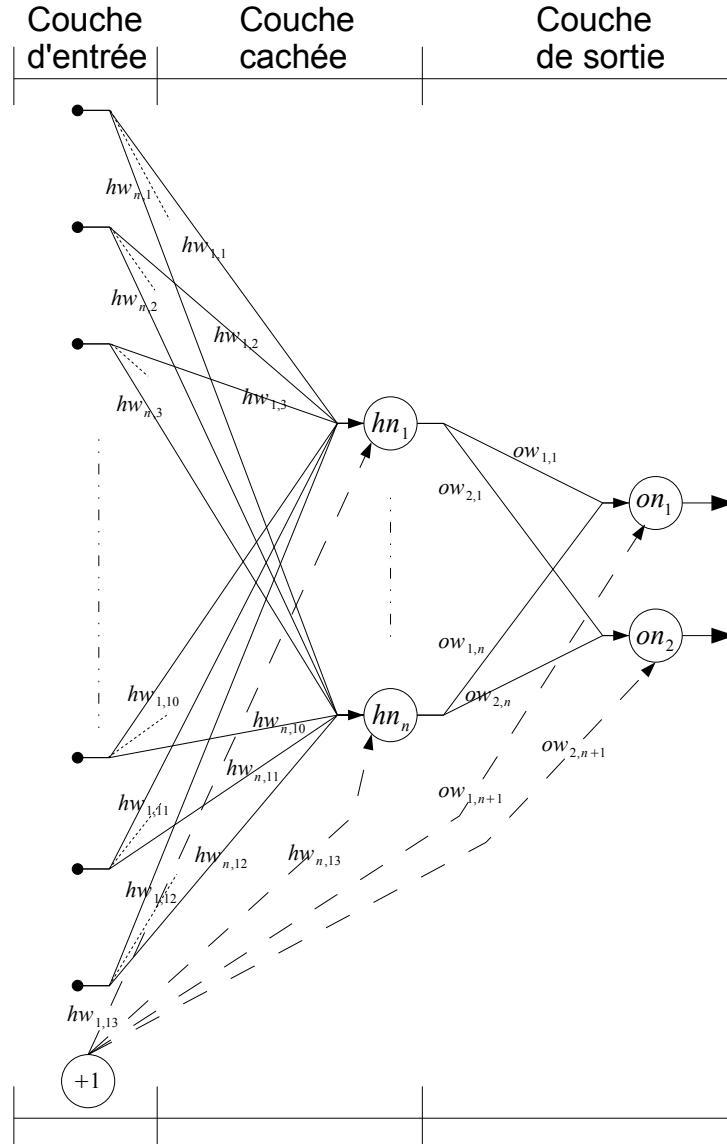


FIG. 3.9 – Modèle de perceptron multicouche utilisé dans notre approche. La première couche est composée de 13 neurones (12 neurones propageant chacun sans le modifier un coefficient d'Haralick de l'individu traité et un neurone de biais). La deuxième couche comporte n neurones cachés (hn_i). La couche de sortie se compose de 2 neurones (on_i). Le neurone de sortie possédant le plus fort potentiel détermine alors la classe d'appartenance de l'exemple. $hw_{i,j}$ représente le poids synaptique reliant le $j^{ème}$ neurone de la couche d'entrée au $i^{ème}$ neurone de la couche cachée et $ow_{i,j}$ celui reliant le $j^{ème}$ neurone de la couche cachée au $i^{ème}$ neurone de la couche de sortie.

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

3.2.2.4 Processus complet de segmentation

La segmentation d'images texturées par contour actif guidé par classificateur supervisé comporte deux principales étapes.

Dans un premier temps, toutes les informations extraites de l'image d'apprentissage sont traitées par une étape d'apprentissage. Son but est alors de créer l'ensemble d'apprentissage $\mathbf{X}$ et de le diviser (en conservant les mêmes proportions d'individus pour chacune des deux classes) afin de créer les deux sous-ensembles $\mathbf{X}_l$ et $\mathbf{X}_e$. L'algorithme d'apprentissage du classificateur est alors exécuté plusieurs fois sur $\mathbf{X}_l$ avec différents jeux de paramètres, et la configuration ayant permis au classificateur d'obtenir les meilleurs résultats sur $\mathbf{X}_e$ est alors conservée.

Dans un deuxième temps, l'étape de segmentation est réalisée. Le contour actif est alors initialisé manuellement dans l'image à traiter, la faible dépendance du modèle à l'initialisation n'entraînant qu'une seule contrainte : qu'au moins un pixel de la région formée par l'intérieur de la courbe appartienne à l'objet recherché.

Le classificateur précédemment appris est alors utilisé pour déformer la courbe, permettant ainsi au modèle de détecter les frontières de l'objet. Le même processus est répété pour chaque image à traiter, sans changer de classificateur. Le processus complet est décrit figure (3.10).

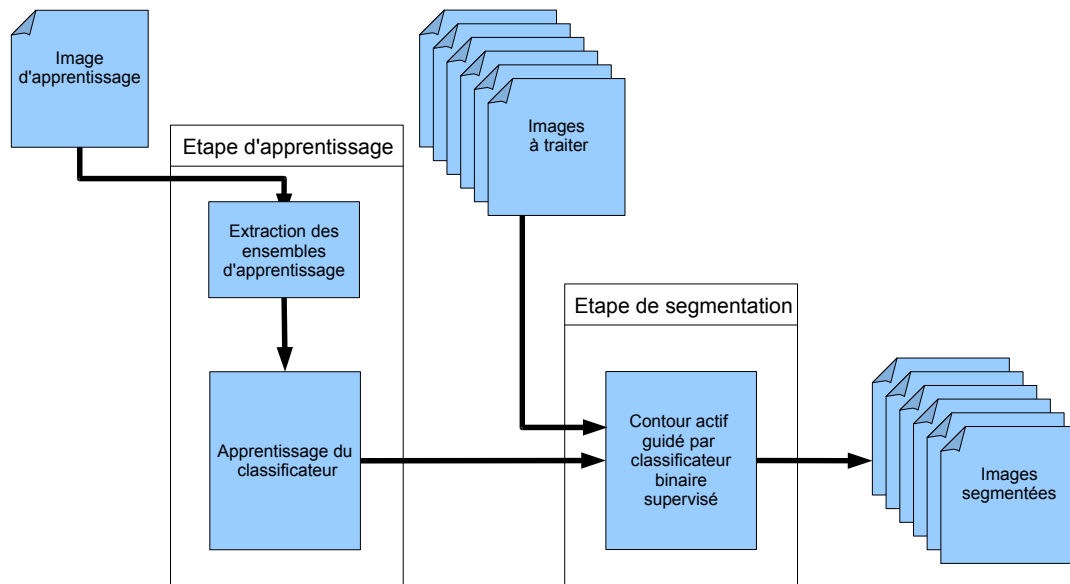


FIG. 3.10 – Processus complet de segmentation : dans un premier temps les ensembles $\mathbf{X}_l$ et $\mathbf{X}_e$ extraits de l'image d'apprentissage sont utilisés afin de réaliser l'algorithme d'apprentissage du classificateur. Dans un deuxième temps, celui-ci est utilisé pour guider le contour actif sur les autres images de la série.

Le contour actif guidé par classificateur a été testé dans le cadre de différentes problé-

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

matiques de segmentation. Tous les résultats sont détaillés au 4.5.

Une application du contour actif guidé par classificateur supervisé à la segmentation d'images échographiques 3D est également présentée au 4.6.

3.2. CONTOURS ACTIFS SUPERVISÉS POUR LA SEGMENTATION D'IMAGES TEXTURÉES PAR ENSEMBLES DE NIVEAUX

Chapitre 4

Application des modèles proposés à la segmentation d'images 2D, 3D et texturées

4.1 Evaluation de la qualité de segmentation

Alors que la rapidité d'un algorithme est aisément quantifiable, soit par sa complexité, soit directement par son temps d'exécution, l'évaluation de la qualité d'une segmentation demeure difficile.

En effet, si l'on se place dans le cadre d'une évaluation visuelle, deux personnes n'auront pas la même perception d'un résultat de segmentation. Lorsque la précision de la méthode ne constitue pas une contrainte forte, une évaluation visuelle peut être envisagée car les faibles variations du résultat de segmentation ne présentent pas une grande importance. En revanche, lorsque la précision est le but du modèle, une évaluation rigoureuse du résultat de segmentation est impérative.

Les critères d'évaluation de la qualité des segmentations peuvent être regroupés en deux familles : les critères non-supervisés et les critères supervisés. Dans le cas de critères non-supervisés, aucune information sur la forme finale idéale de la segmentation n'est disponible. L'évaluation du résultat est généralement réalisée en utilisant des critères statistiques tels que l'homogénéité des régions créées, la courbure régulière des frontières ou encore le contraste entre les régions. Dans le cas de critères supervisés, une segmentation idéale, appelée vérité terrain, est disponible. Celle-ci correspond généralement à une segmentation manuelle réalisée par un expert du domaine d'application de l'algorithme de segmentation (dans le cadre d'application d'imagerie médicale par exemple, la vérité terrain sera définie par un médecin). Les résultats de segmentation sont comparés à cette vérité terrain et une évaluation de la différence entre les deux segmentations est alors déterminée. Cette mesure peut être perçue comme une distance entre le résultat de segmentation et la vérité terrain. Idéalement, afin de garantir l'indépendance de l'évaluation par rapport à l'expert humain, plusieurs vérités terrains sont généralement réalisées par des experts différents sur la même image, et le critère d'évaluation final correspond alors à la moyenne des mesures obtenues

en comparant le résultat de segmentation à chaque vérité terrain.

Nos méthodes d'accélération développées pour les modèles paramétriques ont été appliquées à la segmentation d'images 2D et 3D. Le but premier de l'implémentation 2D étant de valider les approches proposées, seule la rapidité d'exécution a été utilisée comme critère de comparaison lors de cette étape de tests, la précision de segmentation n'étant alors pas évaluée par une métrique mais uniquement de manière visuelle. Lors de l'adaptation des méthodes développées à la segmentation d'images 3D, une évaluation de la précision de segmentation a été menée car il était nécessaire de prouver que l'accélération de l'algorithme n'avait pas pour conséquence une baisse de la qualité de la segmentation. Le critère d'évaluation utilisé est basé sur la distance de Hausdorff H [Huttenlocher *et al.*, 1993, Rucklidge, 1996] définie par :

$$H(I_s, I_t) = \max(h(I_s, I_t), h(I_t, I_s)) . \quad (4.1)$$

I_t représente les frontières de la vérité terrain et I_s le résultat de segmentation évalué. Si l'on définit $d_i(I_s, I_t)$ comme la distance Euclidienne séparant le point i appartenant à I_s du point le plus proche de I_t , alors la distance de Hausdorff orientée $h(I_s, I_t)$ est définie comme : $h(I_s, I_t) = \max_{i \in I_s} d_i(I_s, I_t)$. Ainsi, la distance $H(I_s, I_t)$ peut être considérée comme une évaluation de l'écart maximum entre deux segmentations.

Cette mesure donne une évaluation "dans le pire des cas" de la segmentation, car un unique point de frontière isolé entraîne une distance de Hausdorff très importante. La distance de Hausdorff modifiée H_{mean} introduite dans [Dubuisson et Jain, 1994] a donc été préférée. Celle-ci est définie par :

$$H_{\text{mean}}(I_s, I_t) = \max(h_{\text{mean}}(I_s, I_t), h_{\text{mean}}(I_t, I_s)) . \quad (4.2)$$

La distance de Hausdorff modifiée orientée $h_{\text{mean}}(I_s, I_t)$ est définie par :

$$h_{\text{mean}}(I_s, I_t) = \frac{1}{\text{card}(I_s)} \sum_{i \in I_s} d_i(I_s, I_t) . \quad (4.3)$$

La distance de Hausdorff modifiée peut être considérée comme une évaluation de l'écart moyen entre deux segmentations.

Le comportement des modèles 3D est similaire à celui des modèles 2D et la surface active se déforme par le biais de points de contrôle. La distance de Hausdorff nécessitant des frontières échantillonnées selon la grille de voxels de l'image, la surface active est ré-échantillonnée en utilisant un algorithme de type Bresenham 3D [Dimitrov et Sramek, 2004].

Le but des modèles implicites proposés au 3.2 est d'obtenir une segmentation très précise d'images texturées. L'évaluation de la qualité de segmentation représente alors la mesure de comparaison la plus importante pour juger les performances des approches développées.

La complexité des images appréhendées, additionnée à la capacité des modèles implicites à gérer les changements de topologie, entraînent des résultats de segmentation plus complexes qu'avec les modèles paramétriques, les frontières étant alors très nombreuses. La distance de Hausdorff possède une grande complexité algorithmique et son utilisation pour comparer des frontières complexes n'est pas recommandée. Pour ces différentes raisons, le

choix d'une nouvelle métrique nous est apparu nécessaire pour mesurer les performances des modèles développés.

Une comparaison des principaux critères d'évaluation a été proposée dans [Chabrier *et al.*, 2004] et le critère de Pratt [Pratt *et al.*, 1978] (appelée couramment *Figure of Merit*) est identifié comme le plus efficace. Celui-ci fournit une évaluation rapide de la qualité des frontières de segmentation et correspond à la distance empirique entre la vérité terrain composée de l'ensemble de frontières I_t et la segmentation obtenue, composée de l'ensemble de frontières I_s :

$$PRA(I_s, I_t) = \frac{1}{\max(\text{card}(I_t), \text{card}(I_s))} \sum_{i=1}^{\text{card}(I_s)} \frac{1}{1 + d_i(I_s, I_t)^2}, \quad (4.4)$$

avec $d(I_s, I_t)$ la distance entre le $i^{\text{ème}}$ pixel de I_s et le pixel le plus proche appartenant à I_t . Une valeur $PRA(I_s, I_t) = 1$ correspond à une segmentation parfaite ($I_s = I_t$), l'estimateur diminuant à mesure que la segmentation se détériore.

L'évaluation de la qualité des résultats de segmentation par un seul critère ne permet pas de garantir objectivement les performances. En effet, quels que soient les résultats obtenus, il est juste de se demander si le critère utilisé n'avantage pas certaines approches au détriment des autres, ce qui aurait pour conséquence d'altérer l'authenticité de la comparaison réalisée. Afin de garantir l'indépendance des approches comparées par rapport au critère d'évaluation, une métrique basée sur la comparaison des régions créées par la segmentation a également été retenue : la distance générique d'anomalies [Cardoso et Corte-Real, 2005]. Cette mesure est basée sur la distance de partition définie dans [Gusfield, 2002] par :

Etant donné deux partitions P et Q d'une image Ω , la distance de partition d_{sym} représente le nombre minimum d'éléments devant être supprimés de Ω de manière à ce que P et Q (restreintes aux éléments restants) soient identiques.

En d'autres termes, dans le cas d'une segmentation en deux régions comme celle proposée ici (une région correspondant au fond et l'autre à l'objet recherché), d_{sym} représente le nombre de pixels appartenant à une région différente dans la vérité terrain et dans la segmentation réalisée.

La mesure générique d'anomalies d_{mga} utilisée correspond à la distance de partition normalisée entre les régions créées par la vérité terrain I_t et celles créées par la segmentation réalisée I_s : $d_{mga} = d_{sym}(I_t, I_s)/(N - 1)$, avec N le nombre de pixels de l'image. Une segmentation idéale aura une valeur $d_{mga} = 0$, alors qu'une segmentation inverse à la vérité terrain (i.e. tous les pixels sont étiquetés comme appartenant à la mauvaise région) aura une valeur $d_{mga} = 1$. En pratique, une valeur de 0,5 correspond déjà à une très mauvaise segmentation, car un pixel sur deux étant mal affecté, cela est équivalent à une segmentation aléatoire.

Afin de pouvoir comparer les deux métriques utilisées, une version modifiée du critère de Pratt $PRA_m(I_s, I_t) = 1 - PRA(I_s, I_t)$ est utilisée. Ainsi, quel que soit le critère utilisé, une valeur de 0 correspondra à une segmentation idéale. La mesure augmentera à mesure que la segmentation se dégradera jusqu'à une limite de 1 pour le critère de Pratt modifié environ 0,5 pour la mesure générique d'anomalies.

4.2 Segmentation d'images 2D par modèle paramétrique à anticipation par calcul de voisinages dynamiques

Par soucis de simplicité, les acronymes suivants seront utilisés dans les différents résultats présentés :

- EG : évolution gloutonne classique, sans voisinage dynamique.
- EVAL : évolution par voisinage à anticipation linéaire.
- EVADec : évolution par voisinage à anticipation par décalage.
- EVADef : évolution par voisinage à anticipation par déformation.

Ces quatre méthodes ont été comparées sur plusieurs images synthétiques et naturelles afin d'évaluer l'accélération produite par l'introduction d'un voisinage dynamique dans l'évolution gloutonne. Les images ont été choisies de manière à tester les principales propriétés des contours actifs.

La première image, illustrée en figure (4.1.a), représente un objet naturel (coquillage). Son intérêt est de posséder un arrière plan très bruité pouvant devenir un obstacle tant à la précision du contour qu'à sa rapidité de convergence.

Les figures (4.1.b) et (4.1.c) illustrent la deuxième et la troisième image, réalisées aux rayons X. La différence de contraste entre l'objet recherché et l'arrière plan y est très faible et les frontières sont peu marquées (i.e. les intensités de l'objet recherché et de l'arrière plan sont très proches).

La figure (4.1.d) décrit la dernière image utilisée. Cette image artificielle a été choisie en raison des quatre angles saillants la composant, permettant de tester le comportement du contour actif dans les zones de forte courbure.

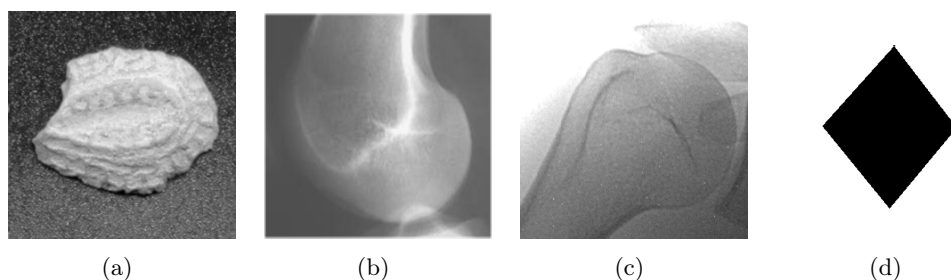


FIG. 4.1 – Images utilisées pour comparer les performances des méthodes proposées : a) Coquillage cassé b) Image aux rayons X du condyle de la jonction tibio-fémorale c) Image aux rayons X de la tête humérale d'une épaule humaine d) Image synthétique de losange.

Le tableau (4.1) compare la rapidité de segmentation des quatre méthodes et la figure (4.2) illustre les segmentations obtenues sur chaque image par la méthode la plus rapide. De manière générale, l'EVAL et l'EVADec améliorent systématiquement le temps d'exécution, avec un coefficient d'accélération moyen de 1,51 pour l'EVAL (i.e. la méthode est 1,51 fois plus rapide que l'évolution gloutonne) et 1,75 pour l'EVADec. Cette dernière fournit les meilleures performances sur les trois images naturelles (figures (4.2.a) à (4.2.c)) alors que

4.2. SEGMENTATION D'IMAGES 2D PAR MODÈLE PARAMÉTRIQUE À ANTICIPATION PAR CALCUL DE VOISINAGES DYNAMIQUES

l'EVAL est la plus rapide sur l'image synthétique (figure (4.2.d)). Cela s'explique par la forte régularité de l'image du losange. En effet, l'EVAL est généralement la plus rapide à atteindre la frontière de l'objet mais le caractère trop exclusif de sa recherche la contraint à réaliser de nouvelles itérations afin de déplacer les points de contrôle le long de la frontière (ce phénomène est décrit au 3.1.2). L'image du losange étant très régulière, les points de contrôle ne nécessitent que très peu de modifications une fois la frontière atteinte. Ces conditions sont donc favorables à l'EVAL.

Bien qu'intéressante, l'accélération apportée par l'EVADef n'est pas aussi appuyée qu'avec les deux autres approches proposées. La rapidité de cette méthode est même inférieure à celle de l'évolution gloutonne classique sur la troisième image. La cause de cette chute de performances est le grand nombre d'itérations réalisées sur cette image qui entraîne une grande déformation du voisinage. Celui-ci devient alors très large, ralentissant ainsi l'évolution du contour actif (même si en pratique la déformation est implémentée de manière à être bornée).

Image	Méthode	Nb. itérations	Temps (s)
Coquillage	EG	38	0,73
	EVAL	26	0,50
	EVADec	18	0,34
	EVADef	26	0,63
Condyle	EG	49	1,76
	EVAL	36	0,67
	EVADec	31	0,59
	EVADef	24	0,80
Epaule	EG	65	1,23
	EVAL	51	1,00
	EVADec	48	0,90
	EVADef	43	2,11
Losange	EG	42	0,77
	EVAL	18	0,33
	EVADec	20	0,34
	EVADef	16	0,52

TAB. 4.1 – Evaluation de la vitesse de segmentation des méthodes proposées et de l'évolution gloutonne classique sur les quatre images 2D testées.

Un point important concerne le réglage des poids affectés aux différentes énergies du modèle d'évolution gloutonne. En effet, ces énergies sont très importantes car elles permettent de contrôler le comportement du contour actif. Ainsi, les énergies de courbure et de continuité vont avoir tendance à retenir les points proches de la courbe. Ce comportement est à l'opposé de la philosophie des approches proposées qui, à chaque itération, cherchent à envoyer chaque point du contour actif le plus loin possible. Dans ces conditions, il est recommandé de réduire légèrement l'influence de ces deux énergies en diminuant leurs poids dans la fonctionnelle d'énergie (1.17). De cette manière les approches par voisinages dynamiques pourront être pleinement efficaces. Dans le cadre des expérimentations

4.3. SEGMENTATION D'IMAGES 3D PAR MODÈLE PARAMÉTRIQUE À ANTICIPATION PAR CALCUL DE VOISINAGES DYNAMIQUES

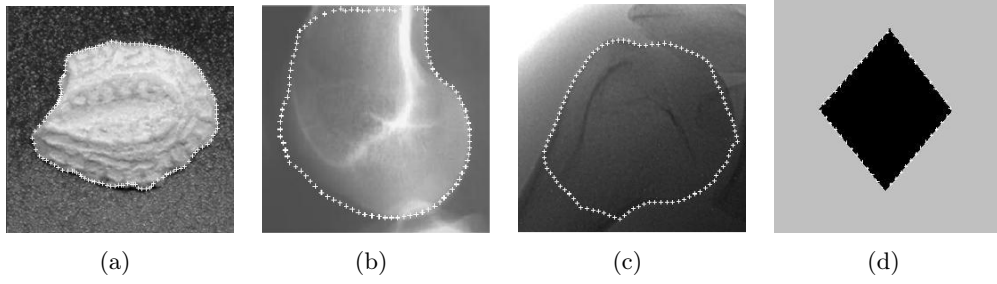


FIG. 4.2 – Illustration des résultats de segmentation obtenus par la méthode la plus rapide sur chaque image. a-c) EVADec. d) EVAL.

présentées ici, les différents poids ont été réglés par essais-erreurs.

Ces résultats encourageants nous ont conduits à implémenter les méthodes EVAL et EVADec dans une problématique de segmentation 3D. La section suivante présente les tests réalisés.

4.3 Segmentation d'images 3D par modèle paramétrique à anticipation par calcul de voisinages dynamiques

Alors qu'un contour actif représenté de manière paramétrique est constitué d'une succession de points de contrôle reliés entre eux par une notion de voisinage deux à deux (voir au 1.1.2.1 pour plus de détails), son équivalent 3D (appelé surface active) correspond à un ensemble de sommets reliés entre eux par des arêtes, chaque sommet possédant alors quatre voisins. Une telle surface correspond à un maillage 3D qui va se déformer en déplaçant ses sommets de manière équivalente à un contour actif se déformant grâce au déplacement de ses points de contrôle.

La méthode de déformation utilisée est une adaptation 3D de l'algorithme d'évolution gloutonne proposée dans [Bulpitt et Efford, 1996]. Le voisinage utilisé autour de chaque point de contrôle n'est plus une fenêtre carrée mais cubique, comme l'illustre la figure (4.3).

Afin de permettre au maillage de conserver une grande précision lorsqu'il doit grandir de manière significative, une étape de remaillage de la surface est réalisée après chaque itération. Le but de cette étape est d'ajouter ou de supprimer des sommets afin de garantir l'homogénéité des distances entre deux sommets adjacents (i.e. les arêtes restent de même longueur). Ainsi, chaque couple de sommets adjacents doit satisfaire la contrainte :

$$d_{min} \leq \|\mathbf{v}_i - \mathbf{v}_j\| \leq d_{max} . \quad (4.5)$$

d_{min} et d_{max} sont fixés tels que $d_{max} \leq 2d_{min}$, et choisis proches de la taille de la grille de voisinage w afin que le ré-échantillonnage n'entre pas en conflit avec l'espace de recherche (correspondant aux mouvements possibles) des sommets existants. Lorsque $\|\mathbf{v}_i - \mathbf{v}_j\| \geq d_{max}$, un nouveau sommet est ajouté au milieu du segment $\mathbf{v}_i\mathbf{v}_j$ et connecté aux voisins

4.3. SEGMENTATION D'IMAGES 3D PAR MODÈLE PARAMÉTRIQUE À ANTICIPATION PAR CALCUL DE VOISINAGES DYNAMIQUES

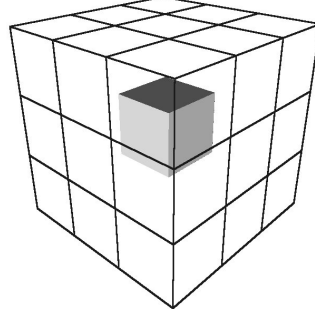


FIG. 4.3 – Voisinage cubique utilisé autour de chaque point de contrôle par l'évolution gloutonne 3D.

commun de $\mathbf{v}_i$ et $\mathbf{v}_j$ (voir figure (4.4.b)). Si $\|\mathbf{v}_i - \mathbf{v}_j\| \leq d_{min}$, le sommet $\mathbf{v}_j$ est supprimé et $\mathbf{v}_i$ est alors déplacé vers le milieu du segment $\mathbf{v}_i\mathbf{v}_j$ (voir figure (4.4.c)).

Afin d'évaluer les performances des approches par EVAL et EVA-Dec dans une problématique de segmentation 3D, plusieurs tests ont été réalisés sur des images synthétiques et naturelles. Le même protocole d'initialisation a été utilisé pour les quatre méthodes, et leurs performances sont comparées en termes de rapidité et de précision.

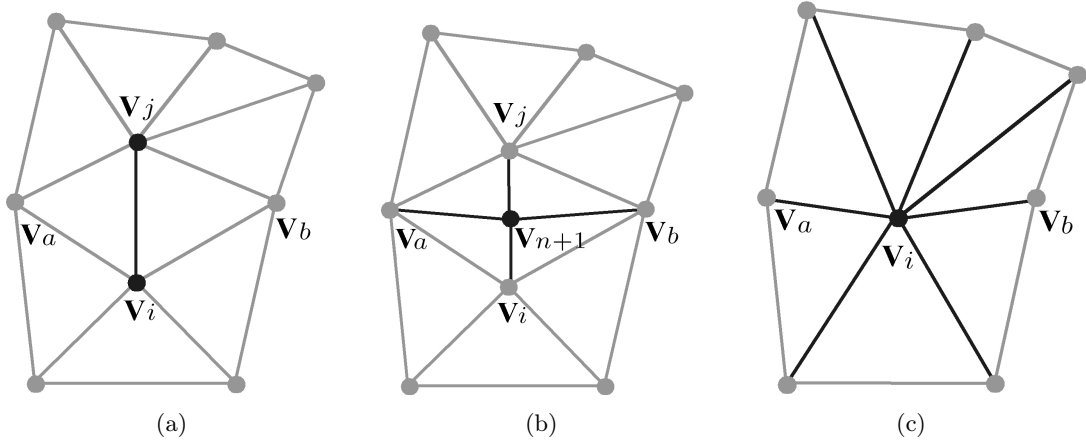


FIG. 4.4 – Principe du remaillage : a) Maillage initial. b) Ajout du sommet $\mathbf{v}_{n+1}$. c) Suppression du sommet $\mathbf{v}_j$.

4.3.1 Segmentation d'images synthétiques

Les images synthétiques, illustrées en figure (4.5), sont obtenues par introduction d'une forme géométrique régulière dans un arrière plan uniforme. L'image est ensuite bruitée de manière à se rapprocher des conditions rencontrées dans des images naturelles.

La première image (figure (4.5.a)) représente une spirale, choisie de manière à évaluer

4.3. SEGMENTATION D'IMAGES 3D PAR MODÈLE PARAMÉTRIQUE À ANTICIPATION PAR CALCUL DE VOISINAGES DYNAMIQUES

l'aptitude de la surface active à procéder à un ré-échantillonnage des points de contrôle en ajoutant des sommets au maillage. La surface active est initialisée avec 12 sommets à l'intérieur de la forme et le maillage final possède un millier de sommets.

La deuxième image présentée en figure (4.5.b) est un modèle de vase 3D utilisé afin de tester la capacité de la surface active à explorer des concavités (le remaillage est donc désactivé). Le maillage est initialisé avec 2500 sommets.

La figure (4.5.c) illustre la troisième image synthétique utilisée représentant trois ellipsoïdes concentriques. L'intérêt de cette dernière image est de fournir un modèle 3D possédant des angles très différents, aussi bien aigus que très obtus. Le remaillage est également désactivé pour cet exemple et la surface est initialisée avec 2500 sommets.

Afin de vérifier que la vitesse des modèles paramétriques constitue toujours un avantage non négligeable par rapport aux modèles implicites, les méthodes développées sont également comparées à la récente implémentation de Shi et Karl [Shi et Karl, 2005a, Shi et Karl, 2005b] (voir au 1.1.2.2 pour plus de détails), permettant d'accélérer les approches par ensembles de niveaux. La fonction d'évolution du contour actif utilisée est de type géométrique, basée sur les gradients d'intensité (similaire à l'équation (1.84) présentée au 1.2.3). Ce modèle a été choisi en raison de sa rapidité.

En plus du temps d'exécution et du nombre d'itérations réalisées, la précision de chaque méthode est évaluée en utilisant la distance de Hausdorff modifiée H_{mean} décrite au 4.1, exprimée en voxels.

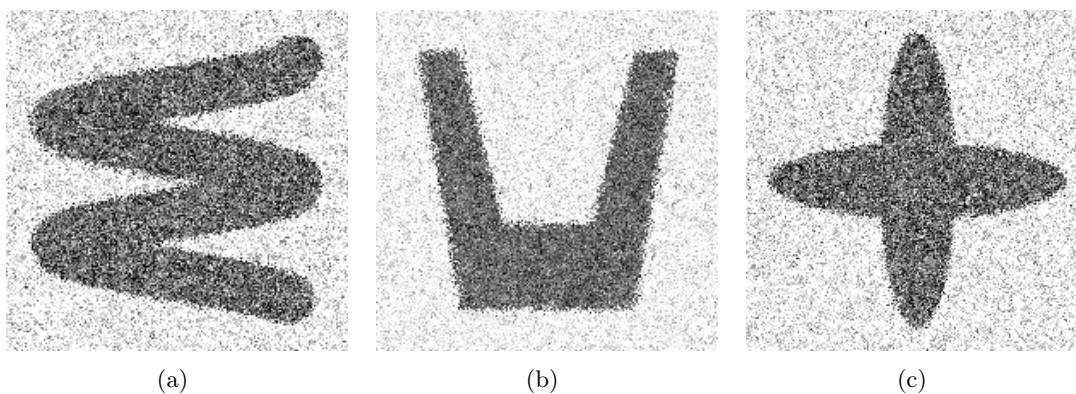


FIG. 4.5 – Représentation d'une coupe de chaque image synthétique utilisée : a) Spirale. b) Vase. c) Ellipsoïdes.

4.3. SEGMENTATION D'IMAGES 3D PAR MODÈLE PARAMÉTRIQUE À ANTICIPATION PAR CALCUL DE VOISINAGES DYNAMIQUES

La largeur initiale de la fenêtre de voisinage L_N jouant un rôle important dans l'algorithme d'évolution gloutonne, plusieurs valeurs sont testées pour chaque méthode. La valeur $L_N = 3$ n'est pas testée avec l'EVA-Dec car elle en annule les effets (la fenêtre ne pouvant se décaler).

Le tableau (4.2) présente les résultats obtenus sur les trois images, et la figure (4.6) les segmentations déterminées par l'approche par voisinage à anticipation linéaire.

Image	Vois.	Méthode	Nb. itérations	Tps. exéc. (s)	H_{mean}
Spirale ($400 \times 400 \times 400$ voxels)	$L_N = 3$	EG	400	0,30	0,497
		LS	65	0,16	0,499
	$L_N = 5$	EG	195	0,52	0,723
		EVAL	63	0,30	0,734
		EVA-Dec	138	0,31	0,728
	$L_N = 7$	EG	145	0,63	0,985
		EVAL	60	0,57	0,997
		EVA-Dec	97	1,11	0,991
	Ensembles de niveaux		1060	441,7	0,995
Ellipsoïdes ($200 \times 200 \times 200$ voxels)	$L_N = 3$	EG	47	0,63	0,534
		EVAL	19	0,41	0,545
	$L_N = 5$	EG	25	1,05	0,716
		EVAL	12	0,69	0,734
		EVA-Dec	16	0,98	0,721
	$L_N = 7$	EG	18	2,24	0,961
		EVAL	12	1,86	0,985
		EVA-Dec	14	2,07	0,972
	Ensembles de niveaux		120	308,2	1,030
Vase ($200 \times 200 \times 200$ voxels)	$L_N = 3$	EG	155	0,98	0,665
		EVAL	68	0,72	0,687
	$L_N = 5$	EG	120	1,27	0,875
		EVAL	50	0,85	0,896
		EVA-Dec	92	1,08	0,892
	$L_N = 7$	EG	109	1,98	1,337
		EVAL	43	1,13	1,554
		EVA-Dec	60	2,20	1,423
	Ensembles de niveaux		210	470,4	0,815

TAB. 4.2 – Comparaison des temps d'exécution et de la précision de segmentation des quatre méthodes testées sur les trois images synthétiques 3D.

Une première conclusion est que les deux méthodes proposées améliorent sensiblement la rapidité de l'algorithme d'évolution gloutonne. L'EVAL améliore systématiquement la rapidité de l'évolution gloutonne, quelle que soit la taille du voisinage, avec un coefficient d'accélération moyen de 1,47. L'accélération apportée par l'EVA-Dec est manifeste pour $L_N = 5$, avec un coefficient moyen de 1,26. La méthode éprouve par contre plus de difficultés lorsque $L_N = 7$ car les calculs entraînés par la création du nouveau voisinage décalé

4.3. SEGMENTATION D'IMAGES 3D PAR MODÈLE PARAMÉTRIQUE À ANTICIPATION PAR CALCUL DE VOISINAGES DYNAMIQUES

deviennent lourds, pénalisant la méthode. Contrairement aux résultats obtenus pour la segmentation 2D, l'EVAL est plus rapide que l'EVADec. Une explication est que le passage en 3D ne modifie pas la complexité de la première car les calculs sont toujours réalisés sur une ligne. En revanche les calculs entraînés par la deuxième sont désormais réalisés sur un cube, et non plus sur un carré, ce qui accroît considérablement la complexité.

La vitesse de segmentation du modèle implémenté en ensembles de niveaux s'avère plus faible que celle des modèles paramétriques, malgré l'utilisation d'une technique d'accélération récente. La supériorité des modèles paramétriques en termes de rapidité reste donc d'actualité.

Il est intéressant de noter que de manière générale, l'augmentation de la largeur de la grille de voisinage, bien que permettant de diminuer sensiblement le nombre d'itérations, ralentit la segmentation quelle que soit la méthode. Ce résultat permet de confirmer que l'EVADef n'aurait pas été à son avantage sur ce type d'application 3D.

Concernant la précision de segmentation, les approches proposées ne dégradent que très peu le maillage obtenu par évolution gloutonne, d'environ 3,4 % pour l'EVAL et 1,9 % pour l'EVADec. L'approche par ensembles de niveaux reste dans les mêmes ordres de grandeur car la fonction d'évolution utilisée a volontairement été choisie simple de manière à favoriser la rapidité.

La taille de la grille de voisinage L_N joue un rôle important dans la qualité du résultat, car c'est elle qui va décider des seuils de remaillage d_{min} et d_{max} . Ainsi, augmenter L_N réduit les possibilités de remaillage et entraîne un modèle possédant un nombre réduit de sommets, désavantageant ainsi la précision de segmentation.

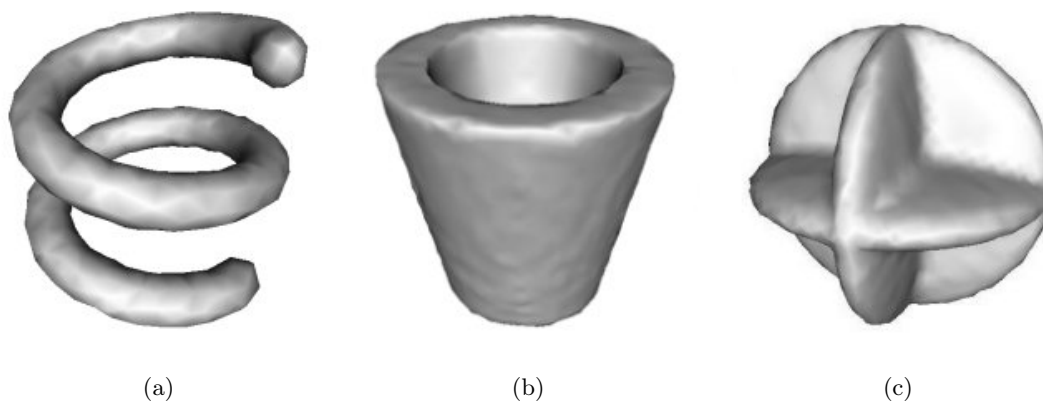


FIG. 4.6 – Segmentations obtenues par l'EVAL sur les trois images synthétiques.

4.3.2 Segmentation d'images naturelles

Les approches proposées ont également été appliquées à la segmentation de l'aorte dans des images de scanner de type tomодensitométrique, composées de plusieurs coupes 2D. Le but de cette segmentation est de permettre aux médecins de détecter et mesurer les premiers signes de l'anévrisme aortique abdominal, correspondant à une dilatation de

4.3. SEGMENTATION D'IMAGES 3D PAR MODÈLE PARAMÉTRIQUE À ANTICIPATION PAR CALCUL DE VOISINAGES DYNAMIQUES

la paroi de l'aorte. Ce type d'images présente en outre l'intérêt d'être composé de gros volumes de données, entraînant des temps de segmentation importants. L'accélération de la segmentation représente donc ici un enjeu crucial.

La surface active est initialisée dans l'image tomодensitométrique comme une sphère à l'intérieur de l'aorte, puis gonflée en utilisant l'énergie ballon couplée à la technique du remaillage.

Le tableau (4.3) compare les approches proposées à l'algorithme d'évolution gloutonne et à l'implémentation implicite de Shi et Karl en termes de rapidité et de précision (le modèle utilisé est le même qu'au 4.3.1). La figure (4.7) illustre le résultat de segmentation obtenu par l'algorithme d'évolution gloutonne sur ce type d'image.

Les premières tendances extraites des résultats de segmentation d'images synthétiques présentées dans la section précédente sont confirmées. L'EVAL reste la plus rapide, avec un coefficient moyen d'accélération de l'évolution gloutonne de 1,4. En revanche, l'EVA-Dec accélère systématiquement l'évolution gloutonne, quelle que soit la taille du voisinage, avec un coefficient moyen de 1,21. L'approche par ensembles de niveaux reste très lente comparée aux approches paramétriques.

Les différentes approches testées sont globalement moins précises que lors des tests sur images synthétiques car les images tomодensitométriques constituent des données plus complexes. En outre, le caractère local des approches utilisées (énergies basées contour et non région) ne permet pas une totale exploration de la structure de l'artère. Néanmoins, les précisions des différents modèles paramétriques restent proches, l'EVAL dégradant légèrement le résultat de l'évolution gloutonne de 2,3 % en moyenne, et l'EVA-Dec de seulement 0,8 %. L'approche par ensembles de niveaux permet d'obtenir des segmentations plus précises, avec une évaluation située dans les mêmes ordres de grandeur que les meilleurs résultats des modèles paramétriques. La capacité des modèles implicites à se propager dans des espaces étroits explique ces résultats.

Image	Vois.	Méthode	Nb. itérations	Tps. exéc. (s)	H _{mean}
Image tomодensitométrique (512 × 512 × 810 voxels)	$L_N = 3$	EG	380	17,57	2,41
		EVAL	152	12,30	2,46
	$L_N = 5$	EG	243	26,88	3,24
		EVAL	121	18,82	3,32
		EVA-Dec	155	22,04	3,26
	$L_N = 7$	EG	197	35,49	4,35
		EVAL	137	24,84	4,46
		EVA-Dec	147	29,10	4,40
	Ensembles de niveaux		1060	202,41	2,98

TAB. 4.3 – Comparaison des temps d'exécution et de la précision de segmentation des quatre méthodes testées sur une image tomодensitométrique.

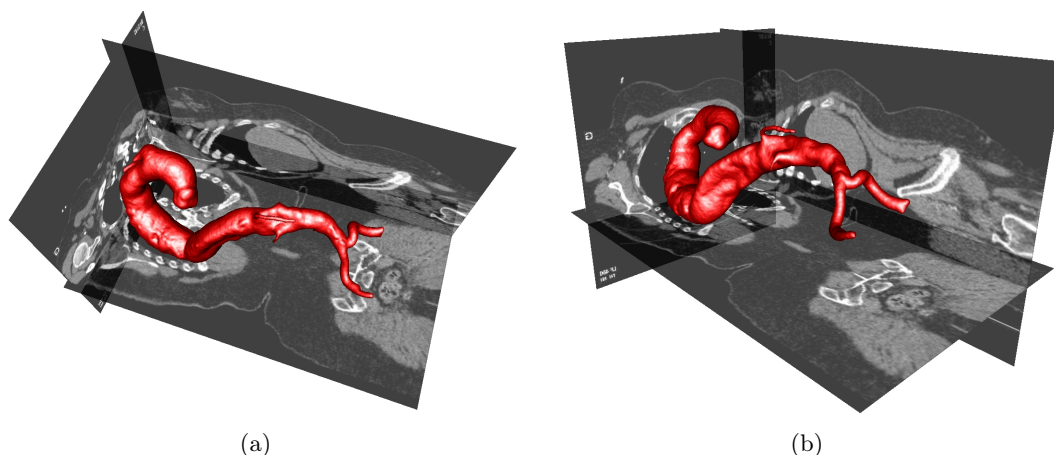


FIG. 4.7 – Segmentation de l’aorte obtenue par l’évolution gloutonne sur une image tomodensitométrique.

4.4 Choix de la méthode

Dans le cadre d’une problématique de segmentation 2D, l’EVADEC s’est avérée la plus efficace, suivie de près par l’EVAL. En ce qui concerne la segmentation 3D, l’EVAL est apparue comme la plus efficace, tant sur les images naturelles que dans une problématique de segmentation d’images médicales. L’EVADEC s’est révélée légèrement en retrait sur les images naturelles, mais très efficace sur les images médicales.

En conséquence, l’EVADEC est recommandée pour les applications 2D, alors que l’EVAL sera plutôt préconisée dans une problématique 3D. Il est également conseillé de restreindre la largeur de la grille de voisinage afin d’assurer une convergence rapide et précise. De manière générale, une largeur $w = 5$ sera la plus efficace avec l’EVADEC, alors qu’une largeur de 3 sera préférée pour l’EVAL. Concernant le réglage des poids de la fonctionnelle d’énergie, il est recommandé de réduire légèrement l’influence des énergies de continuité et de courbure afin de permettre aux méthodes d’être efficaces.

Différentes pistes de recherche sont envisagées pour les modèles paramétriques à anticipation par calcul de voisinages dynamiques. Celles-ci sont présentées dans la conclusion de ce mémoire de thèse.

4.5 Segmentation d'images texturées 2D par contour actif supervisé

Les modèles de contours actifs implicites présentés au 3.2 ont été appliqués avec succès à différentes problématiques de segmentation d'images texturées 2D et 3D. Cette deuxième partie du chapitre présente les tests réalisés. Afin de rendre les modèles indépendant du type de texture traité, ceux-ci sont intégrés dans un processus de segmentation supervisé. Trois problématiques différentes sont traitées : la segmentation d'images synthétiques, d'images échographiques, et enfin d'images satellites. Les différents modèles proposés sont comparés au modèle de contour actif implicite basé région de Chan et Vese [Chan et Vese, 2001, Chan *et al.*, 2000] (détaillé au 1.2.2.3). L'évaluation de la précision de chaque méthode est réalisée par une version modifiée du critère de Pratt ainsi que par la mesure générique d'anomalies (voir au 4.1). Les approches utilisant des classificateurs binaires ont été appliquées à tous les types d'images, et celle basée sur la programmation linéaire uniquement aux images échographiques.

Par soucis de simplicité, les acronymes suivants seront utilisés dans cette partie :

- CA-CV : contour actif basé région de Chan et Vese ;
- CA-PPV : contour actif guidé par l'algorithme des k-Plus-Proches-Voisins ;
- CA-SVM : contour actif guidé par machine à vecteurs supports ;
- CA-RNA : contour actif guidé par réseau de neurones artificiels ;
- CA-PLS : contour actif supervisé par programmation linéaire simple.

4.5.1 Segmentation d'images synthétiques

Afin de confirmer l'indépendance des modèles proposés par rapport aux types de textures segmentés, un protocole de test sur des images synthétiques a été développé. Le but des contours actifs guidés par classificateur est de détecter la ou les frontières séparant deux textures distinctes. Ainsi, quatre séries d'images composées chacune de deux textures extraites de la base de Brodatz [Brodatz, 1966] ont été créées. La première image de chaque série constitue l'image d'apprentissage, à partir de laquelle l'ensemble $\mathbf{X}$ sera extrait afin que le classificateur détermine la règle permettant l'affectation correcte de chaque pixel. Une fois le classificateur appris, le contour actif est lancé sur le reste de la série afin de réaliser les segmentations désirées.

Intégrer exactement les deux textures présentes dans l'image d'apprentissage dans les autres images de test de la série ne présente aucun intérêt. En effet, le contour actif n'éprouvera que peu de difficultés à réaliser la segmentation car pour le classificateur, cela reviendra à réaliser une classification sur l'ensemble d'apprentissage. Les résultats seront donc toujours excellents. De plus dans des images naturelles de même type (deux échographies par exemple), les textures présentes sont proches, mais très rarement identiques. Les modèles développés doivent donc être capables de segmenter des textures dont les caractéristiques sont seulement proches de celles présentes dans l'image d'apprentissage. Dans le but de tester cette propriété, une série de six images de test a été réalisée pour chaque image d'apprentissage, chaque image étant composée des deux textures apprises. Les six images sont ensuite bruitées graduellement de telle sorte que la première possède un bruit nul et

4.5. SEGMENTATION D'IMAGES TEXTURÉES 2D PAR CONTOUR ACTIF SUPERVISÉ

la dernière un bruit maximal. Le bruit utilisé modifie aléatoirement les pixels de l'image (chaque pixel possède une probabilité fixe d'être modifié de 0, 2). Pour chaque pixel modifié, la variation de niveau de gris à appliquer est également tirée aléatoirement sur l'intervalle $[-51\alpha; +51\alpha]$, avec $\alpha \in [0; 5] \cap \mathbb{R}$. Ainsi, le bruit minimal sera obtenu pour $\alpha = 0$ (bruit nul), et le bruit maximal (correspondant à une grande variation du niveau de gris des pixels modifiés) pour $\alpha = 5$. Les nouvelles valeurs de niveaux de gris sont ensuite bornées sur $[0; 255]$. Chaque image de la série correspond à une variation croissante de α par pas de 1. La figure (4.8) illustre, pour chaque série créée, l'image d'apprentissage, l'image test possédant le bruit minimal ($\alpha = 0$), et celle possédant le bruit maximal ($\alpha = 5$).

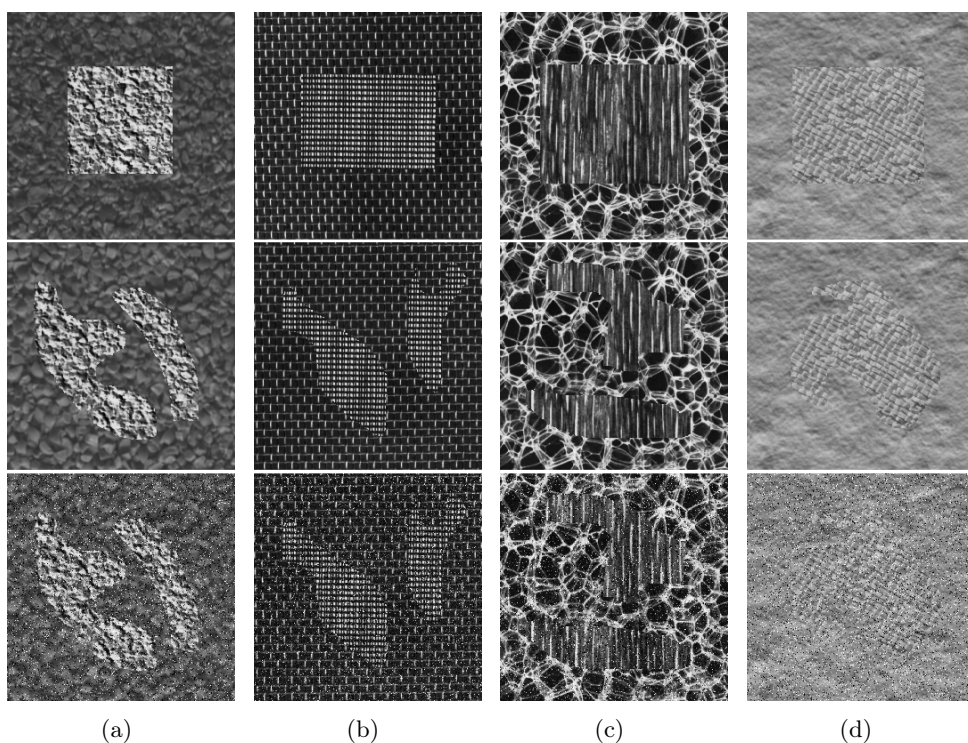


FIG. 4.8 – Illustration des quatre séries d'images synthétiques créées (une série par colonne). La première ligne représente les images d'apprentissage, la ligne du milieu les images de test possédant un bruit nul et la ligne du bas les images de test possédant le bruit maximal.

Pour chaque classificateur, l'ensemble d'apprentissage $\mathbf{X}$ est systématiquement divisé en deux sous-ensembles ($\mathbf{X}_l$ et $\mathbf{X}_e$). La configuration de paramètres obtenant le meilleur taux de classification sur l'ensemble $\mathbf{X}_e$ est conservée et utilisée pour réaliser la segmentation de la série correspondante (voir au 3.2.2.1 pour plus de détails). Pour l'algorithme des k-PPV, des valeurs de k allant de 1 à 15 (par pas de 2) ont été testées. Les différentes configurations de SVM ont été apprises avec les paramètres $C \in \{1; 5; 10; 15\}$ et $\sigma \in \{0, 1; 0, 5; 1; 1, 5; 2\}$. Enfin, dix réseaux de neurones ont également été appris avec un pas d'apprentissage de 0, 1, possédant chacun un nombre différent de neurones sur la couche cachée (variant de 1 à 10).

4.5. SEGMENTATION D'IMAGES TEXTURÉES 2D PAR CONTOUR ACTIF SUPERVISÉ

Le tableau (4.4) fournit l'évaluation des résultats de segmentation des quatre méthodes sur les deux premières séries d'images et le tableau (4.5) sur les deux dernières. Les figures (4.9) à (4.12) présentent, pour les images de valeur de bruit 0, 3 et 5 de chaque série, la vérité terrain, les segmentations obtenues par le modèle de Chan et Vese ainsi que celles obtenues par le modèle guidé par réseau de neurones. Toutes les images résultats sont disponibles dans l'annexe C.

Série 1 (Figure (4.8.a))				Série 2 (Figure (4.8.b))			
Bruit	Méthode	PRA_m	d_{mga}	Bruit	Méthode	PRA_m	d_{mga}
$\alpha = 0$	CA-CV	74,6	7,79	$\alpha = 0$	CA-CV	87,21	22,05
	CA-PPV	45,85	2,49		CA-PPV	36,74	2,14
	CA-SVM	36,24	1,87		CA-SVM	30,00	1,89
	CA-RNA	31,98	1,83		CA-RNA	35,21	1,99
$\alpha = 1$	CA-CV	74,16	8,79	$\alpha = 1$	CA-CV	89,95	21,40
	CA-PPV	47,55	2,60		CA-PPV	32,90	2,13
	CA-SVM	37,55	1,98		CA-SVM	30,15	1,92
	CA-RNA	32,87	1,88		CA-RNA	31,25	1,99
$\alpha = 2$	CA-CV	79,58	7,55	$\alpha = 2$	CA-CV	87,94	19,16
	CA-PPV	58,68	3,93		CA-PPV	32,76	2,43
	CA-SVM	49,4	2,76		CA-SVM	33,89	2,29
	CA-RNA	41,8	2,35		CA-RNA	31,53	2,19
$\alpha = 3$	CA-CV	81,14	8,24	$\alpha = 3$	CA-CV	89,48	23,12
	CA-PPV	84,57	14,08		CA-PPV	39,51	3,27
	CA-SVM	67,12	7,04		CA-SVM	43,95	3,41
	CA-RNA	57,21	4,50		CA-RNA	40,69	3,23
$\alpha = 4$	CA-CV	82,34	8,39	$\alpha = 4$	CA-CV	88,94	20,60
	CA-PPV	97,16	55,23		CA-PPV	79,58	14,45
	CA-SVM	81,27	13,31		CA-SVM	78,20	12,7
	CA-RNA	68,93	7,18		CA-RNA	78,21	13,32
$\alpha = 5$	CA-CV	85,47	9,46	$\alpha = 5$	CA-CV	90,29	33,49
	CA-PPV	98,18	63,22		CA-PPV	93,23	45,56
	CA-SVM	78,3	10,55		CA-SVM	94,45	49,3
	CA-RNA	59,45	6,02		CA-RNA	93,64	45,69

TAB. 4.4 – Comparaison de la précision de segmentation des quatre méthodes testées sur les deux premières images synthétiques. PRA_m correspond au critère de Pratt modifié, et d_{mga} à la mesure générique d'anomalies. Les mesures de précision sont multipliées par un facteur 100.

4.5. SEGMENTATION D'IMAGES TEXTURÉES 2D PAR CONTOUR ACTIF SUPERVISÉ

Série 3 (Figure (4.8.c))				Série 4 (Figure (4.8.d))			
Bruit	Méthode	PRA_m	d_{mga}	Bruit	Méthode	PRA_m	d_{mga}
$\alpha = 0$	CA-CV	85,09	27,16	$\alpha = 0$	CA-CV	89,15	32,76
	CA-PPV	64,32	5,26		CA-PPV	45,42	1,53
	CA-SVM	68,80	10,33		CA-SVM	50,54	2,82
	CA-RNA	35,14	3,12		CA-RNA	47,61	1,39
$\alpha = 1$	CA-CV	85,33	35,75	$\alpha = 1$	CA-CV	89,78	35,99
	CA-PPV	66,20	5,85		CA-PPV	36,09	1,56
	CA-SVM	68,29	10,27		CA-SVM	34,93	1,89
	CA-RNA	36,91	3,22		CA-RNA	35,77	1,51
$\alpha = 2$	CA-CV	85,01	29,41	$\alpha = 2$	CA-CV	90,19	40,79
	CA-PPV	69,87	7,80		CA-PPV	76,01	8,09
	CA-SVM	60,80	6,95		CA-SVM	42,78	2,29
	CA-RNA	45,72	3,44		CA-RNA	77,81	8,82
$\alpha = 3$	CA-CV	85,93	27,06	$\alpha = 3$	CA-CV	91,88	38,49
	CA-PPV	76,27	10,35		CA-PPV	97,58	65,16
	CA-SVM	64,74	6,46		CA-SVM	97,25	59,73
	CA-RNA	56,82	4,28		CA-RNA	66,01	5,98
$\alpha = 4$	CA-CV	85,44	27,16	$\alpha = 4$	CA-CV	91,31	33,77
	CA-PPV	78,52	12,22		CA-PPV	99,12	67,54
	CA-SVM	64,85	5,89		CA-SVM	95,42	59,42
	CA-RNA	58,6	4,49		CA-RNA	82,38	14,09
$\alpha = 5$	CA-CV	86,67	34,61	$\alpha = 5$	CA-CV	94,17	36,86
	CA-PPV	83,56	19,95		CA-PPV	99,76	67,56
	CA-SVM	66,28	6,17		CA-SVM	81,48	23,89
	CA-RNA	53,83	4,42		CA-RNA	89,48	24,98

TAB. 4.5 – Comparaison de la précision de segmentation des quatre méthodes testées sur les deux premières images synthétiques. Les mesures de précision sont multipliées par un facteur 100.

Pour la première série d'images, le paramètre k de l'algorithme k-PPV ayant obtenu le meilleur taux de classification sur $\mathbf{X}_e$ vaut 13 (98,47 % de reconnaissance), le réseau de neurones possède 4 neurones sur la couche cachée (99,11 % de reconnaissance), et les deux paramètres des SVM ont été choisis tels que $C = 10$ et $\sigma = 1,5$ (99,05 % de reconnaissance sur $\mathbf{X}_e$).

Les résultats de segmentation, présentés figure (4.9) et dans la partie gauche du tableau (4.4), mettent en avant la grande précision du CA-RNA. Ce dernier réalise en effet les meilleures segmentations, quel que soit le niveau du bruit présent dans l'image, et augmente la précision du modèle de Chan et Vese de 46 % en moyenne (39 % pour le critère de Pratt et 53 % pour la mesure générique d'anomalies). Les modèles CA-SVM et CA-PPV sont efficaces lorsque le niveau de bruit n'est pas trop important. Quand $\alpha > 2$ pour le CA-PPV et $\alpha > 3$ pour le CA-SVM, les segmentations deviennent moins précises qu'avec le modèle de Chan et Vese.

4.5. SEGMENTATION D'IMAGES TEXTURÉES 2D PAR CONTOUR ACTIF SUPERVISÉ

En revanche, il est intéressant de noter que bien que la précision des modèles proposés décroît généralement à mesure que le bruit augmente, celle du modèle de Chan et Vese reste constante. Ce résultat est logique, car plus le bruit augmente, plus les caractéristiques de textures des pixels de l'image de test s'éloignent de celles de l'image d'apprentissage. Le contour actif doit alors traiter des données qu'il ne connaît pas, ce qui rend la segmentation plus hasardeuse. Le modèle de Chan et Vese, quant à lui, ne possède aucune connaissance *a priori* sur les régions recherchées et le caractère uniforme du bruit ajouté à l'image ne modifie que très peu la différence entre les moyennes d'intensités des régions créées par la courbe. Ce modèle reste donc très peu sensible au bruit. De plus, les deux textures présentes dans l'image possèdent des intensités moyennes très différentes, ce qui avantage le modèle de Chan et Vese et explique ses bons résultats.

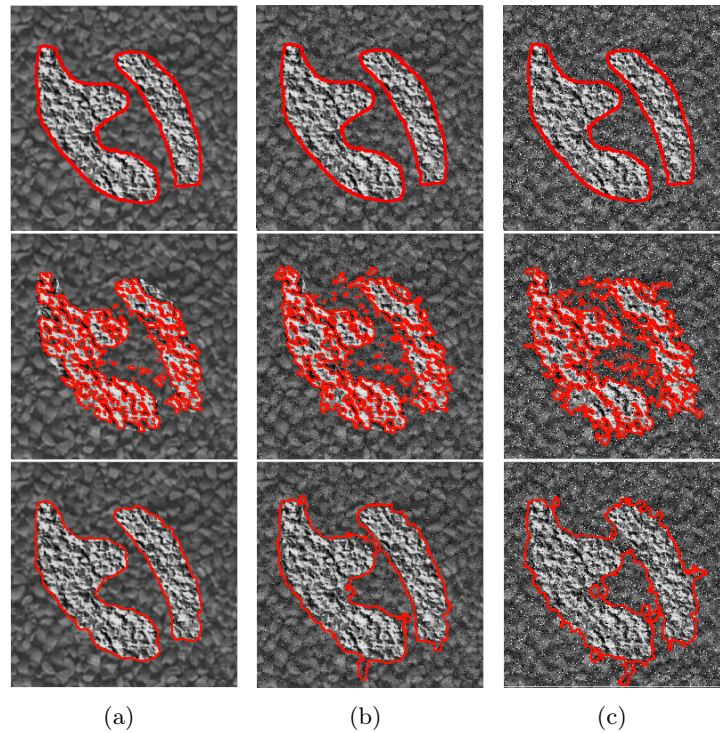


FIG. 4.9 – Résultats de segmentation sur la première série d'images. a) $\alpha = 0$. b) $\alpha = 3$. c) $\alpha = 5$. La première ligne illustre les vérités terrain, la deuxième les segmentations du modèle de Chan-Vese et la troisième celles du modèle guidé par réseau de neurones.

La deuxième série d'images est particulièrement difficile car elle comporte deux textures très proches, l'une étant presque une simple inversion de l'autre au niveau des intensités des pixels. Pour cette raison, à partir d'un niveau de bruit de 5, les résultats ne sont que peu interprétables car les segmentations s'avèrent très mauvaises pour tous les modèles. Tous les résultats de segmentation sont présentés dans la partie droite du tableau (4.4) et sur la figure (4.10). Le classificateur k-PPV est réglé avec 9 voisins (96,5 % de reconnaissance sur $\mathbf{X}_e$), les SVM sont réglées telles que $C = 10$ et $\sigma = 0,5$ (97,7 % de reconnaissance) et le réseau de neurones possède 8 neurones cachés (pour un taux de reconnaissance de 97,2 % sur $\mathbf{X}_e$).

4.5. SEGMENTATION D'IMAGES TEXTURÉES 2D PAR CONTOUR ACTIF SUPERVISÉ

Sur les quatre premières images de la série, les trois méthodes proposées améliorent la précision obtenue avec le modèle de Chan et Vese mais aucune ne ressort véritablement comme la plus précise. Les résultats sont très proches, à tel point que sur la troisième image les deux critères d'évaluation ne fournissent pas exactement les mêmes données (le critère de Pratt considère le CA-PPV comme le plus précis, alors que le CA-RNA est plébiscité par la mesure générique d'anomalies). La diminution de la précision avec l'augmentation du bruit se retrouve également sur cette série, avec une cinquième image où les modèles proposés ne parviennent pas à améliorer le résultat du contour actif de Chan et Vese. Néanmoins, il est légitime de s'interroger sur la pertinence des évaluations lorsque toutes les segmentations sont mauvaises, car en observant la figure (4.10), il est difficile de déterminer quel résultat est le plus précis sur l'image possédant un niveau de bruit de 5.

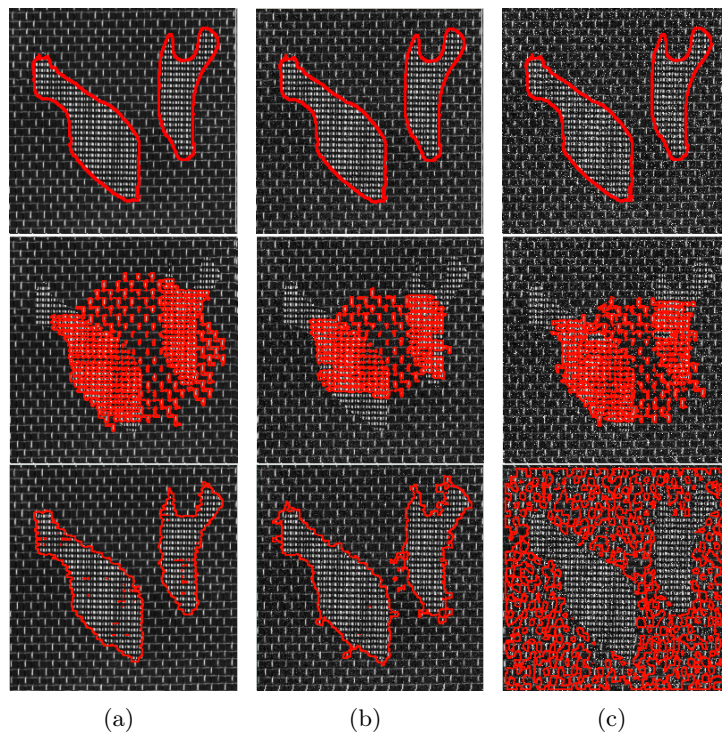


FIG. 4.10 – Résultats de segmentation sur la deuxième série d'images synthétiques.

L'apprentissage réalisé sur la troisième série (figure (4.11) et partie gauche du tableau (4.5)) a conduit à faire évoluer l'algorithme k-PPV avec 3 voisins (pour un taux de reconnaissance de 87,3 % sur $\mathbf{X}_e$), les SVM avec $C = 10$ et $\sigma = 0,5$ (94,5 % de reconnaissance) et le réseau de neurones avec 8 neurones sur la couche cachée (93 % de reconnaissance).

Les contours actifs guidés par classificateur améliorent systématiquement la précision du modèle de Chan et Vese. Les meilleurs résultats sont obtenus avec le CA-RNA, avec une amélioration moyenne de la précision du modèle de Chan et Vese de 65 % (44 % pour le critère de Pratt et 87 % pour la mesure générique d'anomalies). Le CA-SVM est légèrement moins précis, avec une amélioration moyenne de 50 %. Enfin, le CA-PPV représente la moins bonne performance parmi les trois modèles avec une amélioration de 43 %. Toutefois, les résultats du CA-SVM sont irréguliers. Sa précision de segmentation

4.5. SEGMENTATION D'IMAGES TEXTURÉES 2D PAR CONTOUR ACTIF SUPERVISÉ

s'avère être la plus mauvaise des trois classificateurs sur les deux premières images (les moins bruitées), puis celle-ci s'améliore lorsque le bruit s'intensifie.

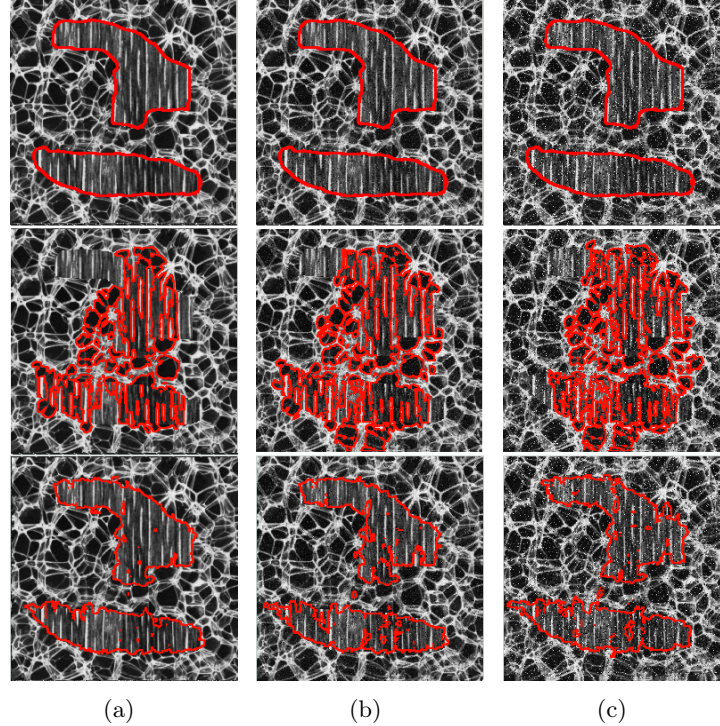


FIG. 4.11 – Résultats de segmentation sur la troisième série d'images synthétiques.

La dernière série d'images présente également deux textures très similaires, ce qui rend la segmentation très difficile. Les résultats de segmentation sont présentés figure (4.12) et partie droite du tableau (4.5). L'algorithme k-PPV évolue avec 9 voisins (98,1 % de reconnaissance sur $\mathbf{X}_e$), les SVM utilisées sont réglées telles que $C = 10$ et $\sigma = 0,5$ (98,3 % de reconnaissance) et le réseau de neurones possède 9 neurones cachés (98,6 % de reconnaissance).

Le contour actif de Chan et Vese éprouve de grandes difficultés dès la première image de la série, car les deux textures possèdent des moyennes d'intensités très proches. Le CA-PPV permet d'améliorer la précision uniquement sur les trois premières images, les résultats de segmentation devenant plus hasardeux dès que le niveau de bruit dépasse 2. Les meilleurs résultats sont obtenus pour les approches CA-SVM et CA-RNA. En revanche, malgré une amélioration moyenne de la précision d'environ 29 % par rapport au modèle de Chan et Vese, la première demeure irrégulière, avec de bons résultats sur les trois premières images, une très forte baisse sur les images de bruit 3 et 4 (avec des résultats inférieurs au modèle de Chan et Vese) et enfin une amélioration sur la dernière image. La deuxième en revanche, bien que n'étant pas toujours la mieux classée, reste régulière et permet une amélioration constante d'environ 50 % de la précision du modèle de Chan et Vese.

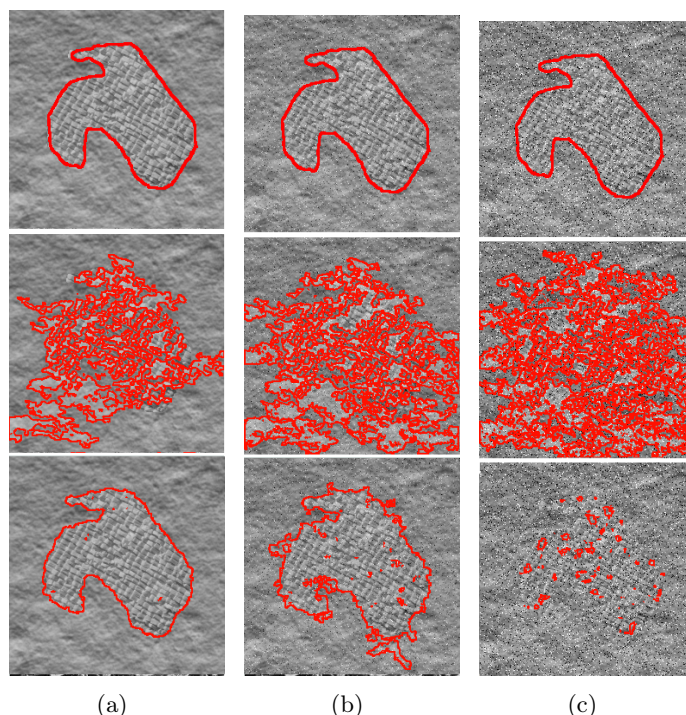


FIG. 4.12 – Résultats de segmentation sur la quatrième série d'images synthétiques.

Les figures (4.13) et (4.14) résument les capacités des quatre modèles comparés en présentant, pour chaque critère d'évaluation, les moyennes des précisions de segmentation de chaque modèle en fonction du niveau de bruit dans l'image. Le premier constat est que le critère de Pratt modifié et la mesure générique d'anomalies fournissent des évaluations cohérentes et font ressortir les mêmes tendances. Cela permet d'assurer qu'aucune approche n'a été avantagée par un critère particulier.

D'une manière générale, le modèle de Chan et Vese souffre d'une mauvaise précision, mais apparaît comme le plus régulier et le moins sensible au bruit. Le CA-PPV permet d'obtenir une meilleure précision lorsque le bruit n'est pas trop élevé (jusqu'au niveau 3 ou 4), mais devient très instable ensuite. Le CA-SVM améliore presque systématiquement la précision du modèle de Chan et Vese mais souffre d'une forte irrégularité dans ses résultats. En effet, lorsque le bruit atteint le niveau $\alpha = 4$, ses résultats sont équivalents à ceux du contour actif de Chan et Vese mais sa précision s'améliore par la suite quand le bruit atteint le niveau $\alpha = 5$. Enfin le CA-RNA obtient les meilleurs résultats de segmentation. De plus, cette approche reste régulière et sa précision décroît de manière quasi-linéaire à mesure que le bruit augmente. Il est ainsi plus aisé de prévoir son comportement.

Les temps d'exécution des différents modèles n'ont pas été détaillés car le but de nos approches était principalement de fournir des résultats précis. Toutefois, même lorsque l'on ne se place pas dans le cadre d'applications temps réel, le temps de segmentation reste un critère important. Ainsi, le CA-RNA possède une vitesse équivalente à celle du modèle de Chan et Vese. Le CA-SVM est légèrement plus lent (en raison d'un processus de classification plus complexe des SVM). Enfin, l'utilisation de l'algorithme k-PPV ralentit le

4.5. SEGMENTATION D'IMAGES TEXTURÉES 2D PAR CONTOUR ACTIF SUPERVISÉ

contour actif de manière significative, car un tri de l'ensemble d'apprentissage est nécessaire lors de chaque classification de pixels (bien qu'en pratique l'ensemble ait été restreint de manière à rendre les segmentations plus rapides).

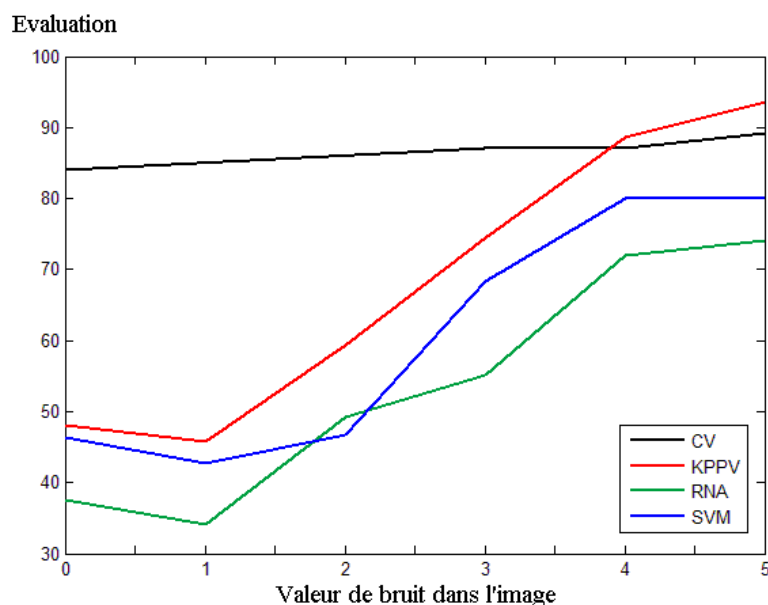


FIG. 4.13 – Comparaison des valeurs moyennes du critère de Pratt modifié en fonction du niveau de bruit introduit dans l'image pour chaque modèle.

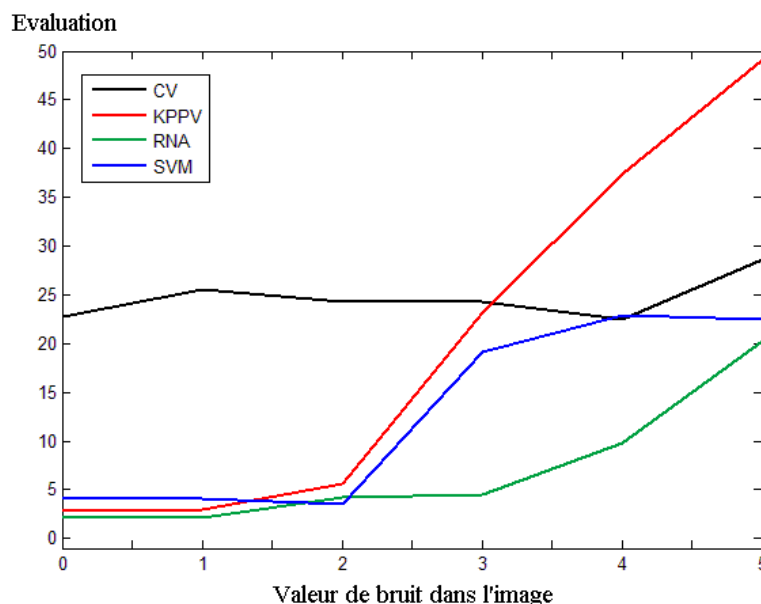


FIG. 4.14 – Comparaison des valeurs moyennes de la mesure générique d'anomalies en fonction du niveau de bruit introduit dans l'image pour chaque modèle.

Concernant les temps d'apprentissage, l'approche par k-PPV est immédiate (un simple

4.5. SEGMENTATION D'IMAGES TEXTURÉES 2D PAR CONTOUR ACTIF SUPERVISÉ

stockage de l'ensemble d'apprentissage est nécessaire), le CA-RNA est appris en moins d'une minute et le CA-SVM requiert un apprentissage plus long (de l'ordre de la demi-heure), dû en grande partie à la taille conséquente de l'ensemble d'apprentissage.

Il est également important de noter que les modèles proposés bénéficient d'une très faible dépendance à l'initialisation. En effet, le contour actif possédant une connaissance *a priori* de l'objet recherché, la courbe initiale ne nécessite que quelques pixels à l'intérieur de celui-ci pour garantir une bonne segmentation. La partie de la courbe placée à l'extérieur de l'objet va se resserrer et s'arrêter sur sa frontière (ou complètement disparaître), alors que la partie à l'intérieur va automatiquement grandir jusqu'à l'atteindre. A l'opposé, le modèle de Chan et Vese est très sensible au placement de la courbe d'origine, car celle-ci va déterminer les moyennes d'intensités initiales des deux régions créées, influençant très fortement la segmentation. La figure (4.15) illustre cette propriété.

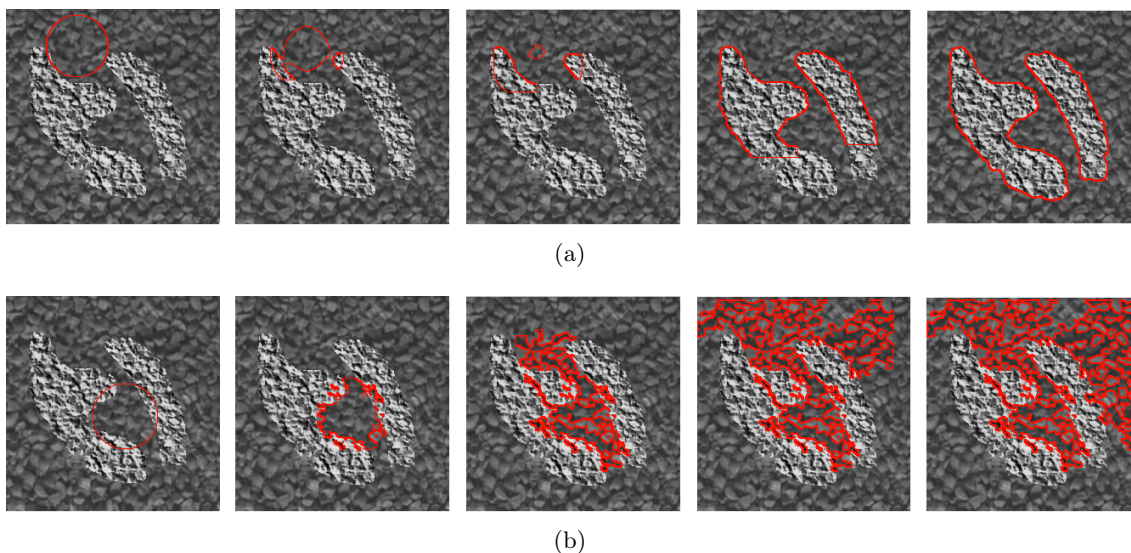


FIG. 4.15 – Illustration de la faible dépendance à l'initialisation des contours actifs guidés par classificateur. a) CA-RNA. b) Modèle de Chan et Vese. Seuls quelques pixels appartenant à l'objet recherché doivent être situés à l'intérieur de la courbe initiale pour assurer une segmentation correcte du modèle guidé par classificateur. En revanche, le modèle de Chan et Vese demeure très sensible à l'initialisation car elle détermine les intensités recherchées.

Les différents résultats présentés ici mettent en avant les bonnes performances obtenues par le CA-RNA. En effet, ses résultats de segmentation sont systématiquement les plus précis, sa vitesse de segmentation est équivalente à celle du modèle de Chan-Vese et son apprentissage est très rapide. Le CA-SVM obtient également de bons résultats mais son irrégularité le rend légèrement imprévisible et son temps d'apprentissage conséquent le désavantage. Le CA-PPV, quant à lui, souffre d'une trop faible capacité à généraliser les données apprises (illustrée par ses difficultés à segmenter correctement les images très bruitées) et d'une vitesse faible. Le modèle de Chan et Vese s'est révélé le moins précis mais reste très peu sensible au bruit présent dans l'image.

4.5.2 Segmentation d'images échographiques

L'analyse d'images médicales représente un important domaine de la vision par ordinateur. Elle permet de fournir une aide au diagnostic aux médecins par l'intermédiaire de différents traitements réalisés sur l'image. En comparaison à des techniques telles la radiographie ou l'Imagerie à Résonance Magnétique (IRM), l'imagerie ultrasonore permet d'obtenir des images en temps réel pour un coût faible et de manière non-invasive pour le patient. En contrepartie, la présence de bruit et d'artefacts (cônes d'ombres, atténuation de luminosité, etc.) rendent généralement l'interprétation des images échographiques ainsi obtenues plus délicate.

Dans cette partie, les trois modèles guidés par classificateur ainsi que l'approche par programmation linéaire sont comparés au contour actif de Chan et Vese sur des images échographiques de la peau. Ces dernières possèdent la particularité d'être très bruitées, les frontières des différents objets n'étant que rarement clairement définies. Le but de ce type d'application est de fournir au médecin une aide au diagnostic par le biais de mesure sur la pathologie segmentée.

La figure (4.16) illustre l'image d'apprentissage utilisée ainsi que les régions C^{in} et C^{out} , et la figure (4.17) décrit la série de quatre images sur laquelle les modèles ont été testés.

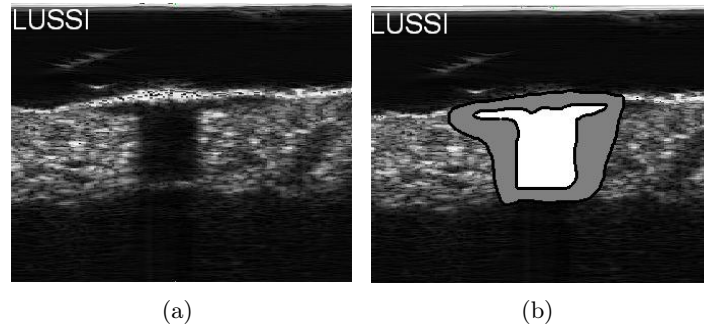


FIG. 4.16 – Image d'apprentissage pour la série d'images échographiques. a) Image d'origine. b) Régions C^{in} (blanc) et C^{out} (gris) créées pour réaliser l'ensemble d'apprentissage.

4.5. SEGMENTATION D'IMAGES TEXTURÉES 2D PAR CONTOUR ACTIF SUPERVISÉ

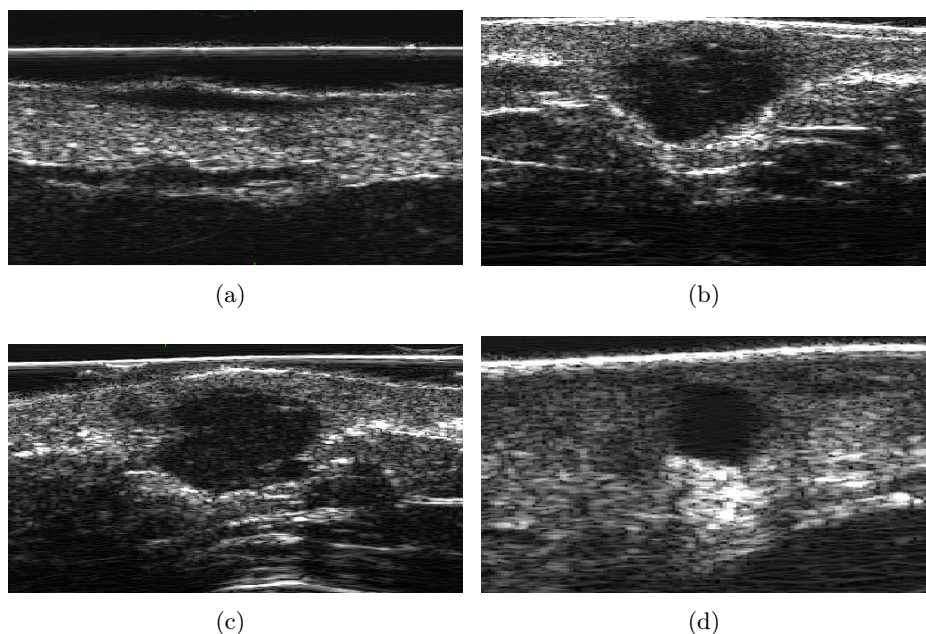


FIG. 4.17 – Images échographiques utilisées pour comparer les différents modèles.

Le tableau (4.6) présente les évaluations de la qualité des segmentations pour les cinq approches. Les figures (4.18) à (4.21) illustrent, pour chaque image de test, la vérité terrain, la segmentation obtenue par le modèle de Chan et Vese et celle obtenue avec le contour actif supervisé par programmation linéaire (un zoom a été réalisé autour de la zone d'intérêt). Toutes les segmentations sont disponibles en annexe D. La différence d'ordre de grandeur entre les deux critères d'évaluation utilisés s'explique par la faible aire des objets segmentés en comparaison de la taille de l'image. En effet, la mesure générique d'anomalies étant normalisée par rapport au nombre de pixels présents dans l'image, sa valeur devient très faible.

Les approches proposées améliorent systématiquement la précision du modèle de Chan et Vese, bien que celui-ci reste très efficace sur ce type d'image (exception faite du contour actif guidé par k-PPV sur la deuxième image). En effet, l'objet à segmenter possède une moyenne des intensités de niveaux gris très différente de celle de l'arrière plan. Cela permet donc au modèle de Chan et Vese de rester compétitif. Néanmoins, les modèles CA-SVM et CA-RNA sont les plus efficaces avec une amélioration moyenne de la précision de 46,2 % pour le CA-SVM (33 % pour le critère de Pratt modifié et 59 % pour la mesure générique d'anomalies) et 44,6 % pour le CA-RNA (31 % pour Pratt et 45 % pour la mesure générique d'anomalies). Le CA-PLS est légèrement moins précis, avec une amélioration moyenne de 40 %. Le CA-PPV n'atteint pas les mêmes résultats, avec une amélioration moyenne de seulement 16 %.

Ces résultats sont cohérents avec ceux obtenus sur les images synthétiques et permettent de situer les performances du CA-PLS par rapport aux autres approches. Les modèles CA-RNA et CA-SVM fournissent des résultats de segmentation plus proches que précédemment mais avec un apprentissage rapide et une vitesse de segmentation élevée, le contour actif guidé par réseau de neurones peut être considéré comme le plus performant.

4.5. SEGMENTATION D'IMAGES TEXTURÉES 2D PAR CONTOUR ACTIF SUPERVISÉ

Images échographiques			
Image	Méthode	PRA_m	d_{mga}
Echo 1 Figure (4.18)	CA-CV	66,91	1,17
	CA-PLS	48,56	0,64
	CA-PPV	69,85	1,20
	CA-SVM	48,51	1,05
	CA-RNA	50,91	0,53
Echo 2 Figure (4.19)	CA-CV	82,07	2,85
	CA-PLS	59,48	1,38
	CA-PPV	73,58	2,26
	CA-SVM	60,06	1,42
	CA-RNA	66,51	1,64
Echo 3 Figure (4.20)	CA-CV	66,95	2,31
	CA-PLS	40,90	1,25
	CA-PPV	58,05	1,85
	CA-SVM	44,20	1,10
	CA-RNA	43,86	1,10
Echo 4 Figure (4.21)	CA-CV	92,55	2,61
	CA-PLS	67,29	1,15
	CA-PPV	74,73	1,24
	CA-SVM	51,36	0,53
	CA-RNA	48,32	0,47

TAB. 4.6 – Comparaison de la précision de segmentation des méthodes testées sur la série d'images échographiques. Les mesures de précision sont multipliées par un facteur 100.

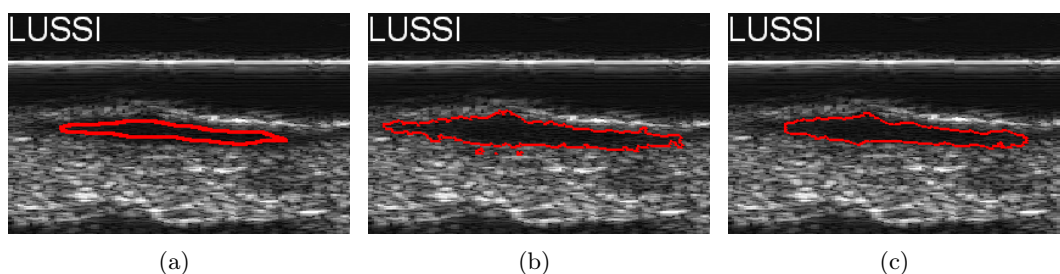


FIG. 4.18 – Résultats de segmentation sur la première image échographique. a) Vérité terrain. b) Modèle de Chan et Vese. c) CA-PLS.

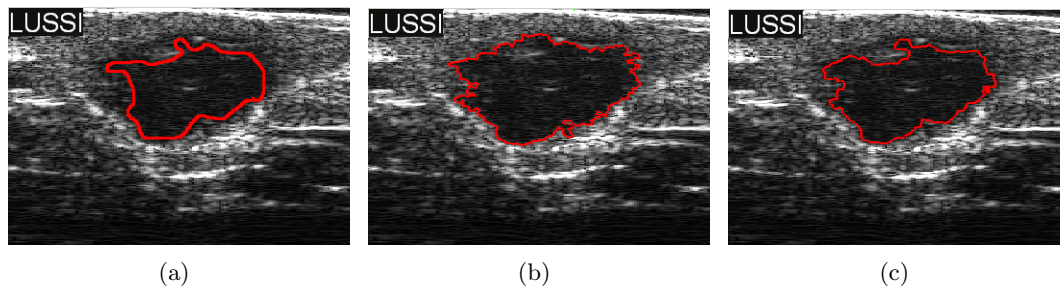


FIG. 4.19 – Résultats de segmentation sur la deuxième image échographique.

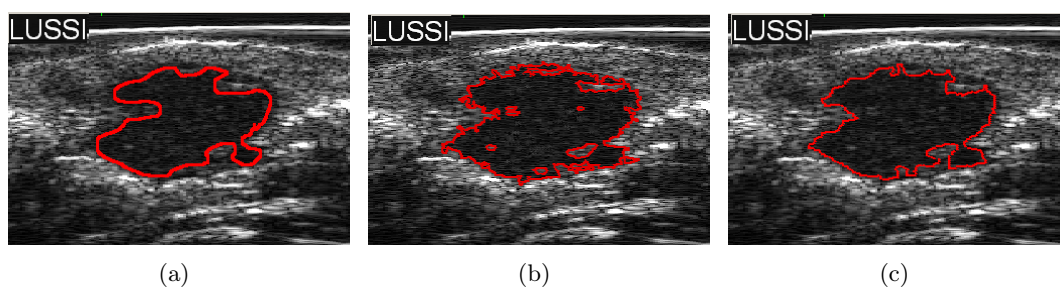


FIG. 4.20 – Résultats de segmentation sur la troisième image échographique.

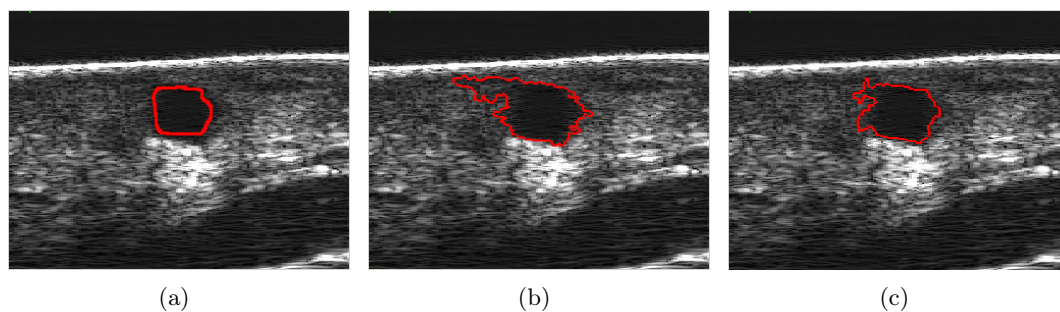


FIG. 4.21 – Résultats de segmentation sur la quatrième image échographique.

4.5.3 Segmentation de zones urbaines dans des images satellites

Les contours actifs guidés par classificateur ont également été testés dans une problématique de segmentation de zones urbaines dans des images satellites. Toutes les images sont issues de la base ASTER [Jimenez-Munoz et Sobrino, 2007].

La figure (4.22) illustre l'image segmentée utilisée pour créer l'ensemble d'apprentissage des classificateurs, et la figure (4.23) les quatre images de test.

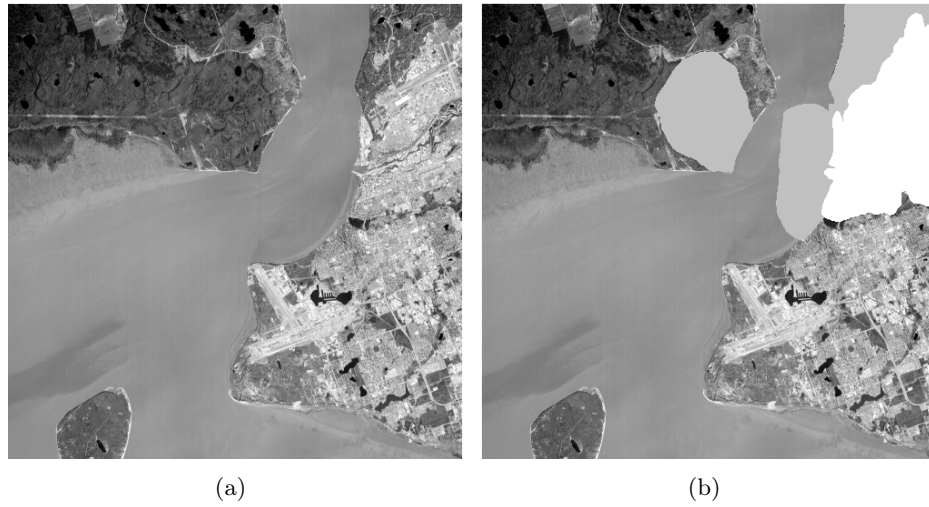


FIG. 4.22 – Image d'apprentissage pour la série d'images satellites. a) Image d'origine. b) Régions C^{in} (blanc) et C^{out} (gris) créées pour réaliser l'ensemble d'apprentissage.

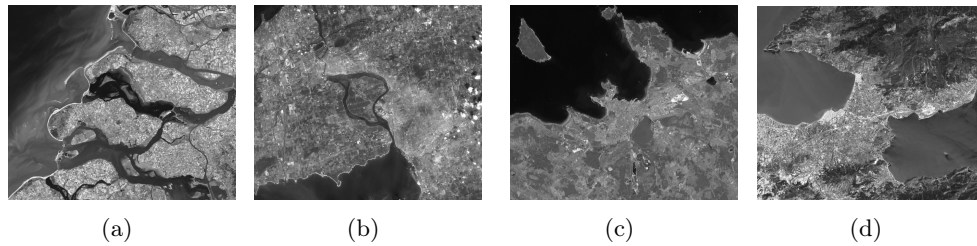


FIG. 4.23 – Images satellites utilisées pour tester les différents modèles.

Le tableau (4.7) présente les résultats des trois contours actifs guidés par classificateur et du modèle de Chan et Vese. Les figures (4.24) à (4.27) illustrent, pour chaque image test, la vérité terrain, la segmentation obtenue par le modèle de Chan-Vese et celle obtenue par le contour actif guidé par réseau de neurones.

4.5. SEGMENTATION D'IMAGES TEXTURÉES 2D PAR CONTOUR ACTIF SUPERVISÉ

Images satellites ASTER (Figure (4.8.c))			
Image	Méthode	PRA_m	d_{mga}
ASTER 1 Figure (4.24)	CA-CV	64,12	12,04
	CA-PPV	59,63	9,36
	CA-SVM	52,11	12,27
	CA-RNA	49,23	9,10
ASTER 2 Figure (4.25)	CA-CV	92,02	47,57
	CA-PPV	76,67	13,13
	CA-SVM	76,90	12,79
	CA-RNA	77,06	12,70
ASTER 3 Figure (4.26)	CA-CV	77,59	14,51
	CA-PPV	63,37	10,68
	CA-SVM	58,74	9,69
	CA-RNA	59,64	9,57
ASTER 4 Figure (4.27)	CA-CV	82,73	15,59
	CA-PPV	74,40	12,25
	CA-SVM	70,15	11,93
	CA-RNA	69,98	10,21

TAB. 4.7 – Comparaison de la précision de segmentation des méthodes testées sur la série d'images satellites. Les mesures de précision sont multipliées par un facteur 100.

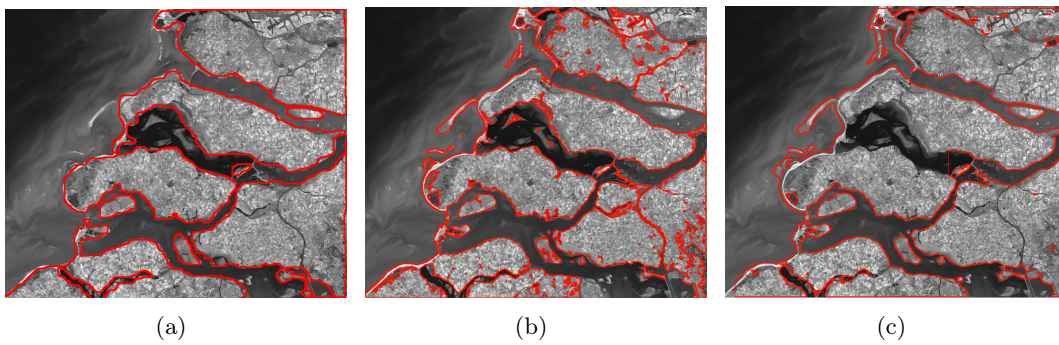


FIG. 4.24 – Résultats de segmentation sur la première image satellite. a) Vérité terrain. b) Modèle de Chan et Vese. c) CA-RNA.

4.5. SEGMENTATION D'IMAGES TEXTURÉES 2D PAR CONTOUR ACTIF SUPERVISÉ

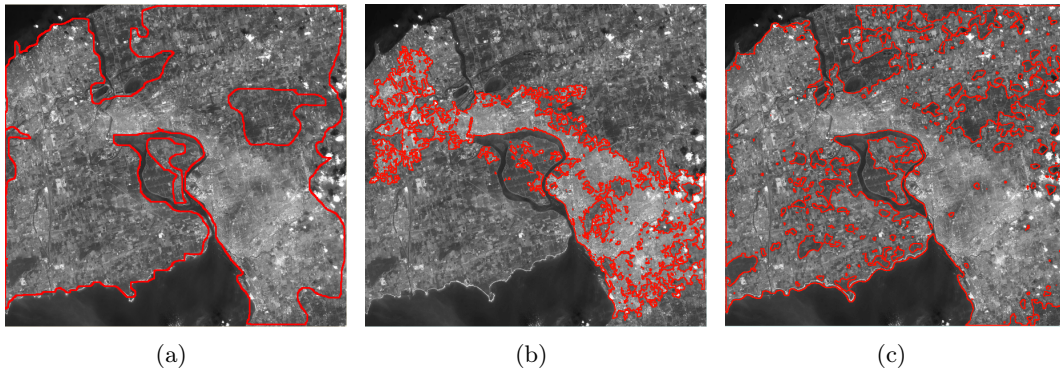


FIG. 4.25 – Résultats de segmentation sur la deuxième image satellite.

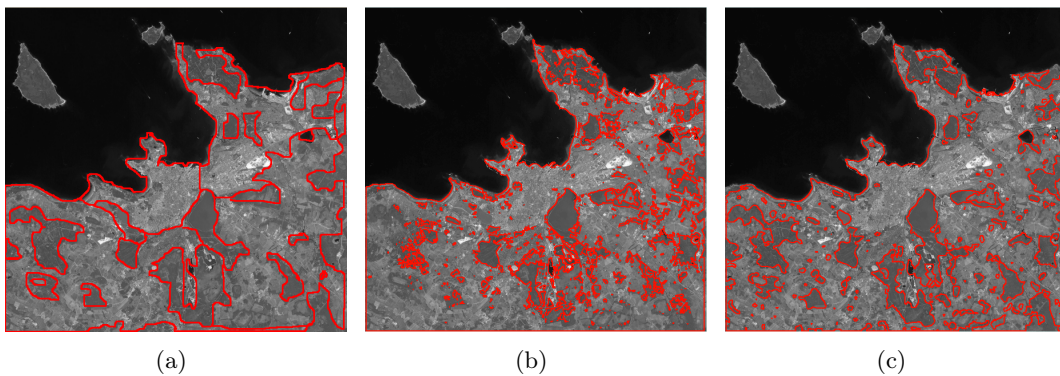


FIG. 4.26 – Résultats de segmentation sur la troisième image satellite.

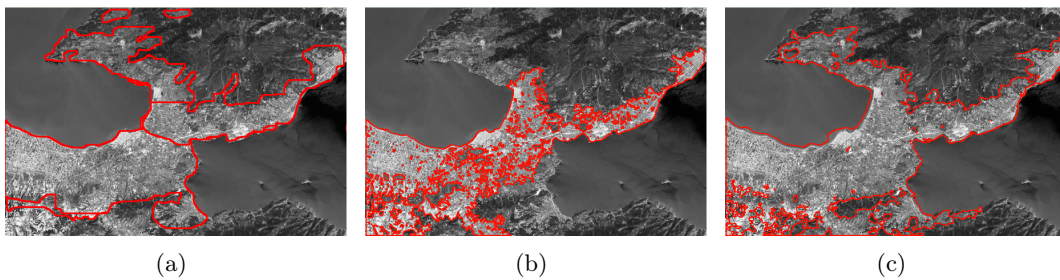


FIG. 4.27 – Résultats de segmentation sur la quatrième image satellite.

Les résultats indiquent une nouvelle fois une amélioration de la précision pour les segmentations obtenues par les contours actifs guidés par classificateur. Même si le CA-RNA fournit les meilleurs résultats, les trois modèles demeurent très proches, à tel point que sur la deuxième et la troisième image, les deux critères d'évaluation sont en désaccord sur l'approche la plus précise. Ainsi, le CA-RNA améliore la qualité de segmentation du modèle de Chan et Vese de 36 % en moyenne, le CA-SVM de 33 % et le CA-PPV de 22 %.

4.5.4 Analyse des performances des modèles implicites supervisés

Dans cette deuxième partie du chapitre, les contours actifs implicites supervisés développés ont été appliqués à différentes problématiques de segmentation et comparés au modèle basé région de Chan et Vese. Les évaluations des résultats de segmentation mettent en avant la précision des approches proposées sur chaque type d'image et confirment que l'utilisation d'un processus supervisé permet de rendre les modèles indépendant par rapport aux types de textures traités.

Le CA-RNA peut être considéré comme le plus précis. La capacité de ce type de classificateur à pondérer les caractéristiques d'entrée est un avantage quand des coefficients d'Haralick s'avèrent peu fiables sur un certain type de texture. En outre, le réseau de neurones bénéficie d'un apprentissage rapide et ne ralentit pas la segmentation en raison d'une règle de classification de très faible complexité. Toutes ces propriétés en font le modèle le plus efficace.

Le CA-SVM obtient des résultats de segmentation très proches du CA-RNA. En revanche, une irrégularité additionnée à un temps d'apprentissage conséquent et une vitesse de segmentation plus faible le rendent moins efficace.

Le CA-PLS s'est avéré légèrement moins précis que les deux modèles précédents, mais son habilité à réaliser une sélection de caractéristiques demeure intéressante.

Enfin le CA-PPV peut être considérée comme le moins efficace. Il souffre d'une difficulté à généraliser les données apprises (phénomène mis en avant lors des tests sur les images synthétiques) et malgré un temps d'apprentissage quasi-immédiat, sa vitesse de segmentation est la plus faible en raison des nombreux tris réalisés.

Malgré une faible précision, le contour actif de Chan et Vese s'est révélé très peu sensible au bruit. Il est toutefois intéressant de noter que celui-ci souffre ici d'une approche basée sur les intensités de niveaux de gris et non sur des caractéristiques de textures. Dès lors, il apparaît de prime abord pertinent d'estimer qu'une comparaison par rapport à un autre modèle basé texture aurait été plus appropriée. Mais, malheureusement, une telle comparaison était délicate car utiliser une approche basée texture nous ramenait exactement au problème que nous avons choisi de résoudre en utilisant des classificateurs supervisés : la sélection des caractéristiques de textures pertinentes en fonction du type d'image traité. De plus, il était nécessaire d'utiliser une approche basée sur les coefficients d'Haralick afin de garantir que les différences de performance n'étaient pas introduites par les caractéristiques de textures utilisées. Ces différentes raisons nous ont conduits à comparer les modèles développés à une approche connue pour son efficacité et considérée comme un contour actif de référence dans la littérature.

Les différentes perspectives de développement des modèles implicites supervisés sont détaillées dans la conclusion de ce mémoire de thèse.

4.6 Segmentation d'images échographiques 3D par contour actif supervisé

La reconstruction d'objet dans les échographies 3D représente un outil d'aide au diagnostic précieux. En effet, les dermatologues doivent régulièrement enlever de manière chirurgicale des pathologies appelées *naevus* (les "grains de beauté"). Afin d'assurer l'ablation du *naevus* dans son intégralité, une grande marge de sécurité est prise lors de la chirurgie et une zone généralement beaucoup plus large est enlevée, entraînant parfois des conséquences esthétiques telles que de larges cicatrices. Afin de permettre au médecin de réaliser un diagnostic plus précis sur les dimensions du *naevus*, tout en évitant des techniques d'imagerie trop invasives, notre modèle de contour actif guidé par réseau de neurones a été étendu à la reconstruction dans des échographies 3D de la peau.

Une échographie 3D est constituée d'un empilement d'échographies 2D. En conséquence, la structure de données 3D correspond à une succession de coupes 2D. La figure (4.28) illustre la création d'une échographie 3D.

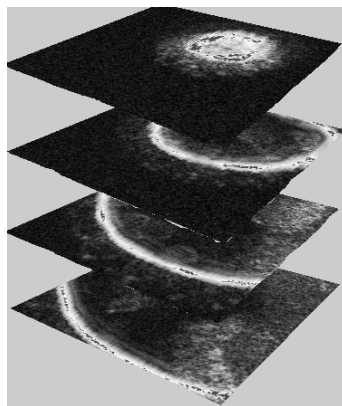


FIG. 4.28 – Illustration de la succession de coupes composant une image échographique 3D.

L'ensemble d'apprentissage est extrait de la segmentation manuelle d'une unique coupe 2D sélectionnée par l'utilisateur (voir figure (4.29)). Les coefficients d'Haralick sont alors calculés en trois dimensions afin de déterminer l'ensemble d'apprentissage du classificateur (les calculs de la matrice de cooccurrence et des coefficients d'Haralick reste inchangés, ce sont la transition U et la fenêtre de voisinage qui passent en trois dimensions). Une fois l'apprentissage réalisé, le CA-RNA (devenu surface active) est utilisé afin de segmenter la pathologie. Une fois l'évolution terminée, le modèle déformable est transformé en maillage.

Les figures (4.30) à (4.32) illustrent la segmentation réalisée sur la première échographie 3D. La figure (4.30) représente l'image échographique d'origine, la figure (4.31) la pathologie reconstruite avec la surface active et enfin la figure (4.32) le maillage réalisé. La segmentation de la deuxième échographie est présentée de manière similaire figures (4.33) à (4.35), la troisième échographie étant illustrée figures (4.36) à (4.38).

Une vérité terrain étant difficile à mettre en place pour ce type de données, les segmentations n'ont pu être évaluées par une métrique. Toutefois, les maillages finaux ont été

4.6. SEGMENTATION D'IMAGES ÉCHOGRAPHIQUES 3D PAR CONTOUR ACTIF SUPERVISÉ

soumis à des dermatologues et validés visuellement.

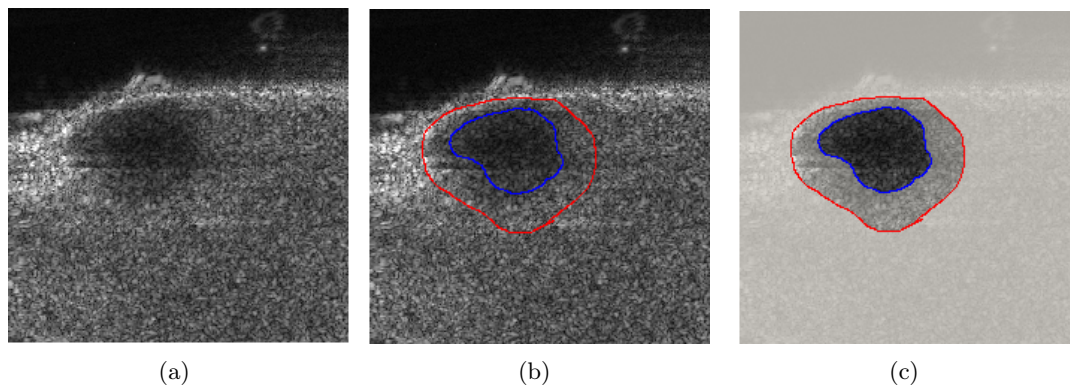


FIG. 4.29 – Illustration de la création de l'ensemble d'apprentissage sur une coupe de l'échographie 3D. a) Coupe initiale. b) Identification manuelle des régions C^{in} et C^{out} par leurs frontières (bleue pour C^{in} et rouge pour C^{out}). c) Calcul des coefficients d'Haralick 3D et extraction de l'ensemble d'apprentissage.

4.6. SEGMENTATION D'IMAGES ÉCHOGRAPHIQUES 3D PAR CONTOUR ACTIF SUPERVISÉ

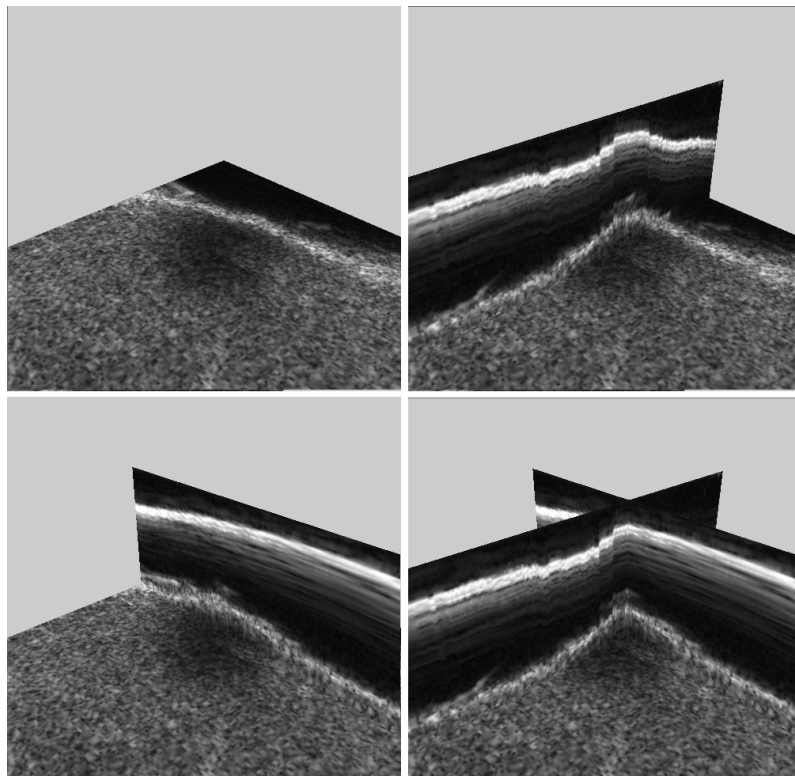


FIG. 4.30 – Illustration de la première échographie 3D à traiter. Le point d'observation reste fixe et les différents axes de coupes sont affichés sur chaque image. La pathologie à segmenter est représentée par la zone sombre au centre.

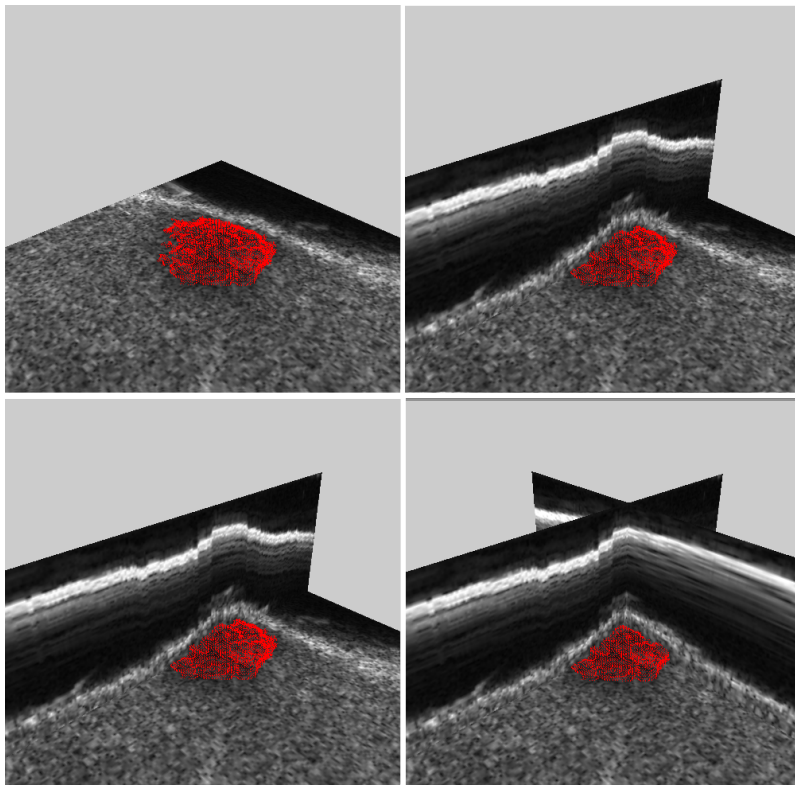


FIG. 4.31 – Segmentation réalisée sur la première échographie 3D.

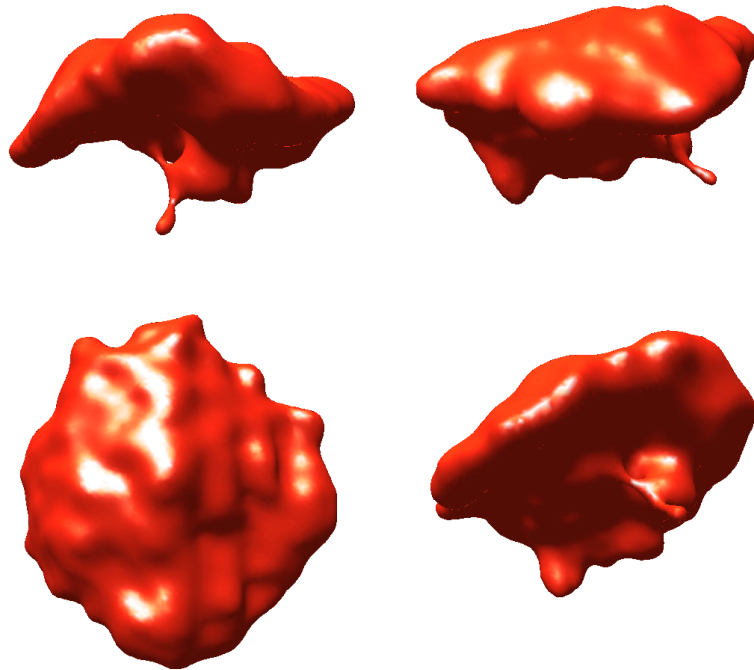


FIG. 4.32 – Maillage reconstruit pour la première échographie 3D.

4.6. SEGMENTATION D'IMAGES ÉCHOGRAPHIQUES 3D PAR CONTOUR ACTIF SUPERVISÉ

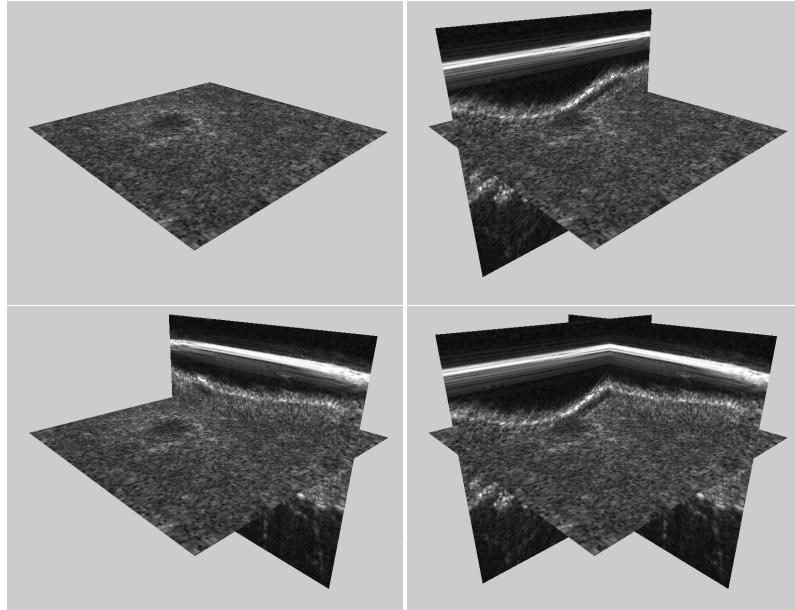


FIG. 4.33 – Deuxième échographie 3D.

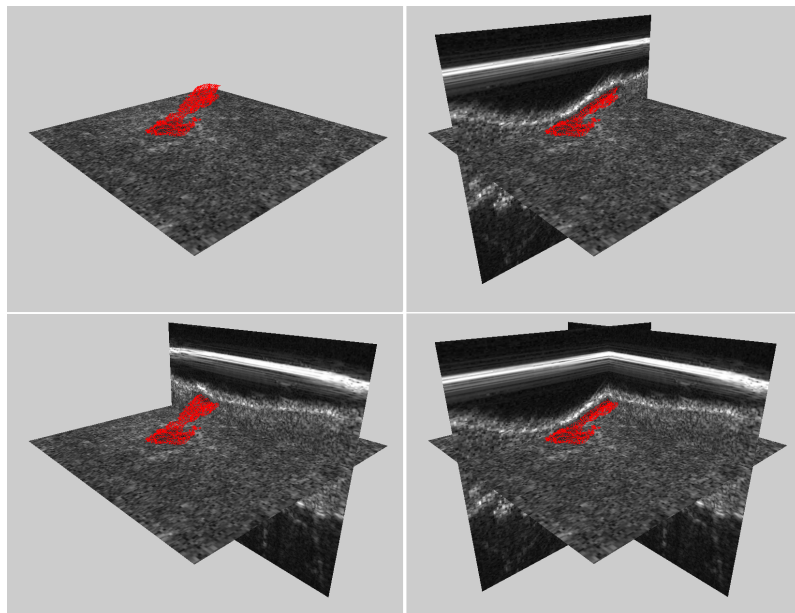


FIG. 4.34 – Segmentation réalisée sur la deuxième échographie 3D.

4.6. SEGMENTATION D'IMAGES ÉCHOGRAPHIQUES 3D PAR CONTOUR ACTIF SUPERVISÉ

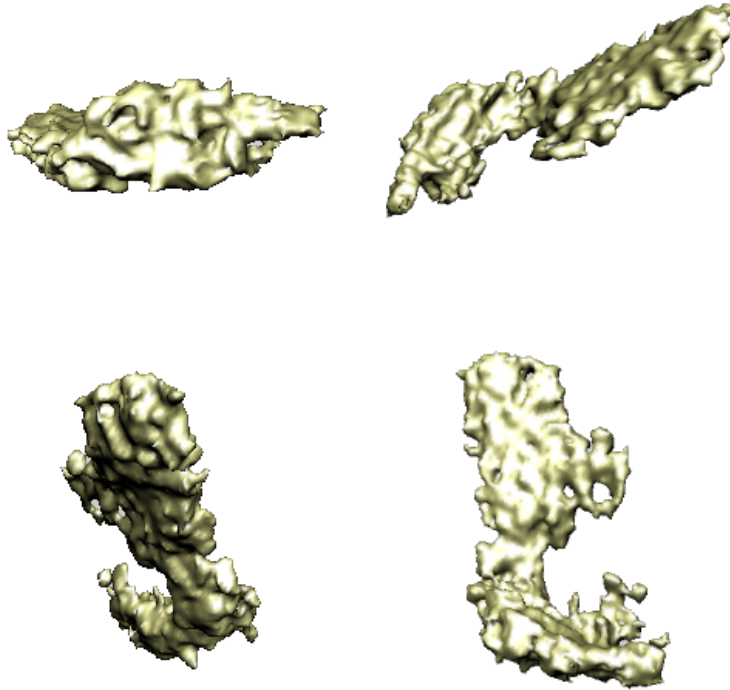


FIG. 4.35 – Maillage reconstruit pour la deuxième échographie 3D.

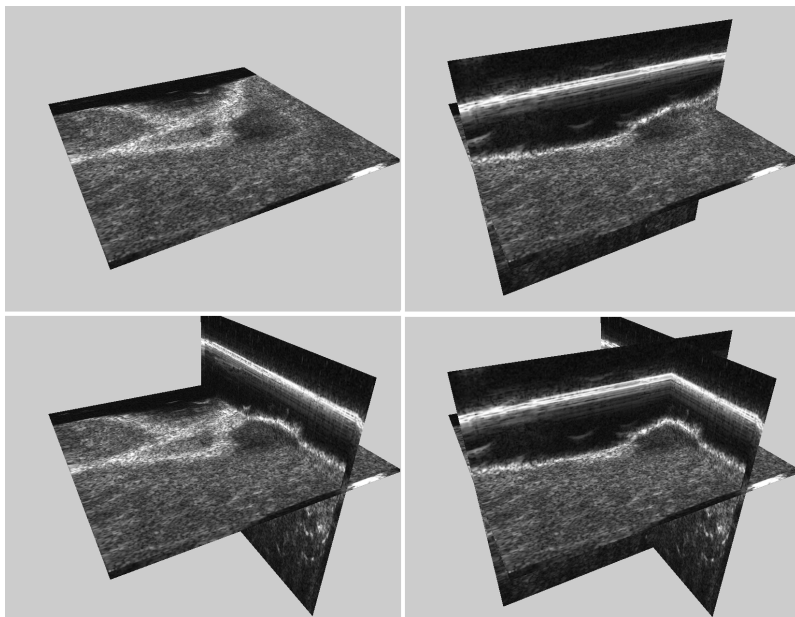


FIG. 4.36 – Troisième échographie 3D.

4.6. SEGMENTATION D'IMAGES ÉCHOGRAPHIQUES 3D PAR CONTOUR ACTIF SUPERVISÉ

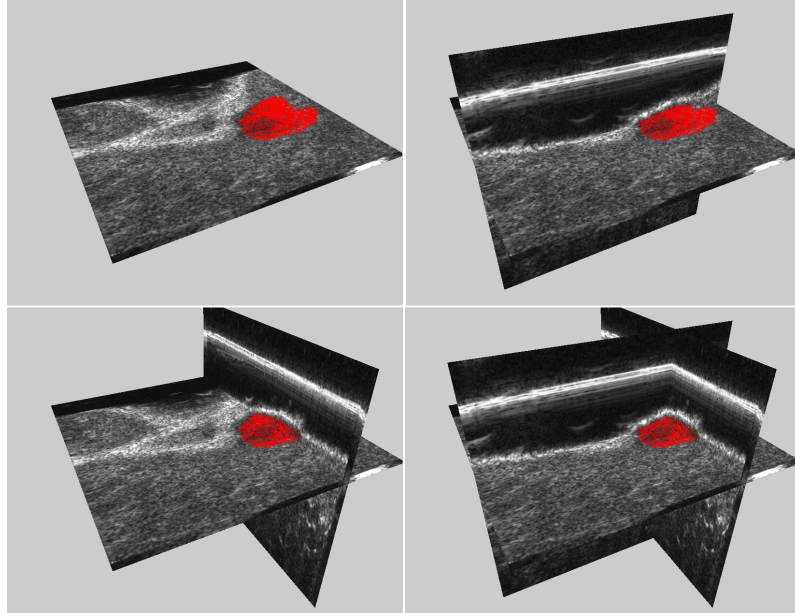


FIG. 4.37 – Segmentation réalisée sur la troisième échographie 3D.

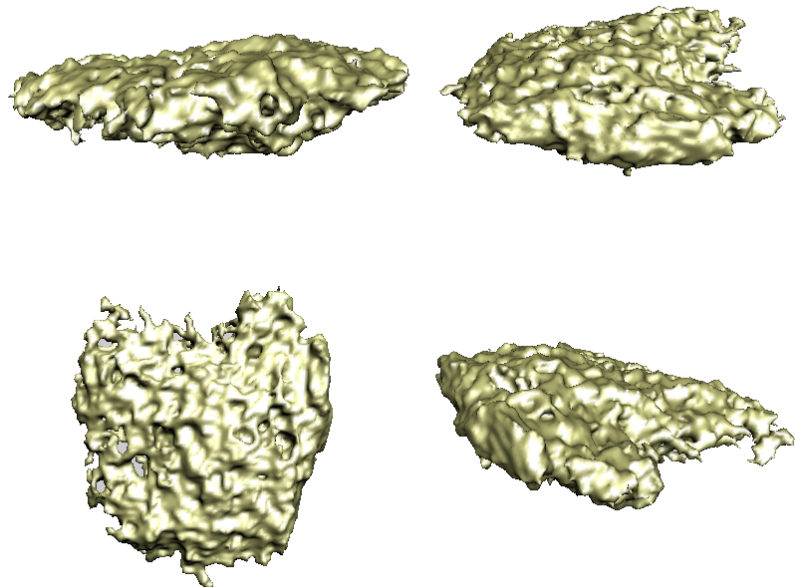


FIG. 4.38 – Maillage reconstruit pour la troisième échographie 3D.

4.6. SEGMENTATION D'IMAGES ÉCHOGRAPHIQUES 3D PAR CONTOUR ACTIF SUPERVISÉ

Conclusion

Ce mémoire de thèse a présenté plusieurs approches développées afin d'améliorer les contours actifs représentés aussi bien de manière paramétrique qu'avec les ensembles de niveaux.

Dans le but d'accélérer les contours actifs paramétriques se déformant par évolution gloutonne, trois approches ont été proposées. Celles-ci sont basées sur une gestion dynamique de la grille de voisinage. L'EVAL (Evolution par Voisinage à Anticipation Linéaire) utilise l'information de la direction suivie par chaque point de contrôle durant l'itération précédente pour déterminer un voisinage linéaire. Celui-ci est alors utilisé en alternance avec le voisinage classique afin de forcer le modèle à explorer la direction la plus intéressante. L'EVADEC (Evolution par Voisinage à Anticipation par Décalage) réalise un décalage de la grille de voisinage de chaque point de contrôle, dans le but d'orienter la recherche de l'algorithme glouton dans la direction suivie à l'itération précédente. Enfin, l'EVADEF (Evolution par Voisinage à Anticipation par Déformation) modifie la forme de la grille de voisinage en analysant la fiabilité des décisions prises par l'évolution gloutonne, et augmente l'espace de recherche de chaque point de contrôle en ajoutant les pixels considérés intéressants.

Ces trois méthodes d'accélération ont été appliquées avec succès à la segmentation d'images 2D et 3D. Lors de tests sur les images 2D, l'EVADEC est apparue comme l'approche la plus rapide, les résultats obtenus par l'EVAL se sont avérés très bons (bien que légèrement inférieurs à ceux de l'EVADEC). En revanche, l'EVADEF n'a pas obtenu d'aussi bons résultats en raison d'une forte dépendance au nombre d'itérations réalisées par l'évolution gloutonne. Pour cette raison, cette approche n'a pas été étendue à la 3D.

Dans le cadre de la segmentation 3D, L'EVAL est apparue comme la méthode la plus efficace, juste devant l'EVADEC. La raison de ce changement est que le passage à la 3D n'augmente pas la complexité de l'EVAL (une ligne conserve le même nombre d'éléments malgré le passage en 3D), alors qu'il accroît celle de l'EVADEC (la grille de voisinage devient cubique, ce qui augmente le nombre de voisins possibles et rend les calculs plus lourds). Néanmoins, les deux approches améliorent la rapidité de l'évolution gloutonne et se sont avérées meilleures que l'implémentation rapide des ensembles de niveaux développée dans [Shi et Karl, 2005a]. De plus, les mesures de précision réalisées indiquent clairement que l'accélération engendrée par les méthodes proposées n'affecte que très faiblement la qualité de la segmentation obtenue.

Ainsi, dans le cadre de problématiques de segmentation 2D avec contrainte de temps

forte (le suivi d'objet dans des vidéos en temps réel par exemple), l'EVA-Dec représentera une approche appropriée. En revanche, pour des applications similaires 3D, l'EVAL lui sera préférée.

Malgré des résultats en dessous des deux autres approches, le principe de l'EVA-Def nous paraît toujours intéressant. Nous réfléchissons donc actuellement à un nouveau procédé permettant de modéliser la fiabilité des décisions prises par l'évolution gloutonne.

Nous étudions également la possibilité d'utiliser les maillages obtenus par nos approches comme bases pour l'utilisation de méthodes de segmentation en parties significatives [Attene *et al.*, 2006, Shamir, 2006, Delest *et al.*, 2007, Tierny *et al.*, 2007].

Afin d'améliorer la précision des contours actifs représentés par ensembles de niveaux sur les images texturées, deux modèles supervisés basés sur les coefficients d'Haralick ont été développés. Afin de garantir une indépendance par rapport aux types de textures traités, pour chacune des deux approches, une segmentation experte est utilisée afin de créer un ensemble d'apprentissage composés de pixels appartenant à l'objet recherché et à la région extérieure proche de sa frontière. Le CA-PLS (Contour Actif guidé par Programmation Linéaire Simple), inspiré du contour actif basé région développé dans [Chan et Vese, 2001], utilise le principe de programmation linéaire afin de déterminer les poids optimaux des coefficients d'Haralick, proportionnels à leur capacité à séparer correctement les individus de l'ensemble d'apprentissage. La deuxième approche insère directement un classificateur binaire supervisé dans l'équation d'évolution du contour actif en utilisant l'implémentation développée dans [Shi et Karl, 2005a]. Les différents paramètres du classificateur sont déterminés par l'ensemble d'apprentissage. Trois modèles, basés sur des classificateurs différents, ont été développés : le CA-PPV (Contour Actif guidé par les k-Plus-Proches-Voisins), le CA-SVM (Contour Actif guidé par Support Vector Machines) et le CA-RNA (Contour Actif guidé par Réseau de Neurons Artificiels).

Ces modèles ont été appliqués à la segmentation d'images texturées 2D et 3D et comparés au modèle de Chan et Vese. Afin de confirmer leur indépendance par rapport aux types de textures segmentés, plusieurs problématiques 2D ont été traitées : la segmentation d'images synthétiques, la détection de pathologie dans des images échographiques et la détection de zones urbaines dans des images satellites. Le CA-PPV a éprouvé certaines difficultés pour généraliser les données de l'ensemble d'apprentissage. En outre, sa vitesse de segmentation s'est avérée être la plus faible. Le CA-PLS et le CA-SVM sont apparus comme des modèles plus précis, avec un avantage pour le deuxième. Toutefois, le CA-SVM souffre d'une certaine irrégularité et d'une vitesse de segmentation élevée et le CA-PLS a montré des capacités certaines à réaliser une sélection de caractéristiques. Le CA-RNA s'est révélé comme le modèle le plus précis. De plus, son faible temps d'apprentissage et sa rapidité de segmentation lui permette d'être considéré comme le meilleur modèle. Celui-ci a d'ailleurs été adapté avec succès à la reconstruction de pathologie dans des échographies 3D.

Nous explorons actuellement plusieurs pistes pour permettre aux modèles développés d'être plus efficaces. Ainsi, un modèle multi-classificateur est à l'étude. Celui-ci intégrera plusieurs classificateurs et guidera son évolution par le principe du vote majoritaire. Nos modèles n'étant pas liés à un type de descripteur de textures particulier, l'intégration

CONCLUSION

de nouvelles caractéristiques (coefficients d'ondelettes, réponses de filtres de Gabor ...) est également prévue afin d'augmenter encore l'indépendance par rapport aux types de textures traités. Enfin, un modèle hybride, bénéficiant de la capacité à réaliser une sélection de caractéristiques du CA-PLS et de la précision du CA-RNA, est également à l'étude. La programmation linéaire sera utilisée uniquement en tant que filtre sur les caractéristiques, en amont du réseau de neurones.

En termes d'application, un logiciel de traitement d'images échographiques 3D est actuellement en cours de développement en partenariat avec l'équipe 5 de l'unité Inserm U930 "Imagerie et Cerveau". Le but de ce logiciel est de fournir des reconstructions 3D de pathologies observables en échographie. Les deux champs d'application principaux sont la dermatologie, avec l'analyse d'échographies similaires à celles présentées au 4.6, et la chirurgie dentaire. Pour la deuxième application, une sonde miniature est en cours de développement. Grâce à elle, le chirurgien dentiste pourra acquérir par lui-même, et de manière immédiate, des images des pathologies de ses patients sans avoir à utiliser des techniques de radiographies, plus longues, coûteuses, et invasives. Nos méthodes de reconstruction 3D permettront alors d'obtenir rapidement une visualisation 3D et plusieurs mesures de la pathologie.

CONCLUSION

Annexe A

Principaux coefficients d'Haralick

Les 12 coefficients d'Haralick utilisés par les contours actifs supervisés pour caractériser les pixels sont détaillés dans cette annexe.

Notations :

N_g : nombre de niveaux de gris pris en compte par l'analyse (*i.e.* : taille de la matrice de cooccurrence).

μ : valeur moyenne des éléments de la matrice de cooccurrence P_U .

σ : écart type de P_U .

$$P_x(i) = \sum_{j=0}^{N_g-1} P_U(i, j).$$

$$P_y(j) = \sum_{i=0}^{N_g-1} P_U(i, j).$$

La matrice de cooccurrence utilisée étant symétrique, nous avons $P_x(i) = P_y(i)$.

$$P_{i+j}(k) = \sum_{i+j=k} \sum P_U(i, j), \quad k \in [0, 2N_g - 1] \text{ (somme sur les diagonales secondaires).}$$

$$P_{i-j}(k) = \sum_{i-j=k} \sum P_U(i, j), \quad k \in [0, N_g - 1] \text{ (somme sur les diagonales principales).}$$

$$HXY1 = - \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (P_U(i, j)) \text{Log}(P_x(i)P_y(j)).$$

De nombreuses traductions des coefficients d'Haralick existent dans la littérature. En conséquence nous conserverons dans cette annexe les appellations d'origine.

<i>Inverse Different Moment :</i>	$k_1 = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{1}{1 + (i-j)^2} (P_U(i, j))$
<i>Contrast :</i>	$k_2 = \sum_{k=0}^{N_g-1} P_{i-j}(k)$
<i>Angular Second Moment :</i>	$k_3 = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} P_U(i, j)^2$
<i>Correlation :</i>	$k_4 = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{(i - \mu)(j - \mu)P_U(i, j)}{\sigma^2}$
<i>Entropy :</i>	$k_5 = - \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (P_U(i, j)) \text{Log}(P_U(i, j))$
<i>Sum of squares :</i>	$k_6 = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (1 - \mu)^2 (P_U(i, j))$
<i>Cluster Shade :</i>	$k_7 = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i + j - 2\mu)^3 (P_U(i, j))$
<i>Cluster Prominence :</i>	$k_8 = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i + j - 2\mu)^4 (P_U(i, j))$
<i>Sum entropy :</i>	$k_9 = - \sum_{i=0}^{2N_g} (P_{x+y}(i)) \text{Log}(P_{x+y}(i))$
<i>Sum variance :</i>	$k_{10} = \sum_{i=0}^{2N_g} (i - V_9)^2 (P_{x+y}(i))$
<i>Sum average :</i>	$k_{11} = \sum_{i=0}^{2N_g} i (P_{x+y}(i))$
<i>Information measures of correlation :</i>	$k_{12} = \frac{k_4 - HXY1}{\max(HX, HY)}$

Annexe B

De la fonctionnelle d'énergie d'un contour actif à son équation d'évolution

Le modèle proposé au 3.2.1.1 se déforme en minimisant l'énergie exprimée à l'aide des ensembles de niveaux suivante :

$$\begin{aligned}
 \mathcal{F}_\epsilon(\phi, \overline{k_j^{in}}, \overline{k_j^{out}}) &= \mu \int_{\Omega} \delta_\epsilon(\phi(\mathbf{x})) |\nabla \phi(\mathbf{x})| d\mathbf{x} \\
 &+ \int_{\Omega} \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{in}} \right|^2 H_\epsilon(\phi(\mathbf{x})) d\mathbf{x} \\
 &+ \int_{\Omega} \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{out}} \right|^2 H_\epsilon(\phi(\mathbf{x})) d\mathbf{x} .
 \end{aligned} \tag{B.1}$$

Nous cherchons alors à déterminer l'équation d'évolution de la courbe $C(s, t)$ permettant d'obtenir un minimum local de $F_\epsilon(\phi, \overline{k_j^{in}}, \overline{k_j^{out}})$. Le modèle étant représenté de manière implicite, c'est en réalité l'EDP contrôlant l'évolution de ϕ relativement au temps t donnée par $\frac{\partial \phi}{\partial t}$ qui va être recherchée.

Comme nous avons $F = F_1 + F_2 + F_3 = \int_{\Omega} f_1(\mathbf{x}, \phi(\mathbf{x}), \phi_{\mathbf{x}}) + \int_{\Omega} f_2(\mathbf{x}, \phi(\mathbf{x}), \phi_{\mathbf{x}}) + \int_{\Omega} f_3(\mathbf{x}, \phi(\mathbf{x}), \phi_{\mathbf{x}})$, avec $\phi_{\mathbf{x}} = \phi_x + \phi_y = \frac{\partial \phi}{\partial x} + \frac{\partial \phi}{\partial y}$, alors en respectant les équations d'Euler-Lagrange, nous obtenons :

$$\begin{aligned}
 \frac{\delta F}{\delta \phi} &= \frac{\delta F_1}{\delta \phi} + \frac{\delta F_2}{\delta \phi} + \frac{\delta F_3}{\delta \phi} \\
 &= \frac{\partial f_1}{\partial \phi} - \frac{d}{dx} \frac{\partial f_1}{\partial \phi_x} - \frac{d}{dy} \frac{\partial f_1}{\partial \phi_y} \\
 &+ \frac{\partial f_2}{\partial \phi} - \frac{d}{dx} \frac{\partial f_2}{\partial \phi_x} - \frac{d}{dy} \frac{\partial f_2}{\partial \phi_y} \\
 &+ \frac{\partial f_3}{\partial \phi} - \frac{d}{dx} \frac{\partial f_3}{\partial \phi_x} - \frac{d}{dy} \frac{\partial f_3}{\partial \phi_y} .
 \end{aligned} \tag{B.2}$$

Chaque terme de l'équation (B.2) peut être développé indépendamment. Ainsi le premier terme devient :

$$\frac{\delta F_1}{\delta \phi} = \frac{\partial \delta_\epsilon(\phi)}{\partial \phi} |\nabla \phi| + \delta_\epsilon(\phi) \frac{\partial |\nabla \phi|}{\partial \phi} - \delta_\epsilon(\phi) \frac{d}{dx} \frac{|\nabla \phi|}{d\phi_x} - \delta_\epsilon(\phi) \frac{d}{dy} \frac{|\nabla \phi|}{d\phi_y} . \quad (\text{B.3})$$

Comme seule l'information présente sur le niveau zéro de ϕ est prise en compte, $\frac{\partial \delta_\epsilon(\phi)}{\partial \phi} |\nabla \phi|$ devient négligeable. $|\nabla \phi| = \sqrt{\phi_x^2 + \phi_y^2}$ et est alors indépendante de ϕ .

Finalement le premier terme devient :

$$\frac{\delta F_1}{\delta \phi} = -\delta_\epsilon(\phi) \frac{d}{dx} \frac{|\nabla \phi|}{d\phi_x} - \delta_\epsilon(\phi) \frac{d}{dy} \frac{|\nabla \phi|}{d\phi_y} \quad (\text{B.4})$$

$$= -\delta_\epsilon(\phi) \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) . \quad (\text{B.5})$$

En utilisant les équations d'Euler-Lagrange pour le second terme nous obtenons :

$$\frac{\delta F_2}{\delta \phi} = \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{in}} \right|^2 \left[\frac{\partial H_\epsilon(\phi(\mathbf{x}))}{\partial \phi} - \frac{d}{dx} \frac{H_\epsilon(\phi(\mathbf{x}))}{d\phi_x} - \frac{d}{dy} \frac{H_\epsilon(\phi(\mathbf{x}))}{d\phi_y} \right] . \quad (\text{B.6})$$

Comme $H_\epsilon(\phi(\mathbf{x}))$ est indépendant de ϕ_x et ϕ_y nous avons :

$$\begin{aligned} \frac{\delta F_2}{\delta \phi} &= \sum_{j=1}^k w_m \left| k_j(\mathbf{x}) - \overline{k_j^{in}} \right|^2 \frac{\partial H_\epsilon(\phi(\mathbf{x}))}{\partial \phi} \\ &= \delta_\epsilon(\phi) \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{in}} \right|^2 . \end{aligned} \quad (\text{B.7})$$

L'expression du troisième terme est obtenue en suivant la même approche :

$$\frac{\delta F_3}{\delta \phi} = -\delta_\epsilon(\phi) \sum_{j=1}^m w_i \left| k_j(\mathbf{x}) - \overline{k_j^{out}} \right|^2 . \quad (\text{B.8})$$

Finalement, $\frac{\delta \mathcal{F}_\epsilon}{\delta \phi}$ peut être exprimé comme :

$$\frac{\delta \mathcal{F}_\epsilon}{\delta \phi} = \delta_\epsilon(\phi) \left[-\operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{in}} \right|^2 - \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{out}} \right|^2 \right] . \quad (\text{B.9})$$

Une fois l'énergie du modèle minimum, la courbe doit s'arrêter. La dérivée fonctionnelle de $\mathcal{F}_\epsilon(\phi, \overline{k_j^{in}}, \overline{k_j^{out}})$ est alors introduite dans un schéma dynamique afin d'assurer la stabilité du modèle :

$$\frac{\delta \mathcal{F}_\epsilon(\phi, \overline{k_j^{in}}, \overline{k_j^{out}})}{\delta \phi} + \frac{\partial \phi}{\partial t} = 0 . \quad (\text{B.10})$$

Cette dernière étape permet d'obtenir l'équation d'évolution du modèle représenté à l'aide des ensembles de niveaux :

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) \left[\operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{in}} \right|^2 + \sum_{j=1}^m w_j \left| k_j(\mathbf{x}) - \overline{k_j^{out}} \right|^2 \right] . \quad (\text{B.11})$$

Annexe C

Résultats de segmentation des modèles guidés par classificateur sur les images synthétiques

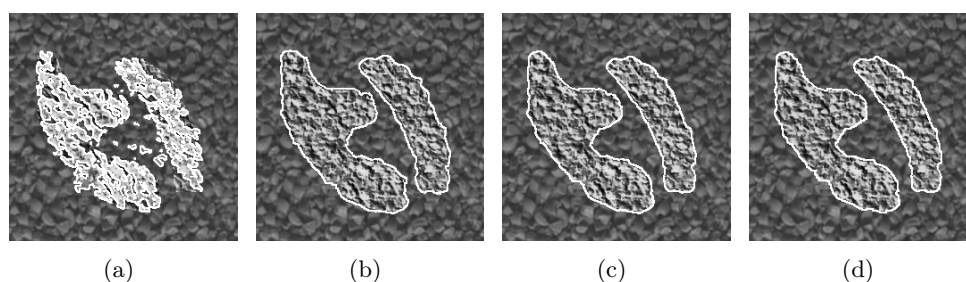


FIG. C.1 – Première série d'image, niveau de bruit 0 : a) Modèle de Chan et Vese. b) Modèle guidé par KPPV. c) Modèle guidé par SVM. d) Modèle guidé par réseau de neurones.

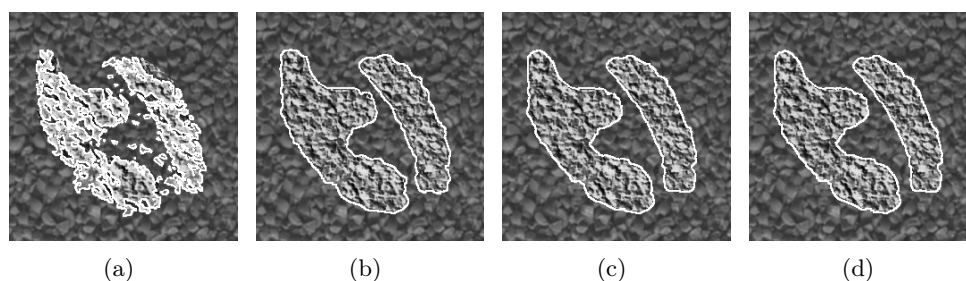


FIG. C.2 – Première série d'images, niveau de bruit 1.

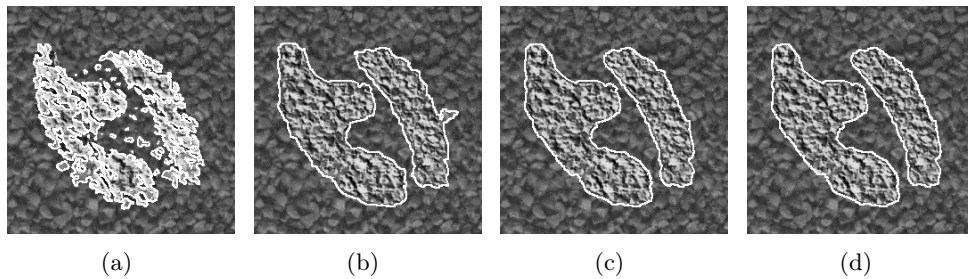


FIG. C.3 – Première série d'images, niveau de bruit 2.

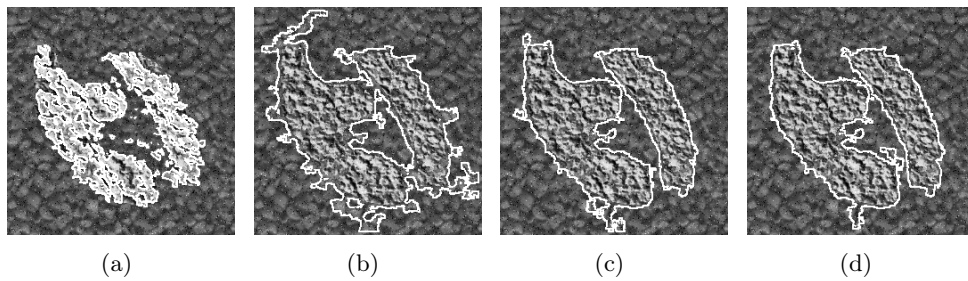


FIG. C.4 – Première série d'images, niveau de bruit 3.

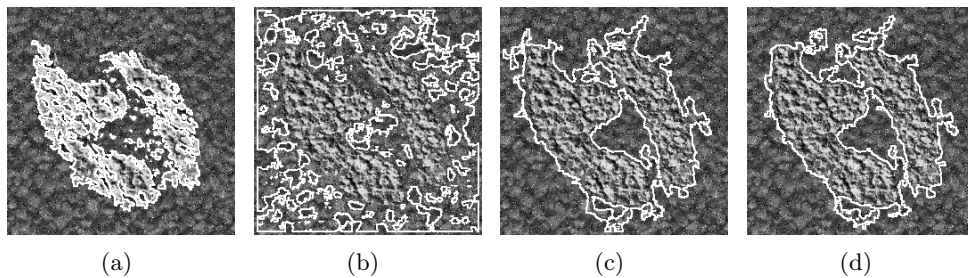


FIG. C.5 – Première série d'images, niveau de bruit 4.

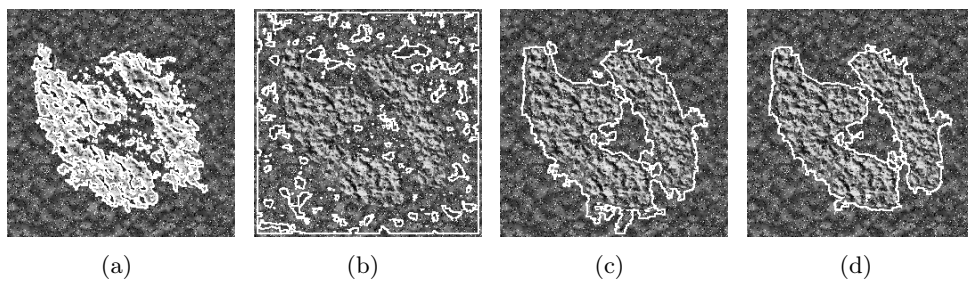


FIG. C.6 – Première série d'images, niveau de bruit 5.

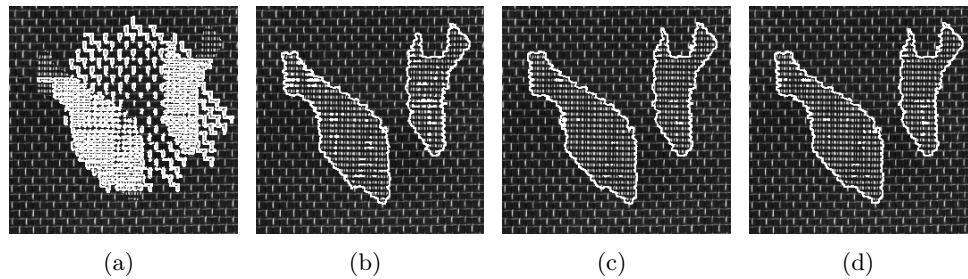


FIG. C.7 – Deuxième série d'images, niveau de bruit 0.

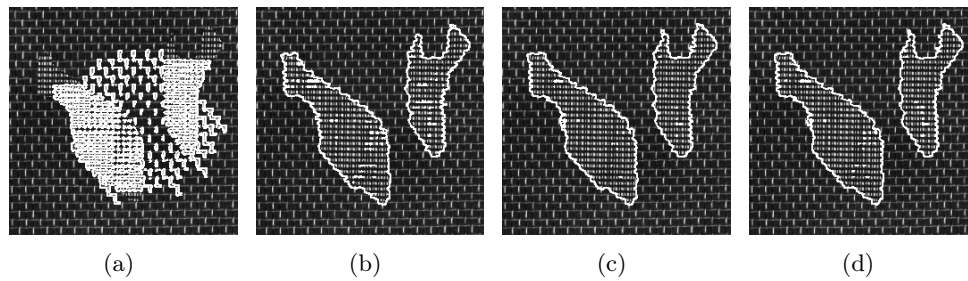


FIG. C.8 – Deuxième série d'images, niveau de bruit 1.

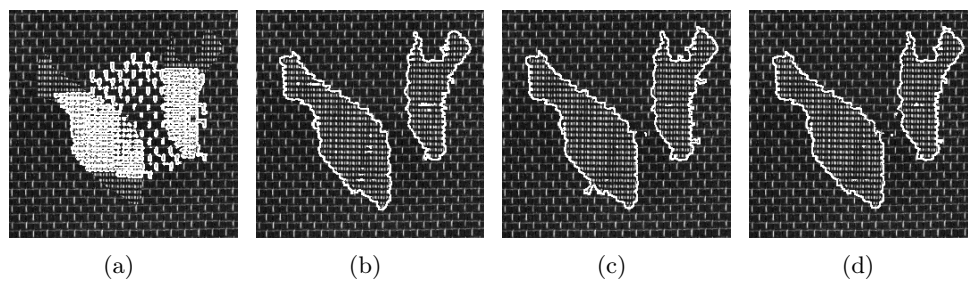


FIG. C.9 – Deuxième série d'images, niveau de bruit 2.

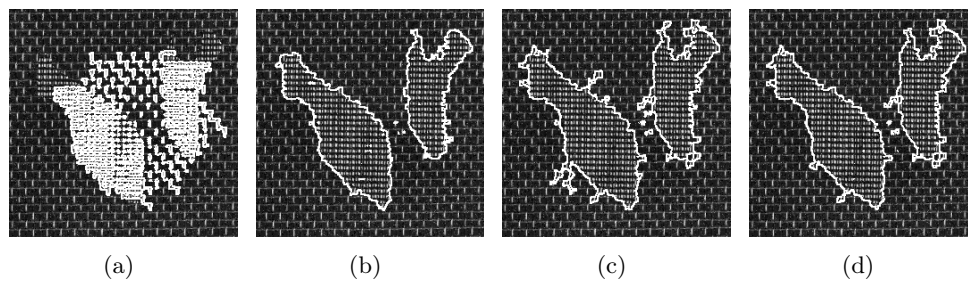


FIG. C.10 – Deuxième série d'images, niveau de bruit 3.

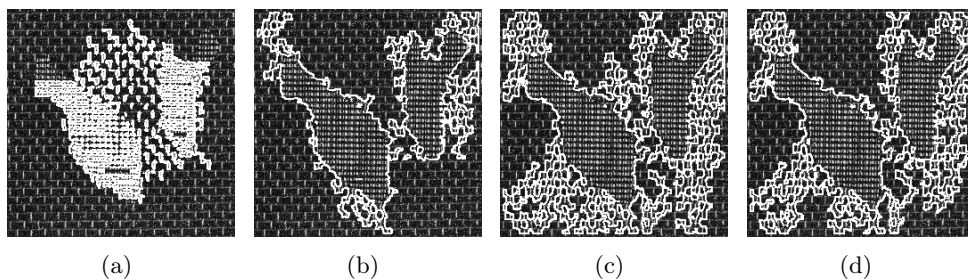


FIG. C.11 – Deuxième série d'images, niveau de bruit 4.

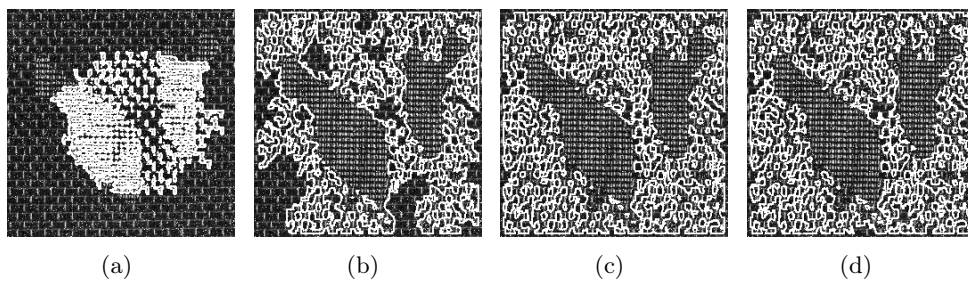


FIG. C.12 – Deuxième série d'images, niveau de bruit 5.

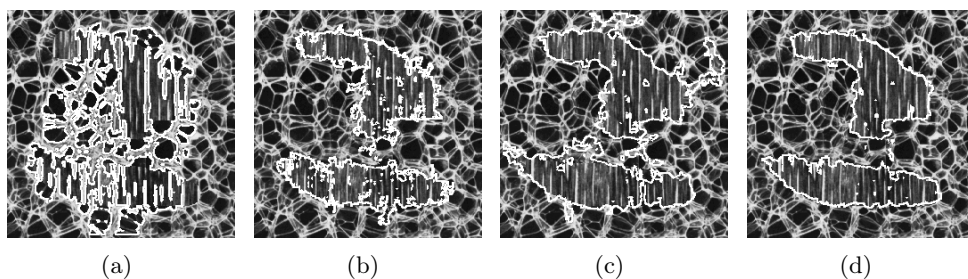


FIG. C.13 – Troisième série d'images, niveau de bruit 0.

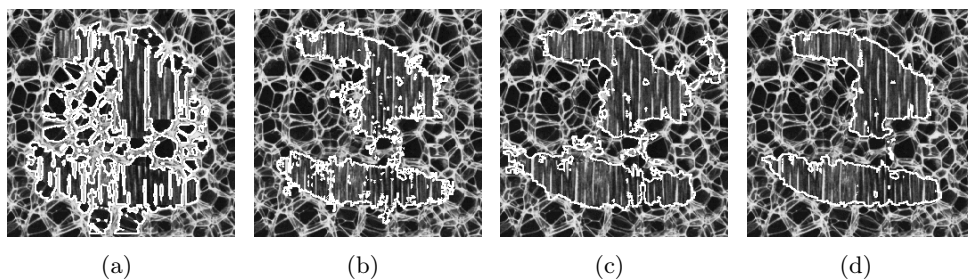


FIG. C.14 – Troisième série d'images, niveau de bruit 1.

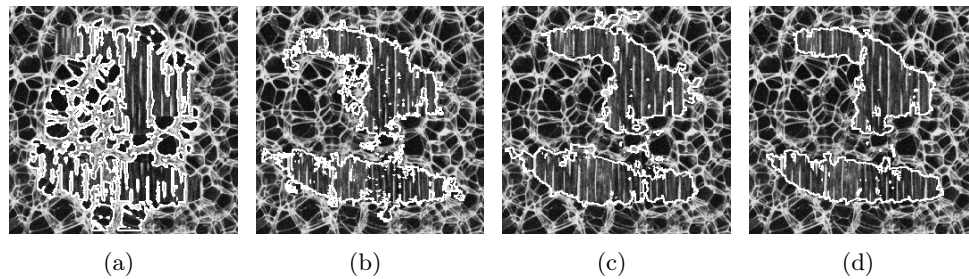


FIG. C.15 – Troisième série d'images, niveau de bruit 2.

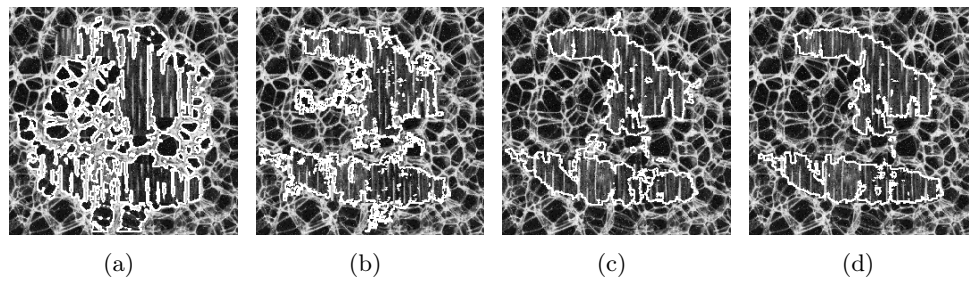


FIG. C.16 – Troisième série d'images, niveau de bruit 3.

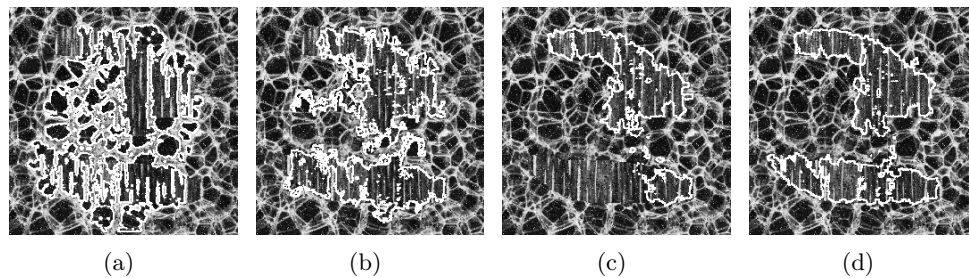


FIG. C.17 – Troisième série d'images, niveau de bruit 4.

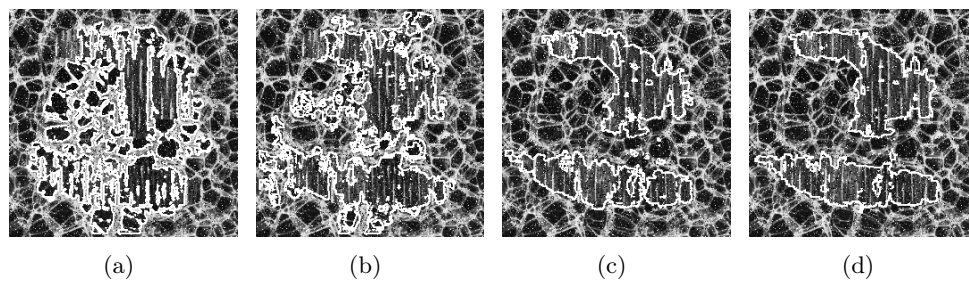


FIG. C.18 – Troisième série d'images, niveau de bruit 5.

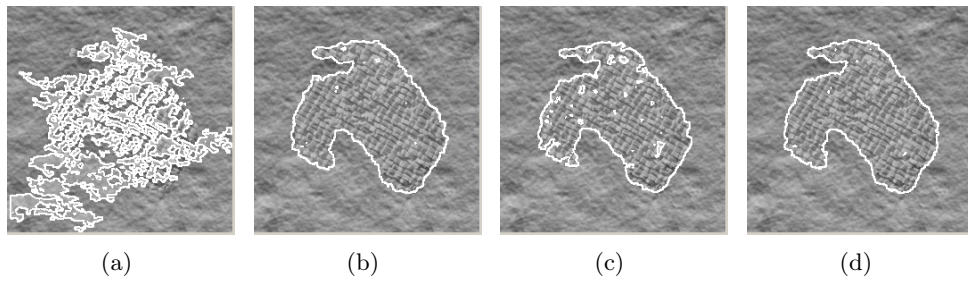


FIG. C.19 – Quatrième série d’images, niveau de bruit 0.

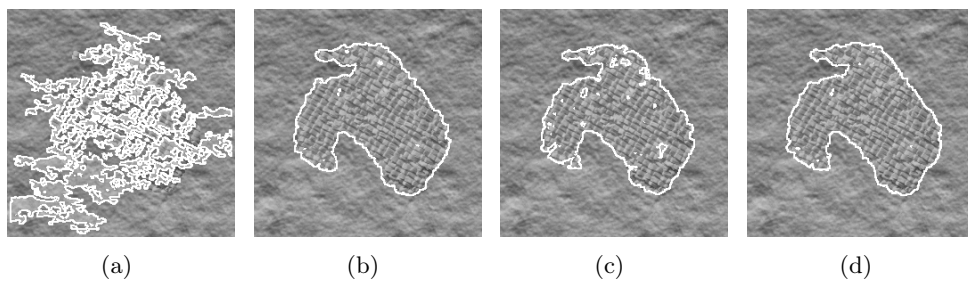


FIG. C.20 – Quatrième série d’images, niveau de bruit 1.

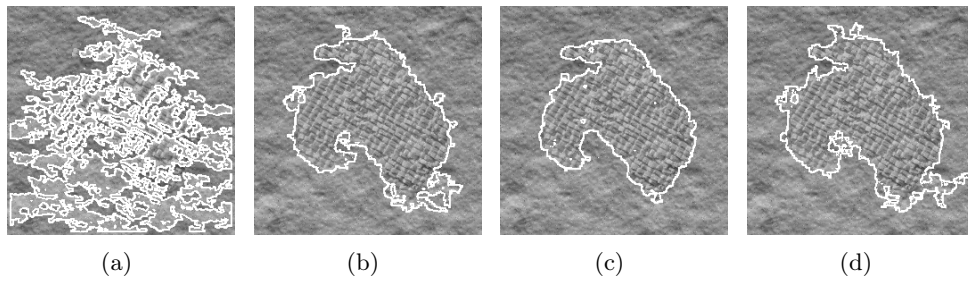


FIG. C.21 – Quatrième série d’images, niveau de bruit 2.

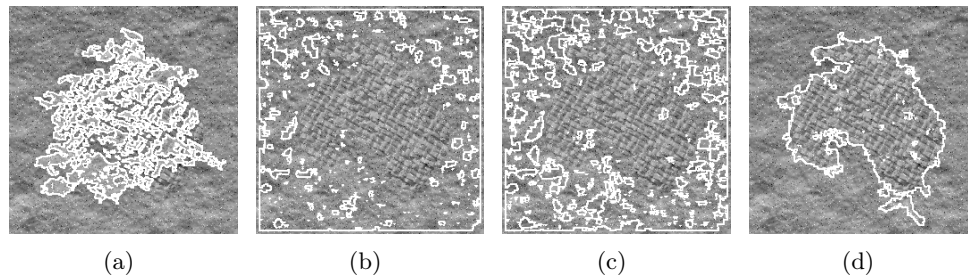


FIG. C.22 – Quatrième série d'images, niveau de bruit 3.

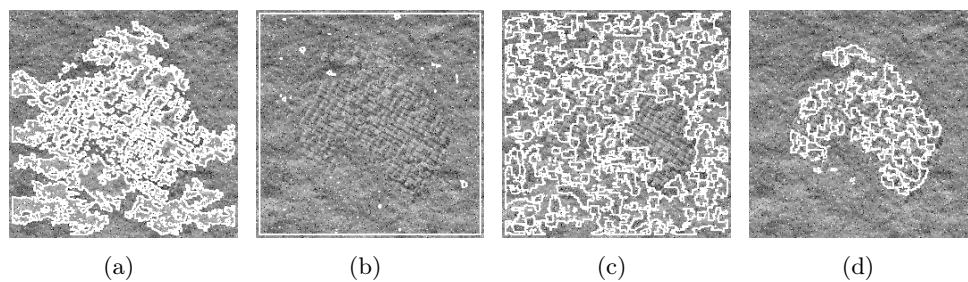


FIG. C.23 – Quatrième série d'images, niveau de bruit 4.

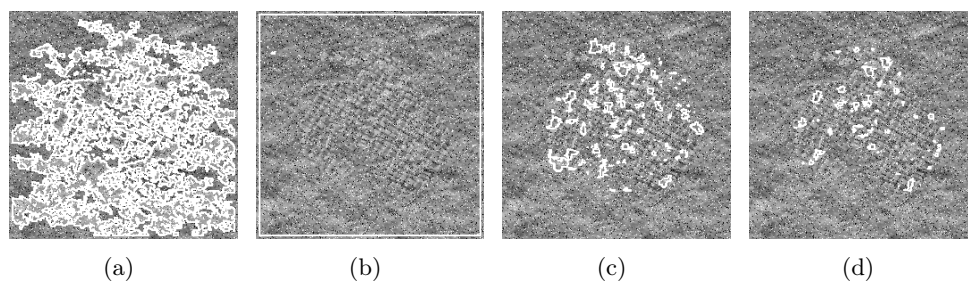


FIG. C.24 – Quatrième série d'images, niveau de bruit 5.

Annexe D

Résultats de segmentation des modèles guidés par classificateur sur les images échographiques

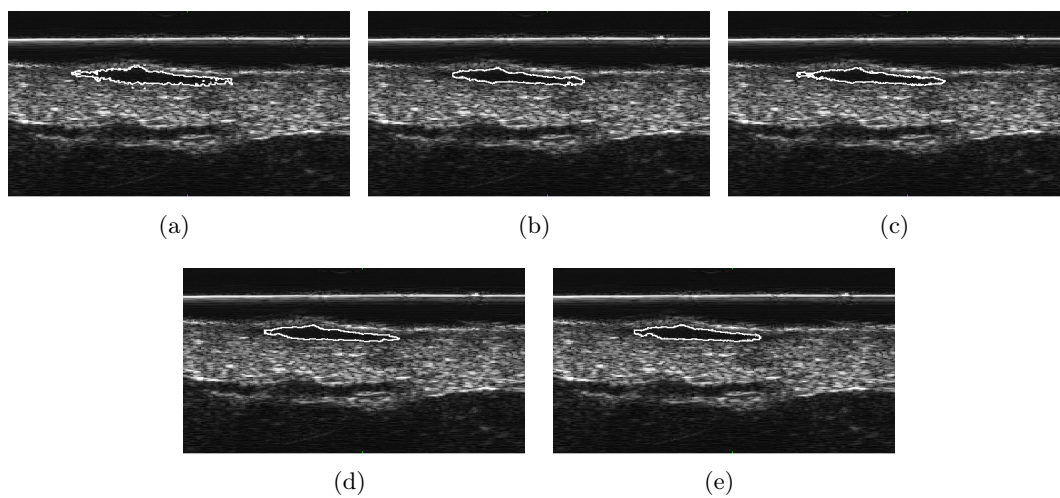


FIG. D.1 – Résultats de segmentation sur la première image échographique : a) Modèle de Chan et Vese. b) contour actif supervisé par programmation linéaire. c) Modèle guidé par KPPV. d) Modèle guidé par SVM. e) Modèle guidé par réseau de neurones.

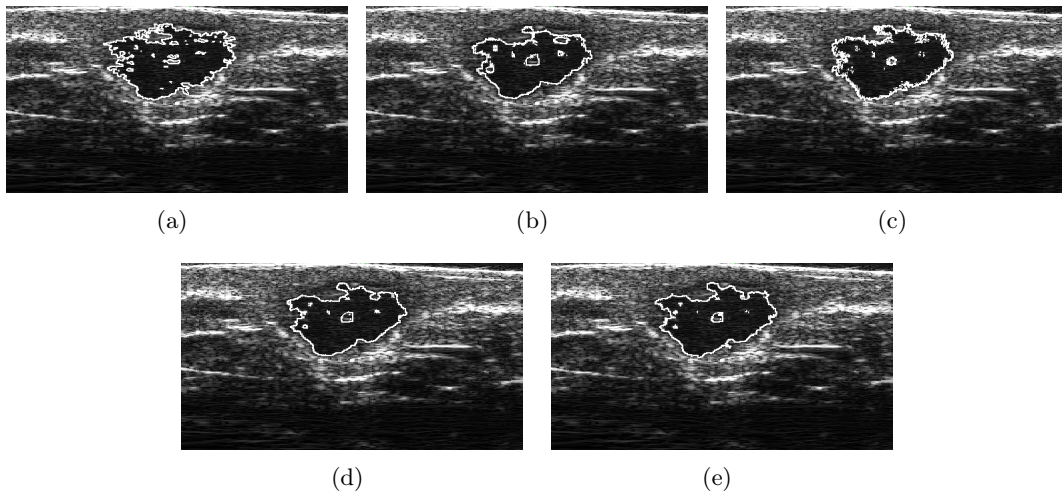


FIG. D.2 – Résultats de segmentation sur la deuxième image échographique.

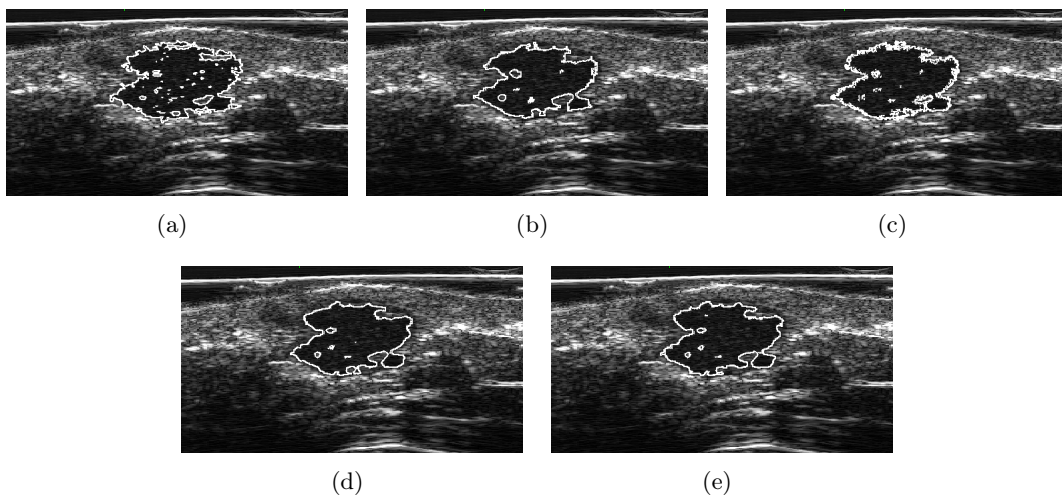


FIG. D.3 – Résultats de segmentation sur la troisième image échographique.

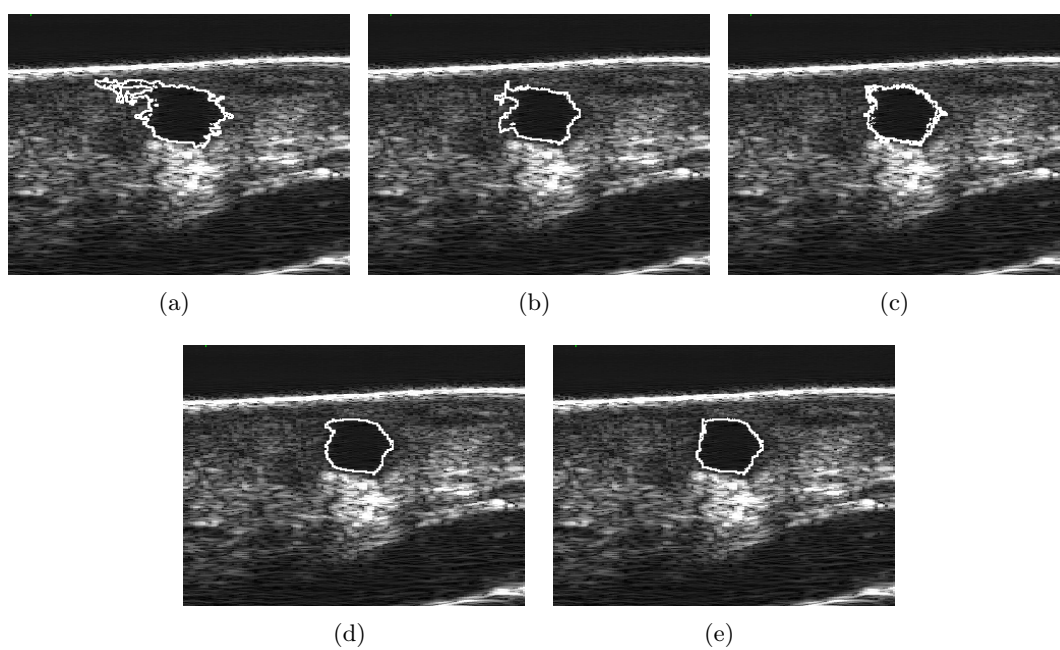


FIG. D.4 – Résultats de segmentation sur la quatrième image échographique.

Annexe E

Résultats de segmentation des modèles guidés par classificateur sur les images satellites de la base ASTER

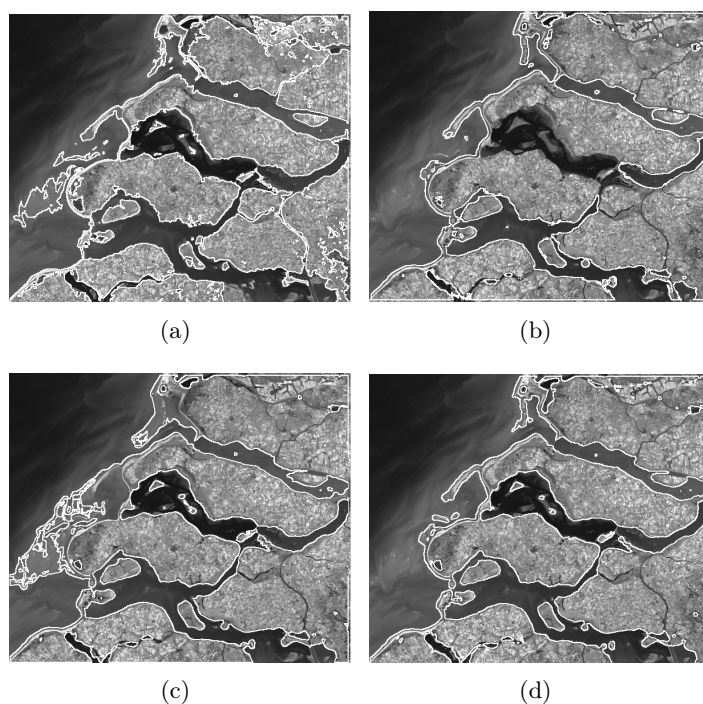


FIG. E.1 – Première image satellite : a) Modèle de Chan et Vese. b) Modèle guidé par KPPV. c) Modèle guidé par SVM. d) Modèle guidé par réseau de neurones.

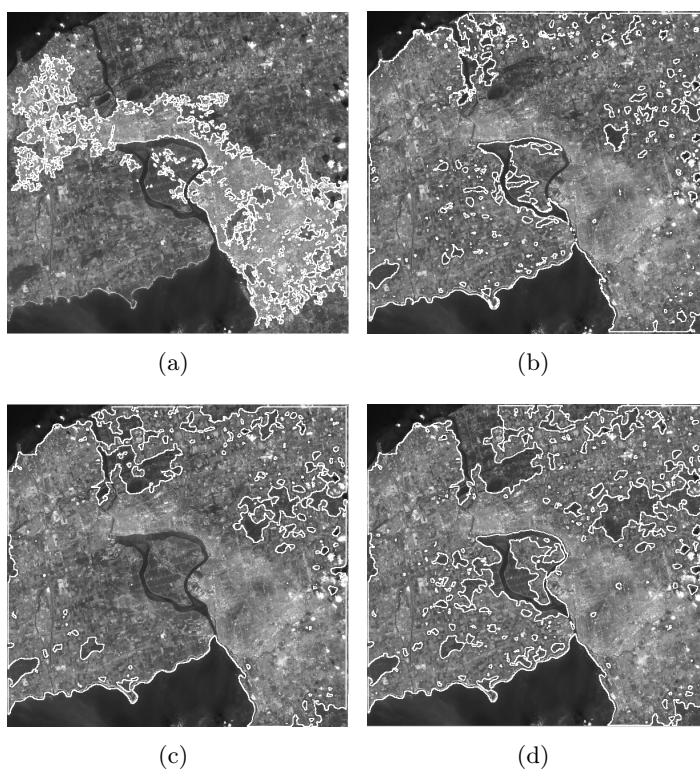


FIG. E.2 – Deuxième image satellite.

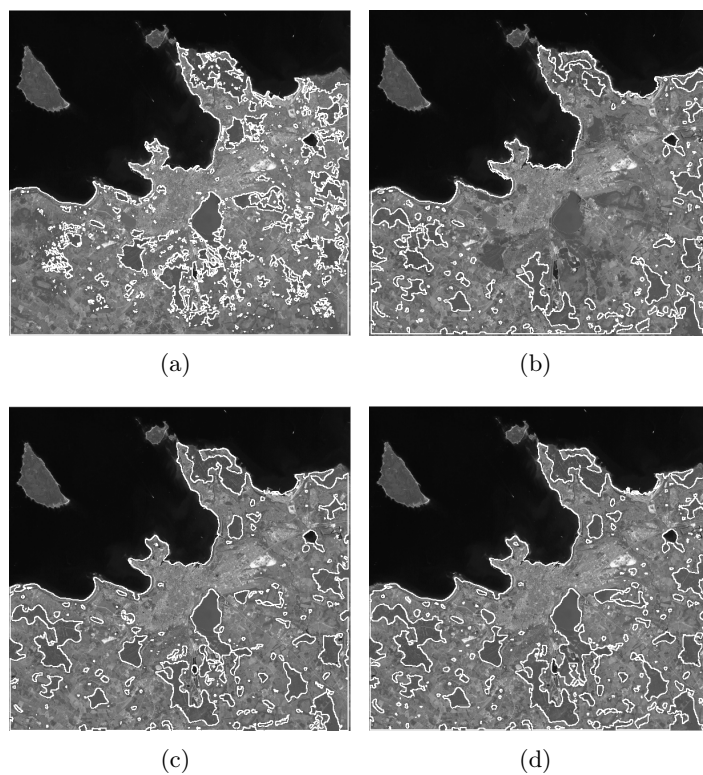


FIG. E.3 – Troisième image satellite.

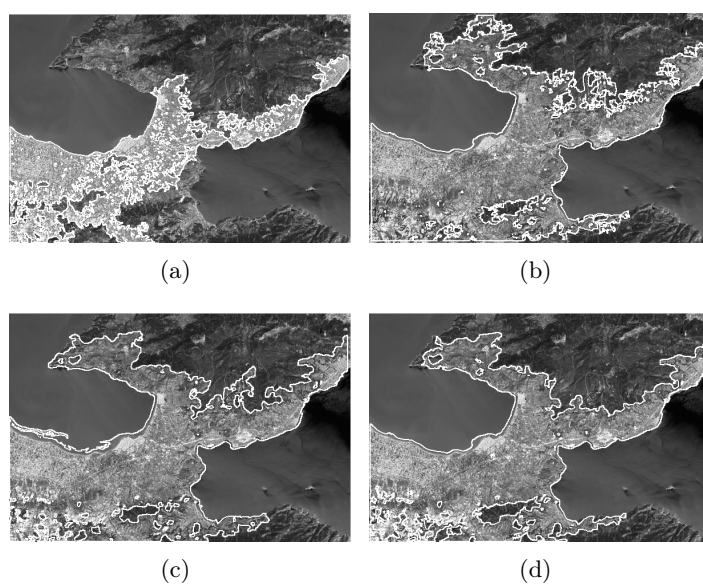


FIG. E.4 – Quatrième image satellite.

Bibliographie

- [Abdulhady *et al.*, 2002] ABDULHADY, M., ABBAS, H. et NASSAR, S. (2002). Fabric fault classification using neural trees. *In proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC02)*, volume 6, pages 114–117, Hammamet, Tunisia.
- [Adalsteinsson et Sethian, 1995] ADALSTEINSSON, D. et SETHIAN, J. A. (1995). A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2): 269–277.
- [Aizerman *et al.*, 1964] AIZERMAN, M. A., BRAVERMAN, E. A. et ROZONOER, L. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *In Automation and Remote Control*, volume 25, pages 821–837.
- [Alemán-Flores *et al.*, 2007] ALEMÁN-FLORES, M., ÁLVAREZ, L. et CASELLES, V. (2007). Texture-oriented anisotropic filtering and geodesic active contours in breast tumor ultrasound segmentation. *Journal of Mathematical Imaging and Vision*, 28(1):81–97.
- [Amini *et al.*, 1990] AMINI, A. A., WEYMOUTH, T. E. et RAIN, R. J. (1990). Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867.
- [Applegate *et al.*, 2007] APPLGATE, D. L., COOK, W., DASH, S. et ESPINOZA, D. G. (2007). Exact solutions to linear programming problems. *Operations Research Letters*, 35:693–699.
- [Arivazhagan et Ganesan, 2003] ARIVAZHAGAN, S. et GANESAN, L. (2003). Texture segmentation using wavelet transform. *Pattern Recognition Letters*, 24(16):3197–3203.
- [Arvis *et al.*, 2004] ARVIS, V., DEBAIN, C., BERDUCAT, M. et BENASSI, A. (2004). Generalization of the cooccurrence matrix for colour images : application to colour texture classification. *Image Analysis and Stereology*, 23:63–72.
- [Ashour *et al.*, 2008] ASHOUR, M. W., HUSSIN, M. F. et MAHAR, K. M. (2008). Supervised texture classification using several features extraction techniques based on ANN and SVM. *In proceedings of the IEEE/ACS International Conference on Computer Systems and Applications (AICCSA08)*, pages 567–574, Doha, Qatar.
- [Attene *et al.*, 2006] ATTENE, M., KATZ, S., MORTARA, M., PATANE, G., SPAGNUOLO, M. et TAL, A. (2006). Mesh segmentation - a comparative study. *In Proceedings of the IEEE International Conference on Shape Modeling and Applications (SMI06)*, pages 14–25, Washington, DC, USA. IEEE Computer Society.
- [Autio et Elomaa, 2003] AUTIO, T. et ELOMAA, T. (2003). Flexible view recognition for indoor navigation based on Gabor filters and support vector machines. *Pattern Recognition*, 36(12):2769–2779.

- [Azencott *et al.*, 1997] AZENCOTT, R., WANG, J.-P. et YOUNES, L. (1997). Texture classification using windowed Fourier filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):148–153.
- [Babonneau *et al.*, 2006] BABONNEAU, F., DU MERLE, O. et VIAL, J.-P. (2006). Solving large-scale linear multicommodity flow problems with an active set strategy and proximal-accpm. *Operations Research*, 54:184–197.
- [Bajcsy et Lieberman, 1976] BAJCSY, R. et LIEBERMAN, L. (1976). Texture gradient as a depth cue. *Computer Graphics and Image Processing*, 5:52–67.
- [Bashar *et al.*, 2003] BASHAR, M. K., MATSUMOTO, T. et OHNISHI, N. (2003). Wavelet transform-based locally orderless images for texture segmentation. *Pattern Recognition Letters*, 24(15):2633–2650.
- [Beaulieu et Goldberg, 1989] BEAULIEU, J.-M. et GOLDBERG, M. (1989). Hierarchy in picture segmentation : A stepwise optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):150–163.
- [Bergonnier *et al.*, 2007] BERGONNIER, S., HILD, F. et ROUX, S. (2007). Local anisotropy analysis for non-smooth images. *Pattern Recognition*, 40(2):544–556.
- [Besag, 1974] BESAG, J. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussions). *Journal of the Royal Statistical Society*, B36:192–236.
- [Besag, 1986] BESAG, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, B48:259–302.
- [Bianconi et Fernández, 2007] BIANCONI, F. et FERNÁNDEZ, A. (2007). Evaluation of the effects of Gabor filter parameters on texture classification. *Pattern Recognition*, 40(12):3325–3335.
- [Blostein et Ahuja, 1989] BLOSTEIN, D. et AHUJA, N. (1989). Shape from texture : Integrating texture-element extraction and surface estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1233–1251.
- [Bossart *et al.*, 1996] BOSSART, P.-L., DAVID, D., DINTEN, J.-M. et CHASSERY, J.-M. (1996). Détection de contours réguliers dans des images bruitées et texturées : une approche par contours actifs multiéchelle. *Traitement du Signal*, 14(2):209–225.
- [Bouman et Liu, 1991] BOUMAN, C. et LIU, B. (1991). Multiple resolutions segmentation of textured images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):99–113.
- [Brodatz, 1966] BRODATZ, P. (1966). *Textures : A Photographic Album for Artists and Designers*. New York : Dover Publications.
- [Bulpitt et Efford, 1996] BULPITT, A. J. et EFFORD, N. D. (1996). An efficient 3D deformable model with a self-optimising mesh. *Image and Vision Computing*, 14(8):573–580.
- [Cao *et al.*, 2008] CAO, G., MAO, Z., YANG, X. et XIA, D. (2008). Optical aerial image partitioning using level sets based on modified chan-tese model. *Pattern Recognition Letters*, 29(4):457–464.
- [Cardoso et Corte-Real, 2005] CARDOSO, J. S. et CORTE-REAL, L. (2005). Toward a generic evaluation of image segmentation. *IEEE Transactions on Image Processing*, 14(11):1773–1782.

- [Cariou et Chehdi, 2008] CARIOU, C. et CHEHDI, K. (2008). Unsupervised texture segmentation/classification using 2-D autoregressive modeling and the stochastic expectation-maximization algorithm. *Pattern Recognition Letters*, 29(7):905–917.
- [Caselles *et al.*, 1993] CASELLES, V., CATTE, F., COLL, T. et DIBOS, F. (1993). A geometric model for active contours. *Numerische Mathematik*, 66:1–31.
- [Caselles *et al.*, 1997] CASELLES, V., KIMMEL, R. et SAPIRO, G. (1997). Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79.
- [Chabrier *et al.*, 2004] CHABRIER, S., LAURENT, H., EMILE, B., ROSENBERGER, C. et P., M. (2004). A comparative study of supervised evaluation criteria for image segmentation. *In proceedings of the XII. European Signal Processing Conference (EUSIPCO 04)*, pages 1143–1146, Vienna, Austria.
- [Chan *et al.*, 2000] CHAN, T., SANDBERG, B. et VESE, L. (2000). Active contours without edges for vector valued images. *Journal of Visual Communication and Image Representation*, 11(2):130–141.
- [Chan et Vese, 2001] CHAN, T. et VESE, L. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277.
- [Chassery et Melkemi, 1991] CHASSERY, J. M. et MELKEMI, M. (1991). Diagramme de voronoï appliqué à la segmentation d’images et à la détection d’événements en imagerie multi-sources. *Traitement du Signal*, 8(3):155–164.
- [Chen et Chen, 1999] CHEN, C.-C. et CHEN, C.-C. (1999). Filtering methods for texture discrimination. *Pattern Recognition Letters*, 20:783–790.
- [Chen *et al.*, 2008] CHEN, Q., SUN, Q.-S., HENG, P.-A. et XIA, D.-S. (2008). Parametric active contours for object tracking based on matching degree image of object contour points. *Pattern Recognition Letters*, 29(2):126–141.
- [Choi et Lam, 2008] CHOI, W. P. et LAM, K. M. (2008). An effective shape-texture weighted algorithm for multi-view face tracking in videos. *In proceedings of the Congress on Image and Signal Processing (CISP08)*, volume 4, pages 156–160, Washington, DC, USA. IEEE Computer Society.
- [Chung et Ho, 1996] CHUNG, R. et HO, K.-K. (1996). Using 2D active contour models for 3D reconstruction from serial sections. *In proceedings of the 1996 International Conference on Pattern Recognition (ICPR96) Volume I*, page 849, Washington, DC, USA. IEEE Computer Society.
- [Clark *et al.*, 1987] CLARK, M., BOVIK, A. C. et GEISLER, W. S. (1987). Texture segmentation using Gabor modulation/demodulation. *Pattern Recognition Letters*, 6(4):261–267.
- [Clarke, 1986] CLARKE, K. C. (1986). Computation of the fractal dimension of topographic surfaces using the triangular prism surface area method. *Computer Geoscience*, 12(5):713–722.
- [Clausi et Jernigan, 2000] CLAUSI, D. et JERNIGAN, H. E. (2000). Designing Gabor filters for optimal texture separability. *Pattern Recognition*, 33(11):1835–1849.
- [Clerc et Mallat, 2002] CLERC, M. et MALLAT, S. (2002). The texture gradient equation for recovering shape from texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):536–549.

- [Cohen, 1991] COHEN, L. D. (1991). On active contour models and balloons. *Computer Vision, Graphics, and Image Processing : Image Understanding*, 53(2):211–218.
- [Cohen et al., 1993] COHEN, L. D., BARDINET, E. et AYACHE, N. (1993). Surface reconstruction using contour models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147.
- [Cohen et Cohen, 1993] COHEN, L. D. et COHEN, I. (1993). Finite element methods for active contour models and balloons for 2D et 3D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147.
- [Cross et Jain, 1983] CROSS, G. C. et JAIN, A. K. (1983). Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:25–39.
- [Dantzig, 1963] DANTZIG, G. B. (1963). *Linear programming and extensions*. Princeton University Press. XVI.
- [Daugman, 1980] DAUGMAN, J. G. (1980). Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20:847–856.
- [Delest et al., 2007] DELEST, S., BONE, R. et CARDOT, H. (2007). Automatically computed markers for the 3D watershed segmentation. In *proceedings of the International Conference on Image Processing (ICIP07)*, volume 6, pages 533–536.
- [Denzler et Niemann, 1995] DENZLER, J. et NIEMANN, H. (1995). Evaluating the performance of active contours models for real-time object tracking. In *proceedings of the Asian Conference on Computer Vision*, volume 2, pages 341–345, Singapore.
- [Dimitrov et Sramek, 2004] DIMITROV, L. I. et SRAMEK, M. (2004). Using 3d-bresenham for resampling structured grids. In *proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium (3DPVT04)*, pages 926–930, Washington, DC, USA. IEEE Computer Society.
- [Doretto et al., 2003] DORETTO, G., CREMERS, D., FAVARO, P. et SOATTO, S. (2003). Dynamic texture segmentation. In *proceedings of the International Conference on Computer Vision (ICCV03)*, volume 2, pages 1236–1242, Nice, France.
- [Dubrovin et al., 1984] DUBROVIN, B. A., FOMENKO, A. T. et NOVIKOV, S. P. (1984). *Modern Geometry - Method and Application I*. Springer-Verlag, New York.
- [Dubuc et al., 1989a] DUBUC, B., QUINIOU, J.-F., ROQUES-CARMES, C., TRICOT, C. et ZUCKER, S. (1989a). Evaluating the fractal dimension of profiles. *Physical Review A*, 39(3):1500–1512.
- [Dubuc et al., 1989b] DUBUC, B., ZUCKER, S., TRICOT, C., QUINIOU, J.-F., ROQUES-CARMES, C. et WEHBI, D. (1989b). Evaluating the fractal dimension of surfaces. *Proceedings of the Royal Society London, series A*, 425(3):113–127.
- [Dubuisson et Jain, 1994] DUBUISSON, M.-P. et JAIN, A. K. (1994). A modified Hausdorff distance for object matching. In *Proceedings of 12th International Conference on Pattern Recognition (ICPR94)*, pages 566–568, Jerusalem, Israel.
- [Epstein et Gage, 1987] EPSTEIN, C. et GAGE, M. (1987). The curve shortening flow. In *Wave Motion : Theory, Modeling, and Computation, Proceedings of the Conference in honor of the 60th birthday of PeterD. Lax*, MSRI Publications, Springer-Verlag, pages 15–59.

- [Feldman *et al.*, 2005] FELDMAN, J., WAINWRIGHT, M. J. et KARGER, D. R. (2005). Using linear programming to decode binary linear codes. *IEEE Transactions on Information Theory*, 51:954–972.
- [Fjortoft *et al.*, 2003] FJORTOFT, R., DELIGNON, Y., PIECZYNSKI, W., SIGELLE, M. et TUPIN, F. (2003). Unsupervised classification of radar images using hidden Markov chains and hidden Markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, 41(3):675–686.
- [Gabor, 1946] GABOR, D. (1946). Theory of communication. *Journal of the Institute of Electrical Engineers (JIEE), London*, 93(III):429–457.
- [Garding, 1995] GARDING, J. (1995). Surface orientation and curvature from differential texture distortion. In *Proceedings of the 5th International Conference on Computer Vision (ICCV95)*, pages 733–739, Washington, DC, USA. IEEE Computer Society.
- [Geman et Geman, 1984] GEMAN, S. et GEMAN, D. (1984). Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):731–741.
- [Grossman et Morlet, 1984] GROSSMAN, A. et MORLET, J. (1984). Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM Journal of Mathematical Analysis*, 15(4):723–736.
- [Guarget-Duport *et al.*, 1996] GUARGET-DUPORT, B., GIREL, J., CHASSERY, J.-M. et PAUTOU, G. (1996). The use of multiresolution analysis and wavelets transform for merging SPOT Panchromatic and Multispectral Image Data. *Photogrammetric Engineering and Remote Sensing*, 62(9):1057–1066.
- [Gusfield, 2002] GUSFIELD, D. (2002). Partition-distance : A problem and class of perfect graphs arising in clustering. *Information Processing Letters*, 82(3):159–164.
- [Haker *et al.*, 2000] HAKER, S., ANGENENT, S., TENNENBAUM, A. et KIKINIS, R. (2000). Nondistorting flattening maps and the 3D visualisation of colon CT images. *IEEE Transactions on Medical Imaging*, 19(7):665–670.
- [Hammersley et Clifford, 1971] HAMMERSLEY, J. M. et CLIFFORD, P. (1971). *Markov field on finite graphs and lattices*. Unpublished.
- [Han *et al.*, 2007] HAN, C.-C., LEE, C.-H. et PENG, W.-L. (2007). Hand radiograph image segmentation using a coarse-to-fine strategy. *Pattern Recognition*, 40(11):2994–3004.
- [Haralick, 1979] HARALICK, R. M. (1979). Statistical and structural approaches to textures. In *Proceedings of the IEEE*, volume 67, pages 786–804.
- [Haralick *et al.*, 1973] HARALICK, R. M., SHANMUGAM, K. et DINSTEIN, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3:610–621.
- [Hartigan et Wong, 1979] HARTIGAN, J. A. et WONG, M. A. (1979). A k-means clustering algorithm. *JSTOR : Applied Statistics*, 28(1):100–108.
- [He et Wang, 1992] HE, D. C. et WANG, L. (1992). Unsupervised textural classification of images using the texture spectrum. *Pattern Recognition*, 25(3):247–255.
- [Heikkila *et al.*, 2009] HEIKKILA, M., PIETIKAINEN, M. et SCHMID, C. (2009). Description of interest regions with local binary patterns. *Pattern Recognition*, 42(3):425–436.

- [Hiremath *et al.*, 2008] HIREMATH, P. S., SHIVASHANKAR, S. et SHIVASHANKAR, S. (2008). Wavelet based cooccurrence histogram features for texture classification with an application to script identification in a document image. *Pattern Recognition Letters*, 29(9):1182–1189.
- [Hong *et al.*, 1999] HONG, H. K., MYUNG, Y. C. et CHOI, J. S. (1999). 3-D analysis of projective textures using structural approaches. *Pattern Recognition*, 32(3):357–364.
- [Houhou *et al.*, 2008] HOUHOU, N., THIRAN, J.-P. et BRESSON, X. (2008). Fast texture segmentation model based on the shape operator and active contour. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR08)*, pages 1–8, Anchorage, Alaska.
- [Hsu *et al.*, 2000] HSU, C. T., KUO, J. L. et WILSON, R. (2000). A multiresolution texture gradient method for unsupervised segmentation. *Pattern Recognition*, 33(11):1819–1833.
- [Huttenlocher *et al.*, 1993] HUTTENLOCHER, D. P., KLANDERMAN, G. A. et RUCKLIDGE, W. J. (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863.
- [Iakovidis *et al.*, 2006] IAKOVIDIS, D. K., A., S. M., A., K. S. et E., M. D. (2006). Segmentation of medical images with regional inhomogeneities. In *ICPR '06 : Proceedings of the 18th International Conference on Pattern Recognition*, pages 976–979, Washington, DC, USA. IEEE Computer Society.
- [Idrissa et Acheroy, 2002] IDRISSE, M. et ACHEROY, M. (2002). Texture classification using Gabor filters. *Pattern Recognition Letters*, 23(9):1095–1102.
- [Ising, 1925] ISING, E. (1925). Beitrag zur theorie des ferromagnetismus. *Zeitschrift fur Physik*, 31.
- [Jedynak *et al.*, 2005] JEDYNAK, B., ZHENG, H. et DAOUDI, M. (2005). Skin detection using pairwise models. *Journal of Image Vision and Computing*, 23(13):1122–1130.
- [Jehan-Besson *et al.*, 2003] JEHAN-BESSON, S., BARLAUD, M. et AUBERT, G. (2003). Dream2s : Deformable regions driven by an eulerian accurate minimization method for image and video segmentation. *International Journal of Computer Vision*, 53(1):45–70.
- [Jimenez-Munoz et Sobrino, 2007] JIMENEZ-MUNOZ, J. C. et SOBRINO, J. A. (2007). Feasibility of retrieving land-surface temperature from ASTER TIR bands using two-channel algorithms : A case study of agricultural areas. *IEEE Geoscience and Remote Sensing Letters*, 4(1):60–64.
- [Julesz, 1975] JULESZ, B. (1975). *Experiments in the visual perception of texture*, volume 232. Scientific American.
- [Kaplan et Jay Kuo, 1995] KAPLAN, L. M. et JAY KUO, C.-C. (1995). Texture roughness analysis and synthesis via extended self-similar (ESS) model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1043–1056.
- [Karmarkar, 1984] KARMARKAR, N. (1984). A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(1):373–395.
- [Karush, 1939] KARUSH, W. (1939). Minima of functions of several variables with inequalities as side conditions. Master thesis, Department of Mathematics, University of Chicago, Chicago, IL, USA.

- [Kass *et al.*, 1988] KASS, M., WITKIN, A. et TERZOPOULOS, D. (1988). Snakes : active contour models. *International Journal of Computer Vision*, 1(4):321–331.
- [Kaufman et Rousseeuw, 1990] KAUFMAN, L. et ROUSSEEUW, P. J. (1990). *Finding groups in data : an introduction to cluster analysis*. John Wiley and Sons, New York.
- [Keller *et al.*, 1989] KELLER, J. M., CHEN, S. et CROWNOVER, R. M. (1989). Texture description and segmentation through fractal geometry. *Computer Vision, Graphics and Image Processing*, 45(2):150–166.
- [Khachiyan, 1979] KHACHIYAN, L. G. (1979). A polynomial algorithm in linear programming. *Soviet Mathematics, Doklady*, 20:191–194.
- [Kim *et al.*, 2001] KIM, K. I., PARK, S. H. et KIM, H. J. (2001). Kernel principal component analysis for texture classification. *Signal Processing Letters*, 8(2):39–41.
- [Kim et Kang, 2007] KIM, S. C. et KANG, T. J. (2007). Texture classification and segmentation using wavelet packet frame and gaussian mixture model. *Pattern Recognition*, 40(4):1207–1221.
- [Kingsford *et al.*, 2004] KINGSFORD, C. L., CHAZELLE, B. et SINGH, M. (2004). Solving and analyzing side-chain positioning problems using linear and integer programming. *Bioinformatics*, 21:1028–1039.
- [Klee et Minty, 1972] KLEE, V. et MINTY, G. L. (1972). *How good is the simplex algorithm ?* Academic Press, New York.
- [Kokare *et al.*, 2004] KOKARE, M., BISWAS, P. K. et CHATTERJI, B. N. (2004). Rotated complex wavelet based texture features for content based image retrieval. In *proceedings of the 17th International Conference on Pattern Recognition (ICPR04)*, volume 1, pages 652–655, Washington, DC, USA. IEEE Computer Society.
- [Kuhn et Tucker, 1951] KUHN, H. W. et TUCKER, A. W. (1951). Nonlinear programming. In *Proceedings of 2nd Berkeley Symposium*, pages 481–492.
- [Kurnaz *et al.*, 2007] KURNAZ, M. N., DOKUR, Z. et ÖLMEZ, T. (2007). An incremental neural network for tissue segmentation in ultrasound images. *Computer Methods and Programs in Biomedicine*, 85(3):187–195.
- [Lam et Yuen, 1998] LAM, C. L. et YUEN, S. Y. (1998). An unbiased active contour algorithm for object tracking. *Pattern Recognition Letters*, 19(5-6):491–498.
- [Leclerc, 1989] LECLERC, Y. (1989). Constructing simple stable descriptions for image segmentation. *International Journal of Computer Vision*, 3:71–102.
- [Li *et al.*, 2008] LI, B., TANG, Z., YUAN, B. et MIAO, Z. (2008). Segmentation of moving foreground objects using codebook and local binary patterns. In *proceedings of the 2008 Congress on Image and Signal Processing (CISP 08)*, volume 4, pages 239–243, Washington, DC, USA. IEEE Computer Society.
- [Li et Staunton, 2008] LI, M. et STAUNTON, R. C. (2008). Optimum Gabor filter design and local binary patterns for texture segmentation. *Pattern Recognition Letters*, 29(5):664–672.
- [Li, 1995] LI, S. Z. (1995). *Markov Random Field Modeling in Computer Vision*. Springer-Verlag New York, Inc.

- [Li, 2001] LI, S. Z. (2001). *Markov Random Field Modeling in Image Analysis*. Springer-Verlag New York, Inc.
- [Lianantonakis et Petillot, 2005] LIANANTONAKIS, M. et PETILLOT, Y. R. (2005). Sidescan sonar segmentation using active contours and level set methods. *In proceedings of Ocean 05 - Europe*, volume 1, pages 719–724.
- [Liang et al., 2001] LIANG, L., LIU, C., XU, Y.-Q., GUO, B. et SHUM, H.-Y. (2001). Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics*, 20(3): 127–150.
- [Lin et Ling, 2003] LIN, E. B. et LING, Y. (2003). Image compression and denoising via nonseparable wavelet approximation. *Journal of Computational Applied Mathematics*, 155(1):131–152.
- [Lippmann, 1987] LIPPMANN, R. P. (1987). An introduction to computing with neural nets. *IEEE Acoustic, Speech and Signal Processing Magazine*, 4:4–22.
- [Lorette et al., 2000] LORETTE, A., DESCOMBES, X. et ZERUBIA, J. (2000). Texture analysis through a markovian modelling and fuzzy classification : Application to urban area extraction from satellite images. *International Journal of Computer Vision*, 36(3):221–236.
- [Luo et al., 2007] LUO, J., MING, D., SHEN, Z., WANG, M. et SHENG, H. (2007). Multi-scale information extraction from high resolution remote sensing imagery and region partition methods based on GMRF-SVM. *International Journal of Remote Sensing*, 28(15):3395–3412.
- [Luo et Savakis, 2001] LUO, J. et SAVAKIS, A. E. (2001). Self-supervised texture segmentation using complementary types of features. *Pattern Recognition*, 34(11):2071–2082.
- [Luo et Mirchandani, 2000] LUO, X. et MIRCHANDANI, G. (2000). An integrated framework for image classification. *In proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP00)*, pages 620–623, Washington, DC, USA. IEEE Computer Society.
- [Malladi et Sethian, 1995] MALLADI, R. et SETHIAN, J. (1995). Level set methods for curvature flow, image enhancement, and shape recovery in medical images. *In proceedings of the conference on Visualization and Mathematics*, pages 329–345.
- [Malladi et Sethian, 1996] MALLADI, R. et SETHIAN, J. A. (1996). Image processing : Flows under min/max curvature and mean curvature. *Graphical Models and Image Processing*, 58(2):127–141.
- [Malladi et al., 1995] MALLADI, R., SETHIAN, J. A. et VEMURI, B. C. (1995). Shape modeling with front propagation : a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175.
- [Mallat, 1989] MALLAT, S. G. (1989). A theory for multiresolution signal decomposition : the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693.
- [Malpica et al., 2003] MALPICA, N., ORTUNO, J. E. et SANTOS, A. (2003). A multichannel watershed-based algorithm for supervised texture segmentation. *Pattern Recognition Letters*, 24(9-10):1545–1554.

- [Mandelbrot, 1977] MANDELBROT, B. B. (1977). *Fractals, Forms, Chance and Dimension*. Freeman, San Fransisco.
- [Mandelbrot, 1983] MANDELBROT, B. B. (1983). *The Fractal Geometry of Nature*. W. H. Freeman and Company, New York.
- [McInerney et Terzopoulos, 1995] MCINERNEY, T. et TERZOPOULOS, D. (1995). A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis. *Computerized Medical Imaging and Graphics*, 19(1):69–83.
- [Metropolis *et al.*, 1953] METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H. et TELLER, E. (1953). Equations of state calculations by fast computing machine. *Journal of Chemical Physics*, 21:1087–1091.
- [Mille *et al.*, 2007] MILLE, J., BONÉ, R., MAKRIS, P. et CARDOT, H. (2007). 2D and 3D deformable models with narrow band region energy. *In proceedings of the IEEE International Conference on Image Processing (ICIP07)*, pages 57–60, San Antonio, USA.
- [Mumford et Shah, 1989] MUMFORD, D. et SHAH, J. (1989). Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685.
- [Muzaffar et Choi, 2008] MUZAFFAR, T. et CHOI, T. S. (2008). Linked significant tree wavelet-based image compression. *Signal Processing*, 88(10):2554–2563.
- [Neumaier et Shcherbina, 2004] NEUMAIER, A. et SHCHERBINA, O. (2004). Safe bounds in linear and mixed-integer linear programming. *Mathematical Programming*, 99:283–296.
- [Ng et Bouzerdoum, 2000] NG, B. W. et BOUZERDOUM, A. (2000). Supervised texture segmentation using dwt and a modified K-NN classifier. *In Proceedings of the 15th International Conference on Pattern Recognition (ICPR00)*, volume 2, pages 545–548. IEEE Computer Society.
- [Nikam et Agarwal, 2008] NIKAM, S. B. et AGARWAL, S. (2008). Local binary pattern and wavelet-based spoof fingerprint detection. *International Journal of Biometrics*, 1(2):141–159.
- [Núñez, 2000] NÚÑEZ, A. M. (2000). Texture segmentation of a 3D seismic section with wavelet transform and Gabor filters. *In proceedings of the International Conference on Pattern Recognition (ICPR00)*, page 3358, Washington, DC, USA. IEEE Computer Society.
- [Ohanian et Dubes, 1992] OHANIAN, P. P. et DUBES, R. C. (1992). Performances evaluation of textural features. *Pattern Recognition*, 25(8):819–833.
- [Ojala *et al.*, 2002] OJALA, T., PIETIKAINEN, M. et MAENPAA, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:971–987.
- [Ojala *et al.*, 1996] OJALA, T., PIETIKÄINEN, M. et HARWOOD, D. (1996). A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59.
- [Osher et Sethian, 1988] OSHER, S. et SETHIAN, J. A. (1988). Fronts propagation with curvature-dependent speed : algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49.

- [Otsu et Kurita, 1988] OTSU, N. et KURITA, T. (1988). A new scheme for practical flexible and intelligent vision systems. *In IAPR Workshop on Computer Vision*, pages 431–435, Tokyo, Japan.
- [Ozden et Polat, 2005] OZDEN, M. et POLAT, E. (2005). Image segmentation using color and texture features. *In proceedings of the 13th European Signal Processing Conference (EUSIPCO05)*, Antalya, Turkey.
- [Paget et Longstaff, 1998] PAGET, R. et LONGSTAFF, I. D. (1998). Texture synthesis and unsupervised recognition with a nonparametric multiscale Markov random field model. *In proceedings of the 14th International Conference on Pattern Recognition (ICPR98)*, volume 2, pages 1068–1070.
- [Paragios et Deriche, 1999] PARAGIOS, N. et DERICHE, R. (1999). Geodesic active regions for supervised texture segmentation. *In proceedings of the International Conference on Computer Vision (ICCV99)*, volume 2, pages 926–932, Washington, DC, USA. IEEE Computer Society.
- [Paragios et Deriche, 2000] PARAGIOS, N. et DERICHE, R. (2000). Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280.
- [Paragios et Deriche, 2002] PARAGIOS, N. et DERICHE, R. (2002). Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247.
- [Paragios et Deriche, 2005] PARAGIOS, N. et DERICHE, R. (2005). Geodesic active regions and level set methods for motion estimation and tracking. *Computer Vision and Image Understanding*, 97(3):259–282.
- [Paragios et al., 2004] PARAGIOS, N., MELLINA-GOTTARDO, O. et RAMESH, V. (2004). Gradient vector flow fast geometric active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):402–407.
- [Peleg et al., 1984] PELEG, S., NAOR, J., HARTLEY, R. et AVNIR, D. (1984). Multiple resolution texture analysis and classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(4):518–523.
- [Peng et al., 2009] PENG, Y., PI, L. et SHEN, C. (2009). A semi-automatic method for burn scar delineation using a modified Chan-Vese model. *Computer and Geosciences*, 35(2):183–190.
- [Pentland, 1984] PENTLAND, A. (1984). Fractal-based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):661–674.
- [Perng et Lin, 2006] PERNG, M. H. et LIN, H. H. (2006). Image compression in the wavelet domain using an AR texture model with compressed initial conditions. *In proceedings of the 5th WSEAS International Conference on Signal Processing, Robotics and Automation (ISPRA06)*, pages 46–51, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society (WSEAS).
- [Pichler et al., 1996] PICHLER, O., TEUNER, A. et HOSTICKA, B. J. (1996). A comparison of texture feature extraction using adaptative Gabor filtering, pyramidal and tree-structured wavelet transforms. *Pattern Recognition Letters*, 29(5):733–742.

- [Pieczynski, 2003] PIECZYNSKI, W. (2003). Modèles de Markov en traitements d'images. *Traitement du Signal*, 20(3):255–278.
- [Pisa et Ruiz, 1998] PISA, I. et RUIZ, E. (1998). Tracking left ventricular wall motion using active contour model : PR-greedy algorithm. *In proceedings of the International Symposium on Computer Graphics, Image Processing and Vision (SIBGRAPI98)*, pages 101–104, Rio de Janeiro, Brazil.
- [Platt et al., 1999] PLATT, J., SCHOLKOPK, C. et SMOLA, A. (1999). *Fast training of support vector machines using sequential minimal optimisation*. MIT Press, Cambridge.
- [Portilla et Simoncelli, 2000] PORTILLA, X. et SIMONCELLI, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70.
- [Pratt et al., 1978] PRATT, W. K., FAUGERAS, O. D. et GAGALOWICZ, A. (1978). Visual discrimination of stochastic texture fields. *IEEE Transactions on Systems, Man and Cybernetics*, 8(11):796–804.
- [Qiao et al., 2007] QIAO, Y. L., SONG, C. Y. et ZHAO, C. H. (2007). Double-density discrete wavelet transform based texture classification. *In proceedings of the Third International Conference on International Information Hiding and Multimedia Signal Processing (IIH-MSP07)*, pages 91–94, Washington, DC, USA. IEEE Computer Society.
- [Qin et al., 2008] QIN, L., ZHENG, Q., JIANG, S., HUANG, Q. et GAO, W. (2008). Un-supervised texture classification : Automatically discover and classify texture patterns. *Image and Vision Computing*, 26(5):647–656.
- [Raghavan et Birchard, 1979] RAGHAVAN, V. V. et BIRCHARD, K. (1979). A clustering strategy based on a formalism of the reproductive process in natural systems. *In proceedings of the Second International Conference on Information Storage and Retrieval, Information Implications into the Eighties*, pages 10–22. ACM.
- [Raghu et al., 1997] RAGHU, P., POONGODI, R. et YEGNANARAYANA, B. (1997). Unsupervised texture classification using vector quantization and deterministic relaxation neural network. *IEEE Transactions on Image Processing*, 6(2):1376–1387.
- [Raju et al., 2008] RAJU, U. S. N., VIJAYA KUMAR, V., SURESH, A. et RADHIKA MANI, M. (2008). Texture description using different wavelet transforms based on statistical parameters. *In proceedings of the 2nd WSEAS International Conference on Wavelets Theory and Applications in Applied Mathematics, Signal Processing and Modern Science (WAV08)*, pages 174–178, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society (WSEAS).
- [Ramakrishnan et Selvan, 2006] RAMAKRISHNAN, S. et SELVAN, S. (2006). Wavelet-based modeling of singular values for image texture classification. *Machine Graphics and Vision*, 15(2):211–225.
- [Randen et Husoy, 1999] RANDEN, T. et HUSOY, J. H. (1999). Filtering for texture classification : A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:291–310.
- [Ribeiro et Hancock, 2002] RIBEIRO, E. et HANCOCK, E. R. (2002). Estimating the perspective pose of texture planes using spectral analysis on the unit sphere. *Pattern Recognition*, 35(10):2141–2163.

- [Richards et Polit, 1974] RICHARDS, W. et POLIT, A. (1974). Texture matching. *Kibernetik*, 16:155–162.
- [Roach et Mason, 1995] ROACH, M. et MASON, J. (1995). Saliency distance transform. *Graphical Models and Image Processing*, 57(6):483–521.
- [Ronfard, 1994] RONFARD, R. (1994). Region based strategies for active contour models. *International Journal of Computer Vision*, 13(2):229–251.
- [Rosenblatt, 1958] ROSENBLATT, F. (1958). The perceptron : A theory of statistical separability in cognitive systems. *Technical report VG-1196-G-1*, M. A. Cornell Aeronautical Laboratory.
- [Rousseau et al., 2003] ROUSSEAU, F., FABLET, R. et BARILLOT, C. (2003). Robust statistical registration of 3d ultrasound images using texture information. *In proceedings of the IEEE International Conference on Image Processing (ICIP03)*, Barcelona, Spain.
- [Rousson et al., 2003] ROUSSON, M., BROX, T. et DERICHE, R. (2003). Active unsupervised texture segmentation on a diffusion based feature space. *In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR03)*, pages 231–234, Madison, WI.
- [Ruan et al., 2002] RUAN, S., MORETTI, B., FADILI, J. et BLOYET, D. (2002). Fuzzy markovian segmentation in application of magnetic resonance images. *Computer Vision and Image Understanding*, 85(1):54–69.
- [Rucklidge, 1996] RUCKLIDGE, W. (1996). *Efficient Visual Recognition Using the Hausdorff Distance*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [Rumelhart et al., 1986] RUMELHART, D. E., HINTON, G. E. et WILLIAMS, R. J. (1986). Learning internal representations by error propagation. *In Parallel Distributed Processes Vol. 1*, pages 318–362. MIT Press, Cambridge, MA.
- [Safont et Marroquin, 1999] SAFONT, L.-V. et MARROQUIN, E.-M. (1999). 3D reconstruction of third proximal femur (31-) with active contours. *In proceedings of the 10th International Conference on Image Analysis and Processing (ICIAP99)*, page 458, Washington, DC, USA. IEEE Computer Society.
- [Samir et al., 2007] SAMIR, C., DAOUDI, M. et STRIVASTAVA, A. (2007). Human identification using facial curves with extension to joint shape-texture analysis. *In proceedings of the 2nd international conference on computer vision theory and applications*, Barcelona, Spain.
- [Schneuders, 2001] SCHNEUDERS, P. (2001). Multivalued image segmentation based on first fundamental form. *In proceedings of the 11th International Conference on Image Analysis and Processing (ICIAP01)*, pages 185–190. IEEE Computer Society.
- [Schupp et al., 2000] SCHUPP, S., ELMOATAZ, A., FADILI, J., HERLIN, P. et BLOYET, D. (2000). Image segmentation via multiple active contour models and fuzzy clustering with biomedical applications. *In proceedings of the International Conference on Pattern Recognition (ICPR00)*, pages 622–625, Washington, DC, USA. IEEE Computer Society.
- [Sethian, 1995] SETHIAN, J. (1995). A review of the theory, algorithms, and applications of level set method for propagating surfaces. *Acta numerica*.

- [Sethian, 1996] SETHIAN, J. A. (1996). A fast marching level set method for monotonically advancing fronts. *In Proceedings of the National Academy of Science*, volume 93, pages 1591–1595.
- [Shamir, 2006] SHAMIR, A. (2006). Segmentation and shape extraction of 3d boundary meshes. *In State-of-the-Art Report, Proceedings of Eurographics 2006*, pages 137–149.
- [Shamos et Hoey, 1975] SHAMOS, M. I. et HOEY, D. (1975). Closest-point problems. *the 16th Annual Symposium on Foundations of Computer Science*, pages 131–162.
- [Sharma et Singh, 2001] SHARMA, M. et SINGH, S. (2001). Evaluation of texture methods for image analysis. *In proceedings of the 7th Australian and New Zealand Intelligent Information System Conference (ANZIIS01)*, pages 117–121, Perth, Western Australia.
- [Shi et Karl, 2005a] SHI, Y. et KARL, W. (2005a). A fast level set method without solving PDEs. *In proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP05)*, volume 2, pages 97–100, Philadelphia, USA.
- [Shi et Karl, 2005b] SHI, Y. et KARL, W. (2005b). Real-time tracking using level sets. *In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR05)*, volume 2, pages 34–41, San Diego, USA.
- [Shor, 1970] SHOR, N. Z. (1970). Convergence rate of the gradient descent method with dilatation of the space. *Kibernetika*, 2:80–85.
- [Singh et Noore, 2008] SINGH, R. and VATSA, M. et NOORE, A. (2008). Integrated multilevel image fusion and match score fusion of visible and infrared face images for robust face recognition. *Pattern Recognition*, 41(3):880–893.
- [Smutek et al., 2003] SMUTEK, D., ŠÁRA, R., SUCHARDA, P., TJAHHADI, T. et ŠVEC, M. (2003). Image texture analysis of sonograms in chronic inflammations of thyroid gland. *Ultrasound in Medicine and Biology*, 29(11):1531–1543.
- [Soille et Rivest, 1996] SOILLE, P. et RIVEST, J.-F. (1996). On the validity of fractal dimension measurements in image analysis. *Journal of Visual Communication and Image Representation*, 7(3):217–229.
- [Solem et al., 2006] SOLEM, J. E., OVERGAARD, N. C. et HEYDEN, A. (2006). Initialization techniques for segmentation with the Chan-Vese model. *In proceedings of the 18th International Conference on Pattern Recognition (ICPR06)*, pages 171–174, Washington, DC, USA. IEEE Computer Society.
- [Stachowiak et al., 2005] STACHOWIAK, G. P., PODSIADLO, P. et STACHOWIAK, G. W. (2005). A comparison of texture feature extraction methods for machine condition monitoring and failure analysis. *Tribology Letters*, 20(2):133–147.
- [Stevens, 1980] STEVENS, K. A. (1980). Surface perception from local analysis of texture and contour. Rapport technique.
- [Stewart et al., 2008] STEWART, L., HE, X. et ZEMEL, R. (2008). Learning flexible features for conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1415–1426.
- [Sundaramoorthi et al., 2007] SUNDARAMOORTHY, G., YEZZI, A. et MENNUCCI, A. (2007). Sobolev active contours. *International Journal of Computer Vision*, 73(3):345–366.

- [Super et Bovik, 1995] SUPER, B. J. et BOVIK, A. C. (1995). Shape from texture using local spectral moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):333–343.
- [Supot *et al.*, 2007] SUPOT, S., THANAPONG, C., CHUCHART, P. et MANAS, S. (2007). Automatic segmentation of blood vessels in retinal image based on fuzzy k-median clustering. In *proceedings of the IEEE International Conference on Integration Technology (ICIT07)*, pages 548–588, Washington, DC, USA. IEEE Computer Society.
- [Sweet et Ware, 2004] SWEET, G. et WARE, C. (2004). View direction, surface orientation and texture orientation for perception of surface shape. In *proceedings of the Graphics Interface conference (GI04)*, pages 97–106, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada. Canadian Human-Computer Communications Society.
- [Takaya, 2008] TAKAYA, K. (2008). Tracking a video object with the active contour (snake) predicted by the optical flow. In *proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE08)*, pages 369–372, Niagara Falls, Ontario, Canada.
- [Talbar *et al.*, 1998] TALBAR, S. N., HOLAMBE, R. S. et SONTAKKE, T. R. (1998). Supervised texture classification using wavelet transform. In *proceedings of the 4th IEEE International Conference on Signal Processing (ICSP98)*, volume 2, pages 1177–1180, Beijing, China.
- [Tesar *et al.*, 2007] TESAR, L., SMUTEK, D., SHIMIZU, A. et KOBATAKE, H. (2007). Medical image segmentation using cooccurrence matrix based texture features calculated on weighted region. In *proceedings of Advances in Computer Science and Technology (ACST07)*, pages 243–248, Phuket, Thailand.
- [Tierny *et al.*, 2007] TIERNY, J., VANDEBORRE, J.-P. et DAOUDI, M. (2007). Topology driven 3D mesh hierarchical segmentation. In *proceedings of the IEEE International Conference on Shape Modeling and Applications (SMI2007)*, pages 215–220, Lyon, France.
- [Tikhonov et Arsénine, 1976] TIKHONOV, A. et ARSÉNINE, V. (Moscou, URSS, 1976). Méthodes de résolution de problèmes mal posés. *Editions MIR*.
- [Toyoda et Hasegawa, 2007] TOYODA, T. et HASEGAWA, O. (2007). Extension of higher order local autocorrelation features. *Pattern Recognition*, 40(5):1466–1473.
- [Tuceryan et Jain, 1990] TUCERYAN, M. et JAIN, A. K. (1990). Texture segmentation using voronoi polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):211–216.
- [Tuceryan et Jain, 1993] TUCERYAN, M. et JAIN, A. K. (1993). *Texture analysis*, pages 235–276. World Scientific Publishing Co., Inc., River Edge, NJ, USA.
- [Turner, 1986] TURNER, M. R. (1986). Texture discrimination by Gabor functions. *Biological Cybernetics*, 55:443–457.
- [Unser, 1995] UNSER, M. (1995). Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, 4(11):1549–1560.
- [Van Nevel, 1998] VAN NEVEL, A. (1998). Texture synthesis via matching first and second order statistics of a wavelet frame decomposition. In *proceedings of the International Conference on Image Processing (ICIP98)*, volume 1, pages 72–76, Los Alamitos, CA, USA. IEEE Computer Society.

- [Vapnik, 1995] VAPNIK, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.
- [Voorhees et Poggio, 1987] VOORHEES, H. et POGGIO, T. (1987). Detecting textons and texture boundaries in natural images. *In proceedings of the International Conference on Computer Vision (ICCV87)*, pages 250–258, London, England.
- [Wagner et al., 2004] WAGNER, M., MELLER, J. et ELBER, R. (2004). Large-scale linear programming techniques for the design of protein folding potentials. *Mathematical Programming*, 101:301–318.
- [Waku et Chassery, 1993] WAKU, J. et CHASSERY, J.-M. (1993). Spécification d’une ondelette pour la représentation en multirésolution d’un contour discret. *Traitement du Signal*, 10(3):231–240.
- [Wan et Hou, 2006] WAN, S.-Y. et HOU, C.-H. (2006). Synergy of symmetric region grow and active contour in reconstruction of a 3D rat. *In proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems (CBMS06)*, pages 436–441, Washington, DC, USA.
- [Wang et He, 1990] WANG, L. et HE, D.-C. (1990). Texture classification using texture spectrum. *Pattern Recognition*, 23(8):905–910.
- [Wang et Meng, 2008] WANG, X. et MENG, J. (2008). A 2-D ECG compression algorithm based on wavelet transform and vector quantization. *Digital Signal Processing*, 18(2):179–188.
- [Wei, 2002] WEI, J. (2002). Image segmentation based on situational DCT descriptors. *Pattern Recognition Letters*, 23(1-3):295–302.
- [Williams et Shah, 1992] WILLIAMS, D. J. et SHAH, M. (1992). A fast algorithm for active contours and curvature estimation. *Computer Vision, Graphics, and Image Processing : Image Understanding*, 55(1):14–26.
- [Witkin, 1981] WITKIN, A. P. (1981). Recovering surface shape and orientation from texture. *Journal of Artificial Intelligence*, 17:17–25.
- [Xu et Prince, 1998] XU, C. et PRINCE, J. (1998). Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369.
- [Xu et Prince, 1997] XU, C. et PRINCE, J. L. (1997). Gradient vector flow : a new external force for snakes. *In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR97)*, pages 66–71, San Juan, Porto Rico.
- [Yagi et al., 2000] YAGI, Y., KAWASAKI, Y., YACHIDA, M. et BRADY, M. (2000). Active contour road model for smart vehicle. *proceedings of the International Conference on Pattern Recognition (ICPR00)*, 3:811–814.
- [Yamamoto et al., 2006] YAMAMOTO, K., ISHII, I. et KUBOZONO, M. (2006). Higher order autocorrelation vision chip. *IEEE Transactions on Electron Devices*, 53(8):1797–1804.
- [Yanover et al., 2006] YANOVER, C., MELTZER, T. et WEISS, Y. (2006). Linear programming relaxations and belief propagation – an empirical study. *Journal of Machine Learning Research*, 7:1887–1907.
- [Yudin et Nemirovskii, 1976] YUDIN, D. B. et NEMIROVSKII, A. S. (1976). Informational complexity and efficient methods for solving complex extremal problems. *Ekonomika i Matem*, 12:357–359.

- [Yui *et al.*, 2002] YUI, S., HARA, K., ZHA, H. et HASEGAWA, T. (2002). A fast narrow band method and its application in topology-adaptive 3-D modeling. *In proceedings of the 16 th International Conference on Pattern Recognition (ICPR02)*, volume 4, pages 122–125, Washington, DC, USA.
- [Zhan et Shen, 2006] ZHAN, Y. Q. A. et SHEN, D. G. (2006). Deformable segmentation of 3-D ultrasound prostate images using statistical texture matching method. *IEEE Transactions on Medical Imaging*, 25(3):256–272.
- [Zhu et Yuille, 1996] ZHU, S. et YUILLE, A. (1996). Region competition : unifying snake/balloon, region growing, and Bayes/MDL/energy for multi-band image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900.
- [Zucker, 1976] ZUCKER, S. W. (1976). Toward a model of texture. *Computer Graphics and Image Processing*, 5:190–202.

Publications de l'auteur

Revues internationales

J. Olivier, J. Mille, R. Boné, and J-J. Rousselle. Dynamic neighborhoods in active surfaces for 3d segmentation. *International Journal for Computational Vision and Biomechanics*, 1(2), July- December 2008.

Conférences Internationales avec comité de lecture

J. Olivier, C. Mocquilon, J-J. Rousselle, R. Boné, and H. Cardot. A supervised texture-based active contour model with linear programming. *In IEEE Conference on Image Processing (ICIP08)*, San Diego, California, October, 12-15 2008.

J. Olivier, J-J. Rousselle, R. Boné, and H. Cardot. Active contours driven by supervised binary classifiers for texture segmentation. *In the 4th International Symposium on Visual Computing (ISVC08)*, Las Vegas, Nevada, December 1-3 2008.

L. Paulhac, J. Olivier, and J-J. Rousselle. Automatic computation of parameter in a fast level set method. *In International Conference on Communication, Computer and Power (ICCCP07)*, pages 427-431, MUSCAT (SULTANATE OF OMAN), November 2007.

J-J. Rousselle, R. Boné, and J. Olivier. Speed up active contours using line search. *In International Conference on Communication, Computer and Power (ICCCP07)*, pages 66-69, MUSCAT (SULTANATE OF OMAN), November 2007.

J. Olivier, J. Mille, R. Boné, and J-J. Rousselle. Active surfaces acceleration methods. *In Computational Modelling of Objects Represented in Images : fundamentals, methods and applications (CompIMAGE06)*, Coimbra, Portugal, October 2006.

J. Olivier, R. Boné, and J-J. Rousselle. Comparison of active contour acceleration methods. *In the 5th International Conference on Visualization, Imaging & Image Processing (VIIP05)*, Benidorm, Spain, September 2005.

Conférences Nationales avec comité de lecture

J. Olivier, R. Boné, H. Cardot, and J-J. Rousselle. Guidage de contour actif par classificateur binaire supervisé pour la segmentation d'images texturées. *In colloque du Groupe de Recherche et d'Etudes du Traitement du Signal (GRETSI09)*, Dijon, France, September, 8-11 2009.

J. Olivier, J. Mille, R. Boné, and J-J. Rousselle. Accélération de surfaces actives. *In Compression et Représentation des Signaux Audiovisuels (CORESA06)*, pages 225-229, Caen, France, November 2006.

J-J. Rousselle, R. Boné, and J. Olivier. Accélération d'un contour actif par "line search". *In Colloque Nationale de Recherche en IUT (CNRIUT06)*, Brest, France, 2006.

Résumé

Dans cette thèse, plusieurs approches permettant d'améliorer les contours actifs sont présentées. Trois méthodes d'accélération ont été développées pour les modèles paramétriques se déformant par évolution gloutonne et appliquées à la segmentation 2D et 3D. Leur principe est d'autoriser une gestion dynamique de la grille de voisinage de chaque point de contrôle du contour actif. Deux modèles supervisés implémentés à l'aide des ensembles de niveaux sont également détaillés. Ceux-ci se basent sur les caractéristiques de textures d'Haralick et utilisent une image d'apprentissage possédant une segmentation experte. Le premier modèle est un contour actif basé région inspiré du modèle de Chan et Vese. Le principe de la programmation linéaire est alors utilisé pour déterminer le poids optimal à affecter à chaque coefficient d'Haralick. Le deuxième modèle introduit un classificateur binaire, appris grâce à la segmentation experte, directement dans l'équation d'évolution du contour actif. Les deux modèles sont appliqués à la segmentation d'images texturées 2D et 3D.

Mots clés

Contours actifs, modèles déformables, évolution gloutonne, accélération, ensembles de niveaux, segmentation supervisée, classification supervisée, segmentation de textures, segmentation d'images échographiques, 3D.

Abstract

In this work, several approaches developed to improve active contours are presented. Three acceleration methods have been developed for parametric models evolving with the greedy algorithm, and applied to 2D and 3D segmentation. Their principle is to dynamically manage the neighbourhood grid of each control point of the active contour. Two supervised level set models are also detailed. Both are based on Haralick texture features and use a learning image with an expert segmentation. The first model is a region-based active contour, inspired by the model developed by Chan and Vese. Linear programming principle is used to determine the optimal weight of each Haralick coefficient. The second model introduces a binary classifier in the motion equation of the active contour, the classifier being learned using the Haralick coefficients, extracted from the learning image. Both models are applied to 2D and 3D textured image segmentation.

Keywords

Active contours, deformable models, greedy algorithm, acceleration, level sets, supervised segmentation, supervised classification, texture segmentation, echographic image segmentation, 3D.