



HAL
open science

Surveillance maritime par analyse d'images satellitaires optiques panchromatiques

Nadia Proia

► **To cite this version:**

Nadia Proia. Surveillance maritime par analyse d'images satellitaires optiques panchromatiques. Informatique [cs]. Université des Antilles-Guyane, 2010. Français. NNT : . tel-00560119

HAL Id: tel-00560119

<https://theses.hal.science/tel-00560119>

Submitted on 27 Jan 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DES ANTILLES ET DE LA GUYANE

THÈSE

présentée pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DES ANTILLES ET DE LA GUYANE

Spécialité : Informatique

par

Nadia PROIA

Titre :

**Surveillance maritime par analyse d'images
satellitaires optiques panchromatiques**

soutenue publiquement le 29 octobre 2010

JURY

| | |
|---|--------------|
| Marie-José LEFEVRE-FONOLLOSA, Expert Senior service PASO, CNES Toulouse | Rapporteurs |
| Grégoire MERCIER, Professeur, TELECOM Bretagne Brest | |
| Josiane ZERUBIA, Directrice de Recherche INRIA, Sophia-Antipolis | Examineurs |
| Richard NOCK, Professeur, Université des Antilles-Guyane | |
| François MARQUES, PDG de Nevantropic, Cayenne | |
| Jacky DESACHY, Professeur, Université des Antilles-Guyane | Directeur |
| Michel PETIT, Directeur de Recherche IRD, Montpellier | Co-directeur |
| Vincent PAGE, Maître de conférences, Université des Antilles-Guyane | Co-encadrant |

Remerciements

Je remercie les membres du jury

- Madame le Professeur Marie-José Lefèvre-Fonollosa et Monsieur le Professeur Grégoire Mercier, pour m’avoir fait l’honneur d’être les rapporteurs de mon manuscrit de thèse de doctorat.
- Madame le Professeur Josiane Zerubia et Monsieur le Professeur Richard Nock, pour avoir accepté de juger ce travail et de participer à ce jury. Je tiens à vous exprimer toute ma gratitude pour avoir consacré une partie de votre temps à lire et juger ce travail.
- Monsieur François Marques, PDG de la société Nevantropic qui a financé ces travaux de recherche.
- Monsieur le Professeur Jacky Desachy, mon directeur de thèse, pour m’avoir fait confiance pour la réalisation de cette thèse de doctorat et qui m’a aidé par ces remarques et conseils.
- Monsieur le Professeur Michel Petit, pour m’avoir fait l’honneur de co-diriger cette thèse.
- Monsieur le Docteur Vincent Page, mon encadrant, pour m’avoir guidé, conseillé et encouragé tout au long de cette thèse.

Je remercie les membres du laboratoire qui ont accepté d’examiner mon manuscrit.

Je remercie également mes collègues de Nevantropic, pour leur aide durant mes recherches.

Table des matières

| | |
|--|-----------|
| Introduction | 1 |
| 1 La problématique | 3 |
| 1.1 Introduction | 3 |
| 1.2 Le contexte | 3 |
| 1.2.1 IBIS (<i>Implementation of Boat Information System</i>) | 4 |
| 1.2.2 DECLIMS (<i>Detection, Classification and Identification of Maritime traffic from Space</i>) | 4 |
| 1.2.3 LIMES (<i>Land and Sea Monitoring for European Security</i>) | 4 |
| 1.2.4 OCEANWAY | 5 |
| 1.3 Les données | 6 |
| 1.3.1 Les images optiques | 6 |
| 1.3.2 Les images SAR | 7 |
| 1.3.3 Les images utilisées | 7 |
| 1.4 Le contenu des scènes | 8 |
| 1.4.1 Les mobiles en mer | 9 |
| 1.4.2 Les sillages | 11 |
| 1.4.3 La terre | 12 |
| 1.4.4 Les zones de mer | 13 |
| 1.4.5 Les nuages | 13 |
| 1.5 Notre problématique | 14 |
| 1.5.1 Nos objectifs | 14 |
| 1.5.2 Notre démarche | 15 |
| 1.6 Les bases d'images | 18 |
| 1.6.1 Base d'images 1 | 18 |
| 1.6.2 Base d'images 2 | 20 |
| 1.6.3 Base d'images 3 | 21 |
| 1.7 Conclusion | 22 |
| 2 Pré-détection des cibles | 23 |
| 2.1 Introduction | 23 |
| 2.2 Méthode de pré-détection | 24 |
| 2.2.1 Principe | 24 |
| 2.2.2 Règle de décision de Neyman-Pearson | 26 |
| 2.2.3 Modèle utilisé | 27 |

| | | |
|----------|--|-----------|
| 2.2.4 | Rapport de vraisemblance généralisé | 29 |
| 2.2.5 | Hypothèse sur les variances | 29 |
| 2.2.6 | Taux de fausse alarme constant | 30 |
| 2.2.7 | L'algorithme de pré-détection | 31 |
| 2.2.8 | Comportement de l'algorithme en présence d'une cible | 31 |
| 2.2.9 | Les paramètres de la méthode | 34 |
| 2.2.10 | Méthode d'évaluation des algorithmes | 34 |
| 2.2.11 | Choix d'une taille de fenêtre d'analyse | 36 |
| 2.2.12 | Choix du seuil de détection | 37 |
| 2.3 | Discussion sur notre modèle | 42 |
| 2.4 | Résultats des expérimentations | 46 |
| 2.4.1 | Résultats sur la base d'images 1 | 46 |
| 2.4.2 | Exemples | 46 |
| 2.5 | Conclusion | 49 |
| 3 | Segmentation et classification | 51 |
| 3.1 | Introduction | 51 |
| 3.2 | La segmentation | 51 |
| 3.2.1 | Les contours actifs | 52 |
| 3.2.2 | Notre approche | 52 |
| 3.2.3 | Energie utilisée | 54 |
| 3.2.4 | Algorithme d'optimisation | 55 |
| 3.2.5 | Couplage de la pré-détection et de la segmentation | 57 |
| 3.2.6 | Exemples | 57 |
| 3.3 | La classification | 61 |
| 3.3.1 | Extraction des caractéristiques | 61 |
| 3.3.2 | Les échantillons d'apprentissage et de généralisation | 67 |
| 3.3.3 | Le classifieur | 68 |
| 3.4 | Conclusion | 74 |
| 4 | Détection des sillages | 77 |
| 4.1 | Introduction | 77 |
| 4.2 | Echantillon d'expérimentation | 80 |
| 4.3 | L'approche contours actifs | 80 |
| 4.3.1 | Recherche de la direction du sillage | 81 |
| 4.3.2 | Segmentation du sillage | 84 |
| 4.3.3 | Exemples | 85 |
| 4.4 | L'approche <i>a contrario</i> | 86 |
| 4.4.1 | Principes de l'approche <i>a contrario</i> | 86 |
| 4.4.2 | Détection des sillages par une approche <i>a contrario</i> | 88 |
| 4.4.3 | Le couplage bateau/sillage | 94 |
| 4.4.4 | Implémentation de la méthode | 102 |
| 4.4.5 | Conclusion | 104 |
| 4.5 | Expérimentation et résultats | 104 |
| 4.5.1 | Approche contours actifs | 104 |

| | | |
|----------|---|------------|
| 4.5.2 | Approche <i>a contrario</i> | 106 |
| 4.6 | Comparaison des deux approches | 106 |
| 4.7 | Conclusion | 110 |
| 5 | Expérimentation et résultats de l'algorithme complet | 111 |
| 5.1 | Introduction | 111 |
| 5.2 | La base d'expérimentation | 112 |
| 5.3 | Les résultats | 112 |
| 5.3.1 | La pré-détection | 112 |
| 5.3.2 | La segmentation | 113 |
| 5.3.3 | La classification | 113 |
| 5.3.4 | L'algorithme complet | 115 |
| 5.4 | Problèmes rencontrés et les solutions proposées | 116 |
| 5.5 | Exemple | 119 |
| 5.5.1 | Les petits bateaux détectés | 120 |
| 5.5.2 | Les grands bateaux détectés | 126 |
| 5.5.3 | Les fausses alarmes | 127 |
| 5.6 | Conclusion | 130 |
| | Conclusion et perspectives | 131 |
| A | Détails des calculs | 135 |
| A.1 | Calculs pour l'algorithme GLRT-Var | 135 |
| A.2 | Calculs pour l'algorithme GLRT | 136 |
| B | Choix du paramètre L_A pour GLRT-Var et GLRT-Gamma | 137 |
| B.1 | Choix de la taille de la fenêtre d'analyse pour GLRT-Var | 137 |
| B.2 | Choix de la taille de la fenêtre d'analyse pour GLRT-Gamma | 138 |
| C | Compléments sur la détection des sillages | 141 |
| C.1 | Influence du nombre de sillages et de θ sur r | 141 |
| C.2 | Influence du nombre de sillages et de r sur θ | 142 |
| | Bibliographie | 144 |
| | Publications | 151 |

Liste des abréviations

| | |
|---------|---|
| CFAR | Constant False Alarm Rate |
| DECLIMS | Detection, Classification and Identification of Maritime traffic from Space |
| GLRT | Generalized Likelihood Ratio Test |
| GPU | Graphics Processing Unit |
| IBIS | Implementation of Boat Information System |
| LIMES | Land and Sea Monitoring for European Security |
| ROC | Receiver Operating Characteristic |
| ROI | Region Of Interest |
| RSO | Radar à Synthèse d'Ouverture |
| SAR | Synthetic Aperture Radar |
| SIG | Système d'Information Géographique |
| TFAC | Taux de Fausse Alarme Constant |
| VMS | Vessel Monitoring System |

Introduction

Dans un contexte de dégradation de la sécurité au niveau mondial, les menaces sécuritaires provoquées par divers trafics (stupéfiants, armes), par l'immigration clandestine, par la pêche illégale ou par la pollution par hydrocarbures ont montré l'importance d'une surveillance des zones maritimes. Ainsi, la surveillance maritime constitue aujourd'hui un énorme enjeu dans le domaine de la sécurité et dans celui de l'économie.

Cette surveillance peut être réalisée par le biais de radars ou de façon aéroportée. Mais l'étendue du territoire à surveiller rend la télédétection aérospatiale bien adaptée à cette problématique. C'est dans ce cadre que s'inscrit cette thèse. Nous nous intéressons particulièrement à la surveillance des activités de pêche; c'est-à-dire à la détection de petits navires (entre 15 et 30 mètres). Nous verrons que les méthodes développées détectent également de grands bateaux.

Cette étude a été réalisée en collaboration avec la société Nevantropic qui a financé ce travail doctoral. Le but de ces travaux est de déterminer la possibilité d'automatiser, partiellement ou totalement, la détection de bateaux à partir d'images satellitaires optiques haute résolution.

Dans le **chapitre 1**, nous exposerons le contexte industriel dans laquelle s'insère notre étude. Ce qui permettra de fixer les contraintes en termes de performances et de temps de traitement. Puis, nous présenterons rapidement les différents types de données satellitaires disponibles pour la surveillance maritime et nous spécifierons celles disponibles dans le cadre de notre étude (images SPOT 5 panchromatique de résolution 5m). Après avoir illustré le contenu des scènes, nous présenterons la démarche adoptée pour la détection des navires : une phase de pré-détection (afin de définir des régions d'intérêts pouvant contenir des bateaux) suivie d'une étape de segmentation (pour définir le contour précis des bateaux) et d'une étape de classification (afin de classer les contours en trois classes : petit bateau, grand bateau, tout ce qui n'est pas un bateau). Enfin nous détaillerons les bases d'images que nous avons constituées pour tester les algorithmes.

Dans le **chapitre 2**, nous décrirons la méthode de pré-détection utilisée. Elle s'appuie sur la théorie bayésienne de la décision. Après avoir présenté le modèle classique, nous discuterons du modèle que nous avons utilisé. Cette pré-détection fournit un ensemble de régions d'intérêts. Parmi elles, nous n'avons pas uniquement des bateaux. Nous avons

également des fausses alarmes dues aux nuages et aux crêtes de vagues. Le but de l'étape suivante sera à la fois d'éliminer les fausses alarmes et de caractériser les bateaux pré-détectés.

Dans le **chapitre 3**, nous détaillerons la méthode utilisée pour segmenter les bateaux. Elle s'appuie sur des contours actifs et utilise également la théorie bayésienne de la décision. Après avoir décrit les caractéristiques extraites pour la classification des bateaux, nous présenterons brièvement le classifieur utilisé. Notre objectif n'est pas de développer une méthode de classification, mais plutôt de tester un ensemble de classifieurs pour en retenir un qui donne des résultats satisfaisants pour notre problématique.

Nous compléterons la caractérisation des navires en présentant, dans le **chapitre 4**, deux méthodes de détection des sillages différentes des méthodes classiques. La première approche s'appuie sur l'utilisation de contours actifs et la seconde est basée sur une approche *a contrario*. Nous comparerons les résultats des ces deux approches.

Afin de compléter l'étude et se replacer dans le contexte industriel et opérationnel de la thèse, il nous a paru important de réaliser des expérimentations de l'ensemble de l'algorithme sur des images complètes pour évaluer les performances et le temps de traitement. Les résultats seront présentés dans le **chapitre 5**.

Enfin nous conclurons sur cette étude et proposerons des perspectives sur la suite des travaux.

Chapitre 1

La problématique

1.1 Introduction

Dans ce chapitre, nous présenterons le contexte industriel dans lequel s'insère cette étude (partie 1.2). En effet, celui-ci conditionnera de nombreux choix. En particulier pour les contraintes de temps de calcul, de performances attendues et de gestion des erreurs.

Puis, nous ferons une description des différentes données disponibles pour répondre à la problématique de la surveillance maritime (partie 1.3).

Après avoir spécifié les données que nous utiliserons, nous présenterons le contenu des scènes étudiées (partie 1.4) et nous exposerons notre problématique en présentant les travaux déjà réalisés dans le domaine tout en décrivant notre propre démarche (partie 1.5). Nous présenterons ensuite la première partie de notre travail qui a été de constituer des bases d'images utilisées pour caractériser et tester nos algorithmes (partie 1.6).

1.2 Le contexte

La détection automatique de bateaux a été intensivement étudiée depuis de nombreuses années et surtout depuis la mise en orbite des satellites radar [18, 65]. Plusieurs projets français et européens se sont intéressés à ce problème de détection de bateaux à partir d'images satellitaires et nous allons présenter brièvement quelques projets récents.

1.2.1 IBIS (*Implementation of Boat Information System*)

Le projet IBIS a débuté en 2003 et s'est achevé en 2005. Il a été financé par le Ministère de la Recherche dans le cadre du réseau Terre et Espace et mené par Spot Image en partenariat avec l'IRD (Institut de Recherche pour le Développement) et l'Ifremer (Institut Français de Recherche pour l'Exploitation de la Mer). L'objectif du projet était de mener une étude sur l'utilisation des données satellitaires hautes et basses résolutions optiques et radars pour l'aide à la surveillance maritime, et de définir des méthodes et des outils qui répondent à cette problématique. Si IBIS s'est attaché à répondre à la problématique de l'aide à la surveillance satellitaire de la pêche en Guyane Française, les résultats de cette étude sont applicables à d'autres zones géographiques.

Le rapport final du projet IBIS [51] a démontré les atouts de l'imagerie satellitaire optique pour la surveillance des zones maritimes. Ce projet a permis de définir un système adapté aux besoins des acteurs de la surveillance maritime.

1.2.2 DECLIMS (*Detection, Classification and Identification of Maritime traffic from Space*)

Le projet DECLIMS [12] a débuté en 2003 et s'est achevé en 2006. C'est un projet européen auquel ont participé 16 partenaires institutionnels et industriels. La mise en place de ce projet vient de la prise de conscience de la fragilité des écosystèmes marins et du fait que les ressources halieutiques ne sont pas inépuisables. Ces constatations ont emmené les autorités civiles et les organismes de gestion de la pêche à demander la mise en place de moyens de contrôle du trafic maritime et en particulier par imagerie satellitaire en vue de limiter la pêche illicite.

Dans le cadre de ce projet, de nombreuses publications ont été réalisées en particulier sur des algorithmes de détection de bateaux dans l'imagerie satellitaire radar [32, 63, 64, 58, 62, 61, 70], mais aussi sur des études sur la signature des bateaux dans différents types d'images satellitaires[29, 30]. Un état des avancées réalisées durant le projet DECLIMS est présenté dans [31, 28]. Cet état met en évidence une bonne utilisation des données satellitaires radars, avec de nombreuses techniques disponibles et de nombreux produits commerciaux. Il met également en évidence la faible utilisation des données satellitaires optiques, avec peu de techniques développées.

1.2.3 LIMES (*Land and Sea Monitoring for European Security*)

Le projet LIMES [41] a débuté en décembre 2006 et s'est achevé en juin 2010. C'est un projet cofinancé par la Commission Européenne et il a été mené par un consortium de 50 partenaires européens dirigé par Telespazio. Ce projet s'intéresse au développement

d'applications et de services relatifs à la sécurité à l'aide de systèmes d'observation de la Terre et de données images satellitaires. Ce projet vise trois types d'applications :

- la surveillance et protection du trafic maritime,
- la surveillance des frontières terrestres et des infrastructures critiques (centrales nucléaires, pipelines),
- l'aide humanitaire, les conséquences des catastrophes naturelles (tremblements de terre, tsunamis), la reconstruction et l'aide au développement.

Ce projet venant juste de s'achever au moment de la rédaction de cette thèse, nous n'avons pas eu accès aux rapports finaux.

1.2.4 OCEANWAY

Dans la continuité des projets IBIS et LIMES, SPOT image et la société Nevantropic SAS ont lancé le projet OCEANWAY. Ce projet a consisté à développer une plate-forme multi-sources de surveillance du trafic maritime en pied d'antenne de réception des satellites SPOT 2,4,5 et ENVISAT en Guyane Française (projet SEAS-Guyane [56]).

Cette plate-forme logicielle inclut des images optiques et radars et un système d'identification des navires par des balises de type VMS (*Vessel Monitoring System*). Les balises VMS équipent tous les navires de pêches français de plus de 15 mètres. Chaque balise transmet toutes les 20 minutes une position, un cap et une vitesse ; ce qui différencie les navires en règle des navires illicites.

Une collaboration avec la marine nationale a permis la mise en place de tests en conditions opérationnelles pendant une durée de 12 mois. Cette campagne de tests s'est achevée par la démonstration réalisée dans le cadre du projet LIMES en juin 2008.

Il est à noter que l'objectif d'OCEANWAY est non pas la surveillance portuaire mais la surveillance du trafic maritime au large et en particulier la surveillance des activités de pêches.

Actuellement dans OCEANWAY, l'image est découpée en dalles. L'opérateur humain traite chacune d'elle séquentiellement. A l'issue du traitement, il fournit en ensemble fiable de détections de bateaux, avec pour chacun d'eux une indication de taille et un cap approximatif.

Le succès d'OCEANWAY a conforté la société Nevantropic dans cette voie et en partenariat avec l'Université des Antilles et la Guyane a lancé une étude sur la faisabilité d'une automatisation de la détection de navires sur les images satellitaires optiques. Notre travail de recherche s'insère donc dans ce projet industriel d'automatisation de la détection et de la classification des navires sur des images satellitaires optiques à haute résolution.

A court terme, le but est de fournir une assistance pour l'opérateur humain en lui proposant des pré-détections qu'il pourra valider ou non. La méthode devra donc fournir des pré-détections fiables (avec pour chacune la taille et le cap), en nombre suffisamment faible pour être exploitables et ainsi réduire le temps de traitement d'une image. Mais l'objectif final, est une automatisation totale du processus de détection des bateaux.

1.3 Les données

Les données jouent un rôle primordial dans la surveillance maritime car elles conditionnent :

- les conditions de surveillance (nuages, nuit),
- ce que l'on distingue en fonction de leur résolution,
- leur capacité de discrimination entre différents types de matériaux.

Les données satellitaires disponibles sont de deux types :

- les images optiques : panchromatiques ou multibandes,
- les images radars ou RSO (Radar à Synthèse d'Ouverture ou SAR pour *Synthetic Aperture Radar*).

1.3.1 Les images optiques

Ces images sont calculées à partir des données acquises par un capteur passif qui va enregistrer l'énergie solaire réfléchi par la surface terrestre. La nuit, ou en présence de nuages, ce capteur devient aveugle.

On distingue deux type d'images : panchromatique ou multispectrale (multibandes). Dans l'imagerie panchromatique, le capteur mesure la réflectance dans une large bande électromagnétique alors que dans l'imagerie multispectrale, le capteur mesure la réflectance dans de nombreuses bandes.

Ces images présentent un aspect assez semblable à celui d'une photographie et sont donc bien adaptées à une interprétation visuelle. Dans la table 1.1, nous présentons les caractéristiques de quelques satellites utilisant un capteur passif. Les caractéristiques sont la bande spectrale, la résolution et la fauchée qui nous donnent la taille de la scène observée.

L'imagerie panchromatique possède l'avantage d'avoir une haute résolution ce qui est intéressant pour détecter des petits navires. L'imagerie multispectrale possède, elle, l'avantage de pouvoir facilement distinguer les zones terrestres et les zones nuageuses, en contre partie sa résolution est moins bonne.

| Satellite | Bande spectrale | Résolution | Fauchée |
|-------------|---|----------------------|---------|
| SPOT 1,2,3 | Panchromatique Multispectral | 10 m 20 m | 60 km |
| SPOT 4 | Panchromatique Multispectral | 10 m 20 m | 60 km |
| SPOT 5 | Panchromatique Multispectral | 5 m et 2,5 m 10 m | 60 km |
| IKONOS 2 | Panchromatique Multispectral | 0,8 m 3,2 m | 11 km |
| QUICKBIRD 2 | Panchromatique Multispectral | 0,61 m 2,4 m | 16,5 km |
| landsat 5 | Multispectral IR thermique | 30 m 120 m | 185 km |
| landsat 7 | Panchromatique Multispectral IR thermique | 15 m 30 m 60 m | 185 Km |

TAB. 1.1 – *Caractéristiques de quelques satellites optiques.*

1.3.2 Les images SAR

Ce type d'image est produit par des satellites utilisant un capteur actif qui produit un rayonnement électromagnétique qu'ils vont diriger vers la Terre et enregistrer la part de rayonnement réfléchi par celle-ci [44]. Ce type de capteur réalise des prises de vue de jour comme de nuit et dans toutes les conditions météorologiques, y compris avec une grande couverture nuageuse. Contrairement aux images optiques, l'interprétation de ces images n'est pas simple pour l'œil humain. Dans la table 1.2 nous présentons les caractéristiques de quelques satellites utilisant un capteur actif. Les caractéristiques sont la résolution et la fauchée.

| Satellite | Résolution | Fauchée |
|--------------|--------------------|--------------------|
| RADARSAT-1 | entre 8 m et 100 m | entre 50 et 500 km |
| ENVISAT/ASAR | 30/150/1000 m | 100/400 km |

TAB. 1.2 – *Caractéristiques de quelques satellites radars.*

1.3.3 Les images utilisées

Dans le cadre de son projet OCEANWAY, notre partenaire Nevantropic traite des images satellitaires optiques haute résolution de type SPOT 5 panchromatiques de résolution 5m. Un exemple d'image est présenté dans la figure 1.1. Elles sont donc les seules images disponibles pour notre étude.

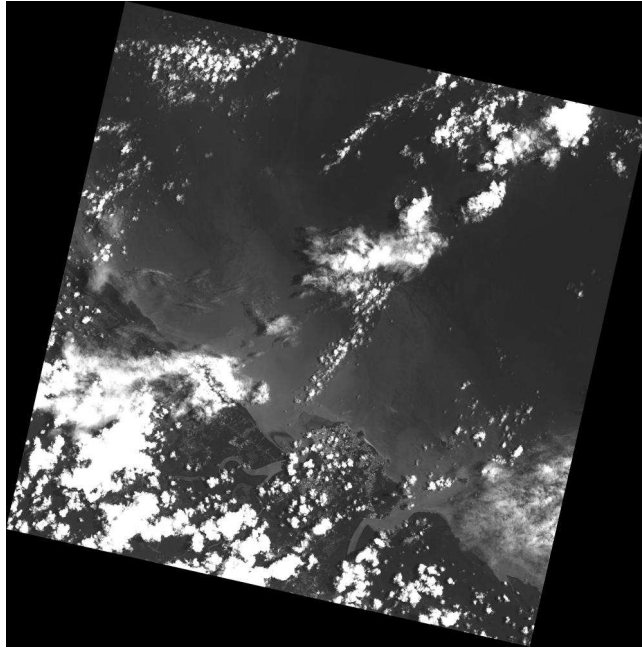


FIG. 1.1 – Exemple d’image SPOT 5 panchromatique de résolution 5m.

Le capteur du satellite SPOT 5 dispose d’une bande spectrale panchromatique qui produit des images en niveaux de gris.

Avec une résolution de 5 mètres, ce type d’image est destiné aux applications nécessitant une bonne précision ; ce qui est le cas pour la surveillance des activités de pêche où les bateaux font entre 15m et 30m.

Sa fauchée de 60 km permet de couvrir de larges zones ce qui est également bien adapté au problème de surveillance maritime. Ces images ont une taille d’environ 14000×14000 pixels ($1,96 \times 10^8$ pixels) ce qui représente une quantité importante d’information à traiter (environ 200 Mo).

De plus ces images sont géoréférencées, la position des côtes peut être connue en utilisant la cartographie côtière ou des SIG (Système d’Information Géographique).

1.4 Le contenu des scènes

Avant de développer une stratégie de détection, il est primordial d’observer le contenu des scènes et la façon dont chaque élément apparaît dans les images retenues. Cette observation va déterminer ce qui nécessitera un intérêt particulier.

Pour présenter le contenu des scènes étudiées, nous distinguerons d’une part les objets

recherchés c'est-à-dire les mobiles en mer et d'autre part les éléments environnementaux à savoir les zones de terre, les zones de mer et les nuages.

1.4.1 Les mobiles en mer

Il existe un grand nombre de types de bateaux. Les Affaires Maritimes possèdent une classification très fine de ces navires. Toutefois la résolution des images utilisées (5m) ne permet pas forcément de distinguer les différents types de bateaux. En effet, il peut être assez difficile de faire la distinction entre un bateau de pêche et un bateau de plaisance de même taille. On nous a demandé dans un premier temps de fonder notre classification des navires sur un critère de taille :

- les gros navires ou bateaux de commerce : porte-conteneurs, tankers, navires de guerre (porte-avions, frégate), minéraliers, pétroliers (figure 1.2),
- les petits navires : bateaux de pêches, navires de plaisance, navires de servitude (figure 1.3).

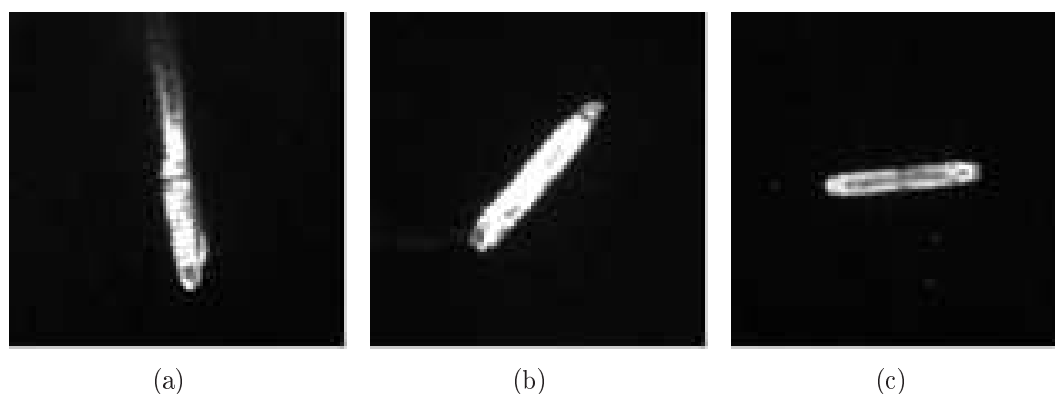


FIG. 1.2 – Quelques exemples de gros navires dans des images SPOT 5 panchromatiques de résolution 5m. L'image (a) représente un porte-conteneur en mouvement, (b) représente un autre porte-conteneur à l'arrêt et dans (c) nous sommes en présence d'un pétrolier.

Dans les images, les petits navires sont de forme très variable, puisque par exemple les bateaux de pêche se présentent sous des formes différentes lorsqu'ils sont en activité de pêche avec leurs filets déployés (figure 1.4(b)) ou non (figure 1.4(a)). Par ailleurs, l'intensité lumineuse réfléchiée par les bateaux peut être aussi variable. D'après les experts d'OCEANWAY, les bateaux peuvent apparaître clairs ou sombres par rapport au fond. Toutefois dans les images dont nous disposons, nous n'avons jamais observé de navires plus sombres que le fond, mais nous en tiendrons quand même compte dans nos choix de méthode.

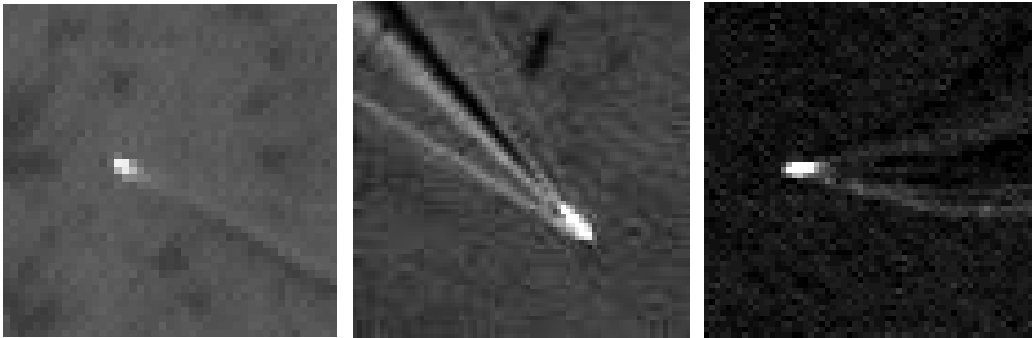


FIG. 1.3 - Quelques exemples de petits navires dans des images SPOT 5 panchromatiques de résolution 5m.

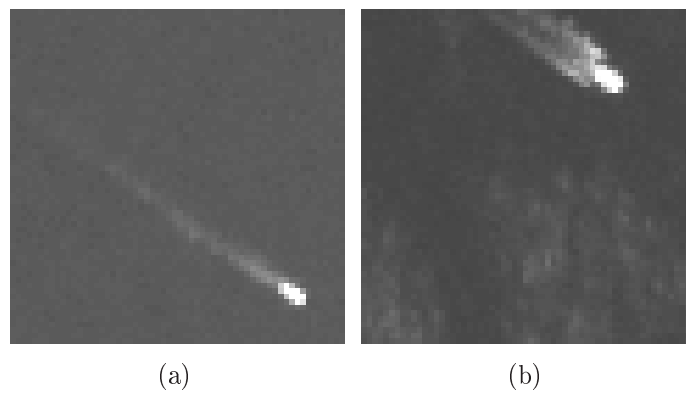


FIG. 1.4 - Exemple d'images d'un crevettier. Dans l'image (a) le crevettier est en navigation simple avec les filets rangés et dans l'image (b) il est en activité de pêche avec ses filets déployés.

1.4.2 Les sillages

Les navires en mouvement laissent une trace derrière eux : le sillage [65]. Le sillage est la turbulence créée dans l'eau par le bateau. Il reste dans la trajectoire du bateau et disparaît très rapidement. Les vagues d'étrave sont un ensemble de vagues qui se propagent dans l'eau de part et d'autre du sillage en dissipant très peu d'énergie ; les vagues d'étrave se propagent très loin et ne restent pas dans le sillage, mais s'en éloignent. Une illustration de sillage est présentée dans la figure 1.5.

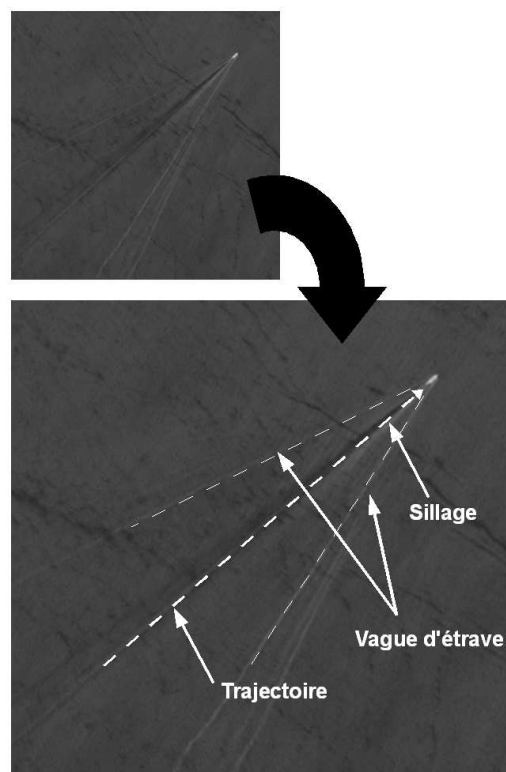


FIG. 1.5 – Description du sillage d'un navire. Il est composé de deux éléments : le sillage principal et les vagues d'étrave.

La caractérisation du sillage permettrait éventuellement de retrouver le cap et la vitesse du bateau, en couplant les observations sur l'image avec des modèles physiques des sillages. Le problème majeur est que ces sillages n'apparaissent pas toujours sous une forme intéressante dans nos images. La totalité du sillage n'est pas toujours très bien visible sur les images comme dans les figures 1.6(b) et 1.6(c) où nous ne distinguons pas les vagues d'étrave.

On peut également noter que leur intensité est variable. Dans la figure 1.6(b) nous avons un sillage sombre alors que dans la figure 1.6(c) nous avons un sillage plutôt clair.

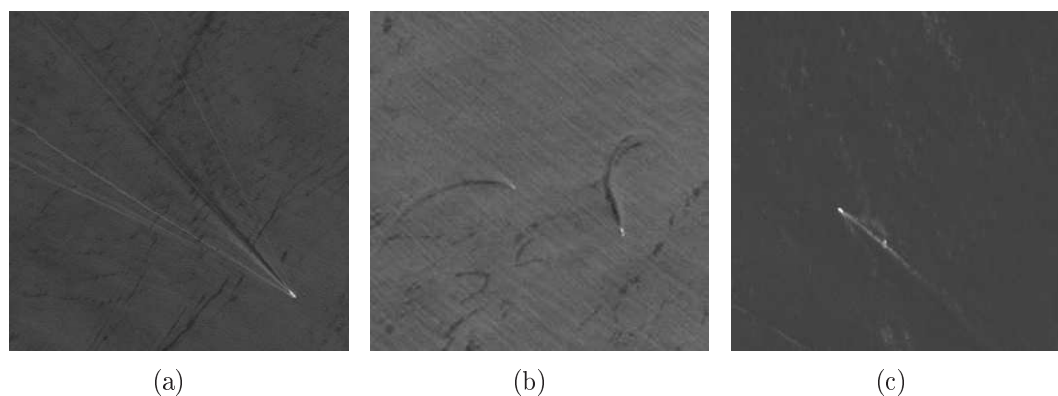


FIG. 1.6 – Quelques exemples de sillages dans des images SPOT 5 panchromatiques de résolution 5m. Dans l'image (a), on distingue bien le sillage et les vagues d'étrave, ce qui n'est pas le cas dans les images (b) et (c). Le sillage peut être d'intensité variable : dans les images (a) et (c), il est clair alors que dans (b) il est foncé.

1.4.3 La terre

Les images peuvent contenir des zones de terre : les côtes comme dans la figure 1.7(b) ou de petites îles comme dans la figure 1.7(a).

Les zones de terre ne faisant pas partie de notre étude et étant donné que les images opérationnelles sont géoréférencées, il est simple de masquer les zones de terre et d'éliminer ainsi le bruit qu'elles génèrent dans les traitements.

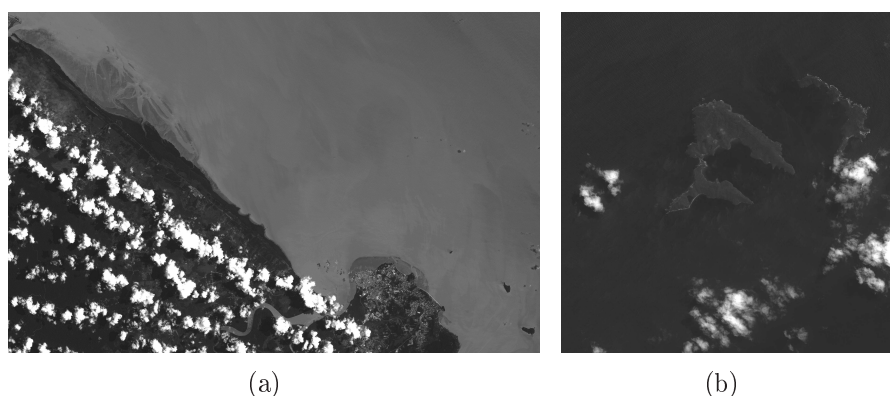


FIG. 1.7 – Exemples de zones de terre.

1.4.4 Les zones de mer

Les zones de mer peuvent apparaître sous différentes formes en fonction des conditions météorologiques (houle et vent), mais aussi en fonction des conditions d'acquisition. Il peut ainsi présenter des textures différentes et des intensités variables. La figure 1.8 illustre différents types de fond.

Dans le cas de forte houle, certaines crêtes de vagues peuvent être confondues avec des petits bateaux, ce qui risque de nous poser des problèmes de fausses détections.

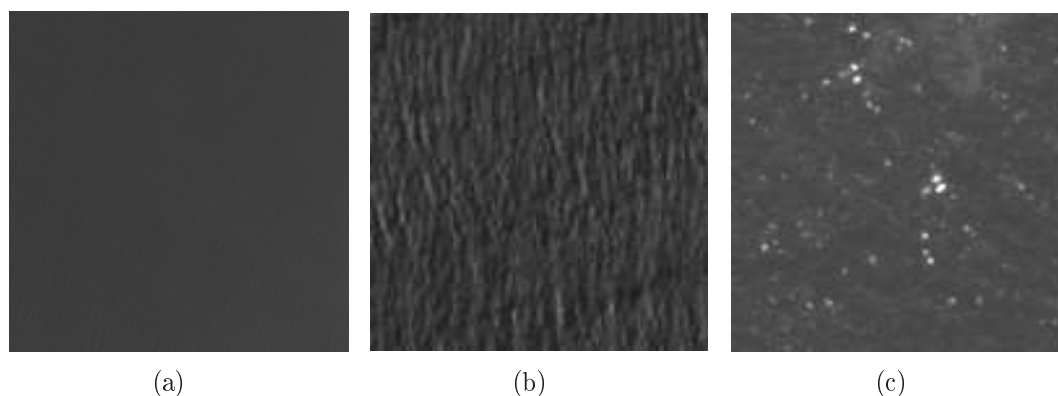


FIG. 1.8 – Quelques exemples de zones de mer. Dans l'image (a), la mer est homogène. Dans (b), la mer présente une texture caractéristique de la présence de houle. (c) est caractéristique de mauvaises conditions météorologiques avec de la houle et du vent qui forment des crêtes de vagues, ces dernières peuvent être parfois confondues avec des petits navires.

1.4.5 Les nuages

Dans les images nous observons également la présence de nuages. Les nuages sont classiquement classés selon dix types de nuages : les cirrus, les cirrocumulus, les cirrostratus, les altocumulus, les altostratus, les nimbostratus, les stratocumulus, les stratus, les cumulus et les cumulonimbus. Dans la figure 1.9 nous présentons quelques exemples de nuages présents dans nos images. Chaque type de nuage est caractérisé par sa taille, sa forme, sa texture, sa transparence, son opacité et ses couleurs qui varient en fonction des constituants et des conditions atmosphériques.

Les nuages peuvent être de taille, de forme et d'aspect variables (les contours sont plus ou moins marqués, parfois diffus comme dans la figure 1.9(b)) et certains petits nuages compacts peuvent se confondre avec des bateaux. Comme pour le fond, ils peuvent présenter des textures différentes et des intensités variables comme cela est illustré dans la figure 1.9.

Certains bateaux peuvent être distingués à travers des nuages diffus comme dans la figure 1.10.

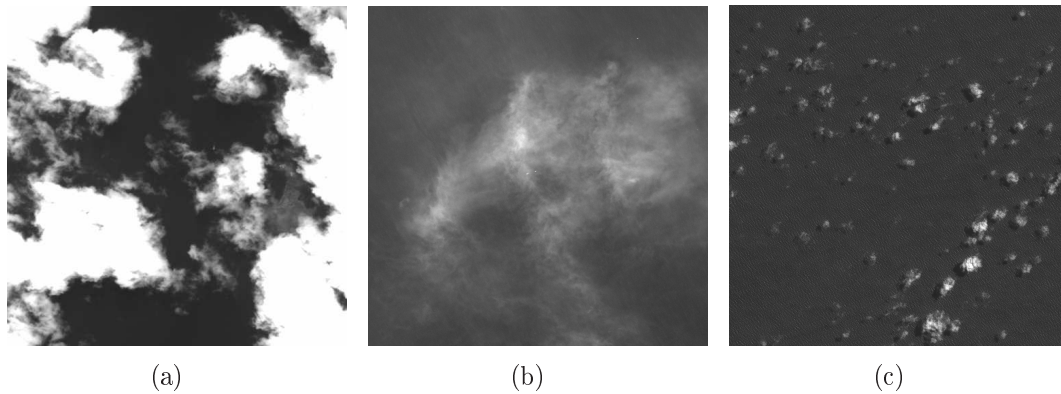


FIG. 1.9 – Quelques exemples de nuages. Dans (a) nous avons des cumulonimbus qui sont des grosses masses nuageuses. Dans (b) nous avons un cirrus qui a un aspect plus diffus. Dans (c) nous avons des petits cumulus qui sont des petits nuages compacts pouvant être parfois confondus avec un navire.

1.5 Notre problématique

Depuis de nombreuses années, la détection automatique de bateaux a été intensivement étudiée. Elle est le plus souvent appliquée sur des images de type SAR (Synthetic Aperture Radar). Dans ce contexte, de nombreux algorithmes ont été développés pour ce type d'images [60, 42] et le lecteur pourra aussi se référer à un état de l'art du domaine dans [11].

En revanche, les images satellitaires optiques sont encore peu exploitées dans ce domaine. Par conséquent les travaux sont encore assez peu nombreux. Toutefois, une méthode est présentée par Corbane et al dans [8]. Elle se décompose en trois phases :

1. une segmentation qui est fondée sur l'utilisation d'un filtre qui va rechercher des pixels brillants par rapport à un fond puis une segmentation par croissance de région,
2. une extraction de caractéristiques,
3. une classification qui utilise des réseaux de neurones.

1.5.1 Nos objectifs

Notre objectif principal est d'étudier la possibilité de définir une méthode de détection de cibles dans une image satellitaire optique de type SPOT 5 - panchromatique de résolution 5m. Les cibles sont de forme et d'intensité variées, les zones maritimes présentent des états de mer variables et sont éventuellement parsemées de nuages.

Nous voulons fournir à partir d'une image la liste des navires présents, avec pour chacun des navires ses caractéristiques : son type, sa position dans l'image, sa taille et sa direction de déplacement quand cela est possible.

Nous avons vu qu'à terme, le but est de fournir une assistance à un opérateur humain en lui proposant des pré-détections qu'il pourra valider ou non. Dans cette optique nos objectifs sont :

- de détecter le plus de bateaux possible, l'idéal étant de tous les détecter,
- d'avoir le moins de fausses détections possible,
- d'avoir un temps de traitement inférieur à celui de l'opérateur humain (environ 45 minutes).

1.5.2 Notre démarche

Pour ce problème, plusieurs approches sont envisageables. Une première approche pourrait envisager d'associer à chaque type de bateau recherché un motif et de balayer l'image à la recherche de ce motif. Cette approche pose plusieurs problèmes. Le premier est la définition des motifs. En effet, nous avons vu dans la partie 1.4.1 que les mobiles en mer pouvaient avoir des formes assez variables et de ce fait, les motifs risquent d'être assez nombreux. D'autre part, pour chaque motif, il faut tester l'ensemble des orientations possibles. Cela risque de poser un problème de temps de calcul étant donné le nombre de motifs à tester et la taille des images (200Mo). Or, précédemment, nous avons vu que le temps de traitement est une des contraintes principales de notre problématique. C'est pour ces raisons que nous nous sommes orientés vers une autre approche.

Nous avons observé le photo-interprète du projet OCEANWAY traiter une image pour tenter de déterminer s'il était possible de formaliser le processus de recherche de mobiles en mer. Il va dans un premier temps tenter de détecter des objets qui se distinguent du fond, puis pour chaque objet faire une analyse minutieuse de l'objet en zoomant, en jouant avec le contraste et en prenant en compte l'environnement de l'objet. Ce processus est difficile à formaliser. Mais dans notre méthode de détection de cibles nous allons reprendre quelques concepts : détecter des zones qui se distinguent du fond puis analyser ces zones pour confirmer ou non la présence d'un bateau.

Les zones de terre seront masquées comme cela a été préconisé dans la partie 1.4.3.

Concernant le traitement des nuages, une approche serait de masquer les nuages. Pour cela, il faudrait réaliser un pré-traitement qui les détectent puis les masquent. La détection des nuages est un problème complexe et sa mise en place risque d'être coûteuse en temps de calcul. De plus, il existe un risque de ne pas détecter un bateau qui serait sous des nuages diffus comme dans la figure 1.10. Pour l'ensemble de ces raisons, nous avons préféré de ne pas masquer les nuages et d'intégrer leur traitement dans la partie classification (figure 1.11) de notre approche.

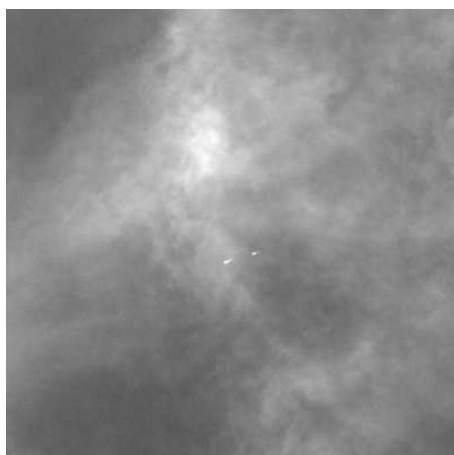


FIG. 1.10 - Exemple d'image où deux bateaux sont sous des cirrus.

Nous pouvons ainsi décomposer le processus de traitement en différentes phases :

1. l'application d'un masque de terre pour ne traiter que les zones de mer,
2. la pré-détection des cibles : fournit des régions d'intérêt (ROI),
3. la classification des pré-détections en trois classes : petit bateau, grand bateau et tout ce qui n'est pas un bateau ; à l'issue de cette phase nous avons la liste des bateaux détectés dans l'image.

Cette phase se décompose en trois parties :

- a la segmentation de chacune des pré-détections, ce qui fournit un ensemble de contours localisés dans l'image,
 - b l'extraction des caractéristiques de chaque contour. Pour cette étape, nous allons extraire un certain nombre de caractéristiques et déterminer celles qui sont les plus appropriées pour réaliser une classification efficace,
 - c la classification des contours,
4. la détection du sillage : pour chaque bateau détecté à l'étape précédente nous allons tenter de trouver son sillage (celui-ci n'est pas toujours visible, mais lorsqu'il est observable, il constitue une information supplémentaire pour la caractérisation des bateaux),
 5. la caractérisation des navires réalisée avec les informations fournies lors des phases précédentes.

L'ensemble du processus est illustré dans la figure 1.11.

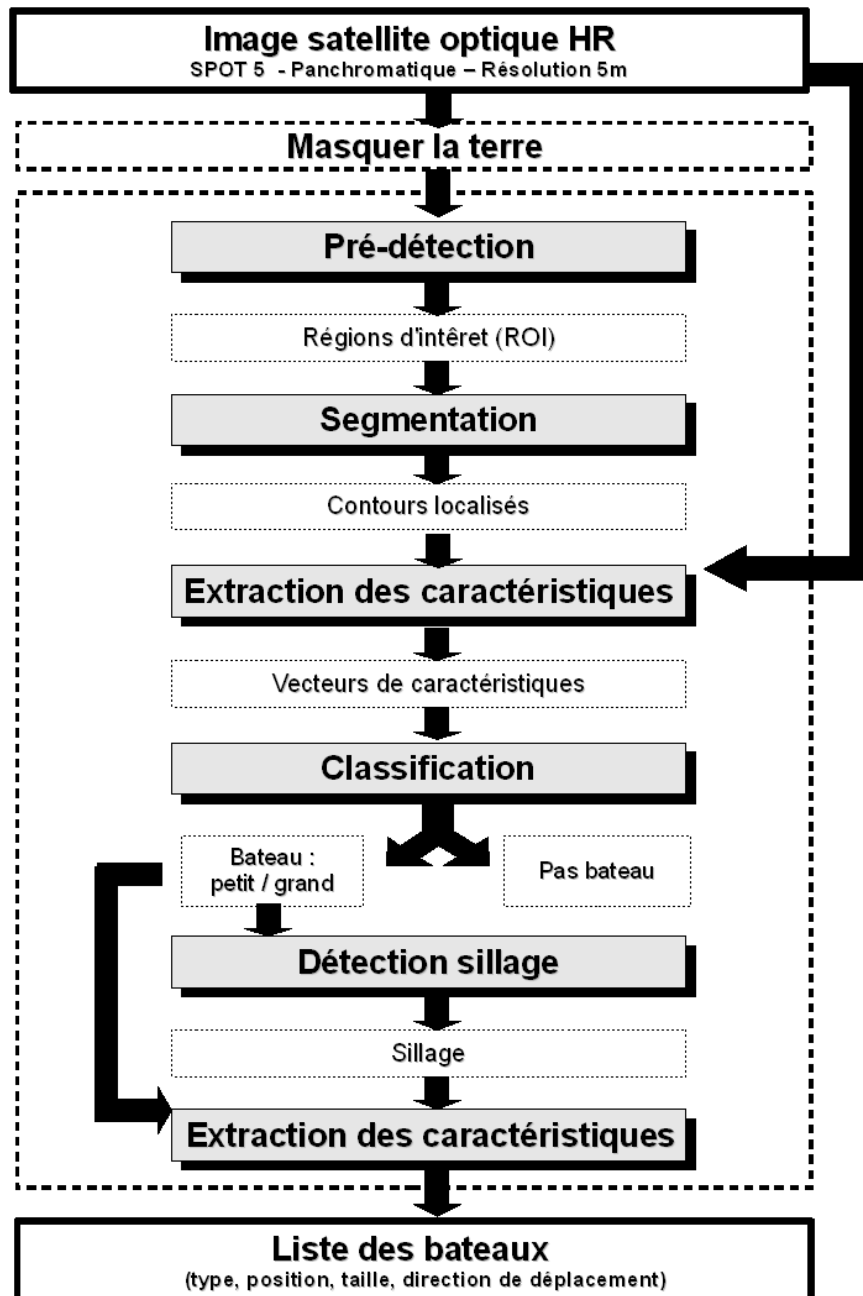


FIG. 1.11 – Description de notre démarche pour la détection de bateaux.

1.6 Les bases d'images

Pour réaliser notre étude, nous avons à notre disposition une quarantaine d'images complètes SPOT 5 panchromatiques de résolution 5m issues de la station SEAS-Guyane. Les prises de vue ont été faites sur plusieurs zones de la région Antilles-Guyane à différentes dates, dans des conditions météorologiques différentes et avec des conditions d'acquisition différentes.

Dans un premier temps, nous avons sélectionné parmi ces images un ensemble représentatif des différentes situations rencontrées en termes d'état de mer (mer calme, mer agitée) et en termes de couverture nuageuse.

Pour quelques images, nous avons à notre disposition une vérité terrain réalisée par les experts de Nevantropic durant des campagnes de tests réalisées avec la collaboration de la Marine Nationale. Ces vérités terrains fournissent un référentiel pour l'inspection visuelle des images. Pour la plupart des images, nous avons réalisé nous même une inspection visuelle minutieuse pour détecter la présence de mobiles en mer.

Ce travail a mis en évidence les différentes situations qui peuvent être rencontrées dans la pratique et ainsi de constituer trois bases d'images d'expérimentation que nous allons décrire.

1.6.1 Base d'images 1

Cette première base a pour objectif de mettre en évidence les problèmes que l'on peut rencontrer dans la pratique. Il est représentatif de différentes situations : mer calme, mer agitée et présence de nuages. Dans un souci de rapidité de traitement, nous avons choisi d'utiliser des petites images de taille 500×500 extraites des images complètes.

Cette base d'images est composée de 57 images de taille 500×500 pixels et contient 67 cibles. Nous avons constitué quatre sous-bases, classées selon les situations que nous pouvons rencontrer :

- 1.A : mer calme (illustré dans la figure 1.12(a)).
 - 1.B : mer agitée (illustré dans la figure 1.12(b)).
 - 1.C : présence de nuages (illustré dans la figure 1.12(c)).
 - 1.D : peu de contraste entre la cible et le fond (illustré dans la figure 1.12(d)).
-

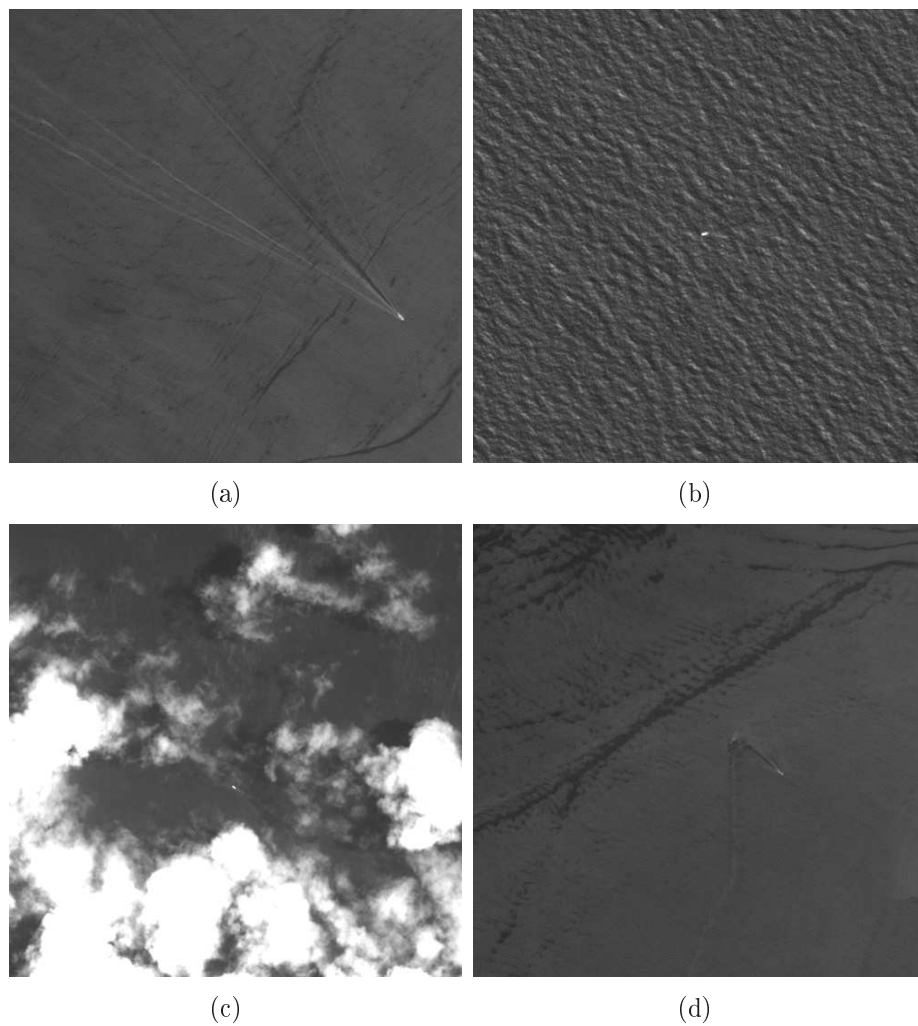


FIG. 1.12 – Exemples d'images extraites de la base d'images 1.

Dans (a) nous avons une cible et une mer calme (1.A).

Dans (b) nous avons une cible et une mer agitée (1.B).

Dans (c) nous avons une cible et des nuages (1.C).

Dans (d) nous avons un contraste faible entre la cible et le fond (1.D).

1.6.2 Base d'images 2

La seconde base est utilisée pour la phase de classification (figure 1.11), elle est constituée de 193 extraits d'images complètes et contient 224 cibles. Nous avons dans cette base d'images un panel représentatif de mobiles en mer (petits et grands navires) et des objets qui ne sont pas des navires et dont certains pourraient être confondus avec eux : crêtes de vague (figure 1.13(c) ou petits nuages (figure 1.13(d)).

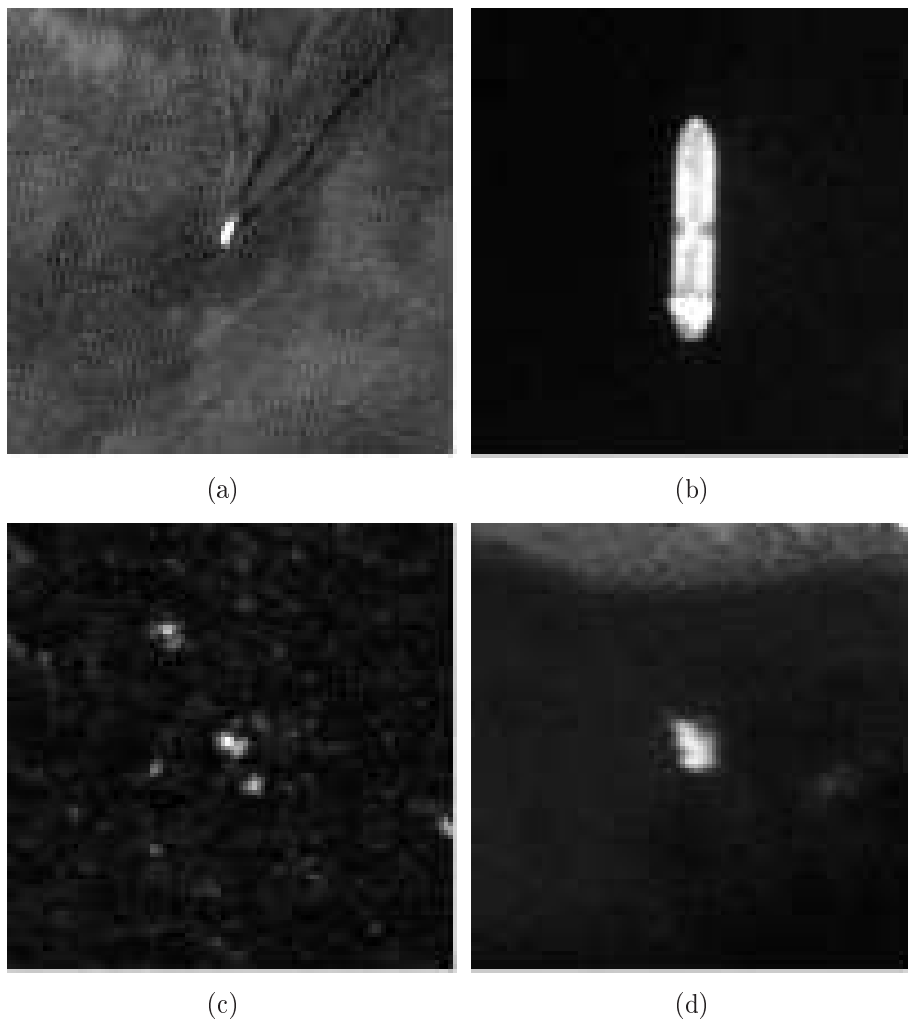


FIG. 1.13 – Exemple d'images extraites de la base d'images 2.

(a) est un exemple de petit navire.

(b) est un exemple de grand navire.

Les images (c) et (d) sont des exemples d'objets pouvant être confondus avec des navires, dans (c) nous avons des crêtes de vagues dues à de mauvaises conditions météorologiques et dans (d) nous avons un petit nuage.

1.6.3 Base d'images 3

La troisième base d'images est composée de 23 images complètes de taille 140000×140000 . Ces images contiennent différentes textures, différents états de mer et des nébulosités variées. Ce jeu d'images sera utilisé pour mesurer les performances des algorithmes en situation opérationnelle, les différentes situations y étant représentées de façon réaliste. Un exemple d'image complète est présenté dans la figure 1.14.

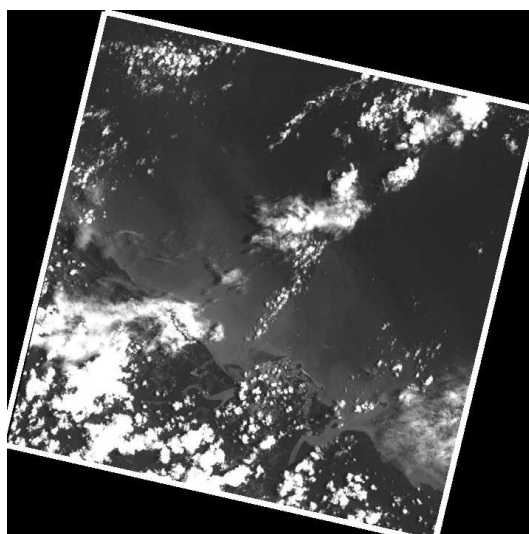


FIG. 1.14 – Exemple d'image de la base d'images 3. C'est une image complète de type SPOT 5 panchromatique de résolution 5m. La zone entourée d'un rectangle blanc correspond aux pixels utiles c'est-à-dire la zone traitée.

Le tableau 1.3 récapitule la taille et le nombre de bateaux des différentes bases d'images.

| Base d'images | Taille (pixels) | Nombre d'images | Nombre de bateaux |
|---------------|--------------------|-----------------|-------------------|
| 1.A | $6,43 \times 10^6$ | 23 | 32 |
| 1.B | $2,50 \times 10^6$ | 10 | 5 |
| 1.C | $4,08 \times 10^6$ | 14 | 16 |
| 1.D | $1,73 \times 10^6$ | 10 | 14 |
| 2 | $4,82 \times 10^7$ | 193 | 224 |
| 3 | $2,24 \times 10^9$ | 23 | 161 |

TAB. 1.3 – Récapitulatif des trois bases d'images. Pour la base d'images 3, la taille représente les pixels utiles. C'est-à-dire que nous ne prenons pas en compte les bords noirs de l'image.

1.7 Conclusion

Dans ce chapitre nous avons présenté notre problématique et dans quel contexte elle s'insère.

Après avoir présenté rapidement les différents types d'images disponibles (avec leurs avantages et leurs inconvénients) pour la surveillance maritime, nous avons spécifié lesquelles étaient disponibles pour notre étude : des images satellitaires optiques haute résolution de type SPOT 5 panchromatiques de résolution 5m.

Nous avons ensuite exposé le contenu des scènes afin d'avoir une idée précise des différents éléments présents (mobiles en mer, sillages, zones terrestres, zones maritimes, nuages) et la façon dont ils apparaissent dans les images. C'est cette étude des images qui a orienté le choix des méthodes utilisées.

Après avoir exposé les contraintes en termes de performances et de temps de calcul, nous avons présenté la démarche générale retenue, avec trois grandes étapes : la pré-détection des bateaux, la classification de ces bateaux et la détection de leur sillage.

Nous avons également présenté une étape importante de notre travail qui a été de constituer différentes bases d'images et la « vérité terrain » pour le contenu des images. Ces bases d'images seront utilisées tout au long de notre étude pour caractériser et tester les algorithmes.

Dans les chapitres suivants nous allons décrire en détail les méthodes utilisées pour chaque étape du processus de traitement décrit dans la partie 1.11.

Chapitre 2

Pré-détection des cibles

2.1 Introduction

La première étape de la méthode consiste à pré-détecter les bateaux. Nous avons vu dans le chapitre 1 que les cibles à détecter sont en priorité des bateaux de pêche dont la taille est comprise entre 15 et 30 mètres, ce qui correspond à des tailles entre 3 et 5 pixels. Dans les images, les cibles sont de forme très variable, puisque les bateaux de pêche se présentent sous des formes différentes lorsqu'ils sont en activité de pêche (filets déployés) ou non. Il est donc délicat de définir un modèle de forme pour les bateaux recherchés. Par ailleurs, l'intensité lumineuse captée par SPOT 5 pour ces bateaux est variable en fonction des caractéristiques de prise de vue et des conditions d'illumination.

Compte tenu de ce qui a été dit, les pistes qui s'offrent à nous sont relativement peu nombreuses. Un groupe de méthodes évidentes exploitant l'information de taille est celui de la détection par ondelettes [60, 47]. Néanmoins, les cibles étant de taille très réduite, ceci reviendrait à ne considérer que l'échelle de résolution la plus haute (et donc à utiliser l'équivalent d'un filtre linéaire) ou à exploiter l'information multi-échelle, par suivi d'extremums entre les résolutions [26] ce qui pose de nombreux problèmes de mise en correspondance.

L'utilisation de techniques de morphologie mathématique [35] semblerait prometteuse pour sonder l'image en utilisant un élément structurant de la taille des cibles recherchées. En particulier, une érosion devrait faire disparaître les cibles, ce qui constitue un problème dual du notre. Cependant, la mise au point de la séquence d'opérateurs morphologiques étant grandement empirique nous avons préféré nous orienter vers d'autres méthodes.

Une méthode est proposée par Corbane et al dans [8]. Elle consiste en une segmentation basée sur l'utilisation d'un filtre qui va rechercher des pixels brillants par rapport à un fond, puis en une segmentation par croissance de région. Cette méthode de recherche de

pixels brillants dans l'image ne prend en compte ni des critères de forme ni le fait que la cible n'est pas forcément plus claire que le fond, or d'après les experts de notre partenaire industriel cela peut être possible.

Dans notre approche, nous avons voulu ajouter cette notion de forme. Ceci permet d'éliminer une partie des fausses alarmes dues à la présence de nuages. Nous avons également voulu prendre en compte le fait que la cible n'est pas forcément plus claire que le fond. La méthode que nous décrivons est basée sur la théorie de la décision bayésienne. Cette méthode bien connue en imagerie radar a été utilisée avec succès pour la détection de contours dans des images SAR [49, 50] ainsi que pour la détection de bateaux avec des données polarimétriques RSO [43].

Notre contribution a consisté à modifier le modèle initial de la méthode, ce qui a amélioré les performances en terme de probabilité de détection et de probabilité de fausse alarme.

La méthode qui est à l'origine adaptée à la détection des petites cibles permet toutefois, comme nous le verrons, de détecter également des grands bateaux.

Dans ce chapitre, nous présenterons dans la partie 2.2 la méthode de pré-détection utilisée, puis nous discuterons dans la partie 2.3 le choix de notre modèle. Nous présenterons ensuite dans la partie 2.4 les résultats des tests sur la base d'images 1. Puis nous illustrerons par quelques exemples dans la partie 2.4.2, pour enfin conclure dans la partie 2.5.

2.2 Méthode de pré-détection

2.2.1 Principe

Cette méthode s'appuie sur la théorie bayésienne de la décision et consiste à tester statistiquement deux hypothèses pour choisir la plus vraisemblable. Dans notre cas, il s'agit de déterminer si une cible est présente ou non.

En chaque point de l'image, nous allons déterminer s'il existe une cible à cette position ou non. Nous utilisons pour cela une fenêtre glissante F_{p_i} qui se déplace en chaque pixel p_i de l'image. Dans cette fenêtre d'analyse, nous cherchons à effectuer un test qui départage

les deux hypothèses suivantes :

$$\left\{ \begin{array}{l} H_0 : \text{la fen\^etre ne couvre que du fond sans aucune cible,} \\ \quad \text{la r\^egion correspondante est not\^ee } A. \\ H_1 : \text{la cible est au centre de la fen\^etre,} \\ \quad \text{la r\^egion correspondante est not\^ee } w \text{ et} \\ \quad \text{la r\^egion not\^ee } \bar{w} \text{ correspond \^a l'ext\^erieur de la cible.} \end{array} \right.$$

Les deux hypoth\^eses H_0 et H_1 sont illustr\^ees dans la figure 2.1.

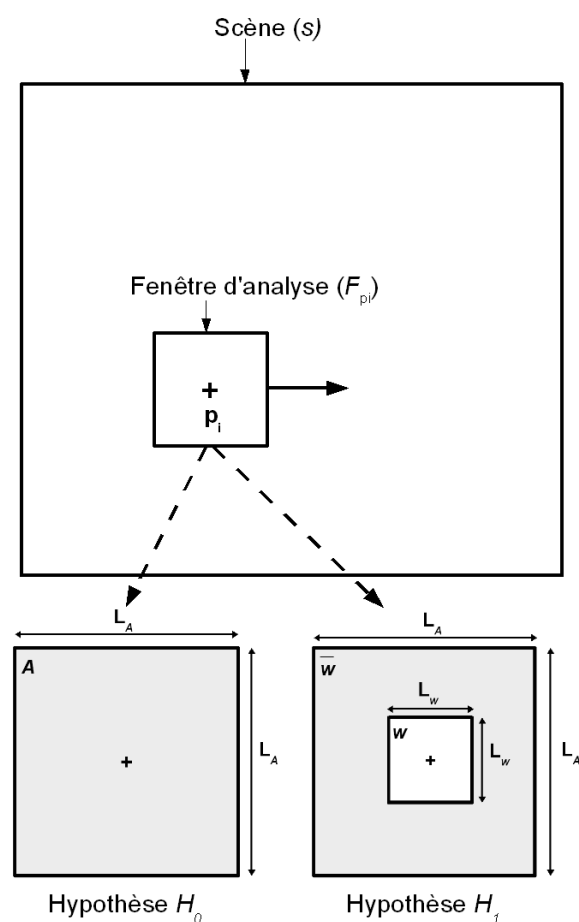


FIG. 2.1 – Sc\^ene (s) sous les deux hypoth\^eses H_0 et H_1 . La fen\^etre d'analyse est un carr\^e de taille $N_A = L_A \times L_A$ pixels, la cible recherch\^ee (w) est consid\^er\^ee \^etre un carr\^e de taille $N_w = L_w \times L_w$ et la r\^egion \bar{w} est de taille $N_{\bar{w}} = N_A - N_w$. Le choix d'un carr\^e plut\^ot qu'une autre forme rel\^eve d'un souci de simplicit\^e d'impl\^ementation.

Nous nous int\^eressons aux probabilit\^es suivantes :

$$\left\{ \begin{array}{l} P(H_0|s) : \text{La probabilit\^e de l'hypoth\^ese } H_0, \text{ connaissant la sc\^ene } (s). \\ P(H_1|s) : \text{La probabilit\^e de l'hypoth\^ese } H_1, \text{ connaissant la sc\^ene } (s). \end{array} \right.$$

Nous définissons le rapport des probabilités de la façon suivante :

$$R = \frac{P(H_1|s)}{P(H_0|s)} = \frac{P(s|H_1)}{P(s|H_0)} \times \frac{P(H_1)}{P(H_0)} \quad (2.1)$$

où $P(H_0)$ est la probabilité que la fenêtre F_{pi} ne couvre que du fond, indépendamment de la scène observée et $P(H_1)$ est la probabilité que de la cible soit au centre de la fenêtre F_{pi} , indépendamment de la scène observée.

La prise de décision est donnée par les règles suivantes :

$$\begin{cases} R > 1 \Rightarrow \text{détection d'une cible } (H_1) \\ R < 1 \Rightarrow \text{non détection d'une cible } (H_0) \end{cases} \quad (2.2)$$

La grandeur $P(H_0|s)$ (respectivement $P(H_1|s)$) est nommée vraisemblance de l'hypothèse H_0 (respectivement H_1).

Nous définissons le rapport de vraisemblance de la manière suivante :

$$\Lambda = \frac{P(s|H_1)}{P(s|H_0)} \quad (2.3)$$

La prise de décision donnée par l'équation 2.2 est équivalente à la suivante :

$$\begin{cases} \Lambda > \frac{P(H_0)}{P(H_1)} \Rightarrow \text{détection d'une cible} \\ \Lambda < \frac{P(H_0)}{P(H_1)} \Rightarrow \text{non détection d'une cible} \end{cases} \quad (2.4)$$

2.2.2 Règle de décision de Neyman-Pearson

Nous introduisons deux mesures de performance :

- La **probabilité de détection** PD : probabilité de décider qu'une cible est présente dans la fenêtre d'analyse (H_1) lorsqu'elle est réellement présente.
- La **probabilité de fausse alarme** PFA : probabilité de décider qu'une cible est présente dans la fenêtre d'analyse (H_1) alors qu'il n'y a que du fond (H_0).

La règle de décision de Neyman-Pearson [55] consiste à fixer un niveau de risque α acceptable pour la probabilité de fausse alarme (PFA) et à maximiser la probabilité de détection (PD) pour cette probabilité de fausse alarme. D'après le lemme de Neyman-Pearson, l'équation 2.3 réalise cette optimisation pour toute PFA choisie.

Le seuil de décision est maintenant fixé à partir d'un taux d'erreur acceptable et non plus sur la connaissance des probabilités $P(H_0)$ et $P(H_1)$ qui sont difficiles à évaluer.

La prise de décision est désormais donnée par l'équation suivante :

$$\begin{cases} \Lambda > \tau \Rightarrow \text{détection d'une cible} \\ \Lambda < \tau \Rightarrow \text{non détection d'une cible} \end{cases} \quad (2.5)$$

Nous prenons le logarithme du rapport de vraisemblance, ce qui est équivalent à prendre la différence des logvraisemblances, nous obtenons le résultat suivant :

$$\log\Lambda = \log(P(s|H_1)) - \log(P(s|H_0)) \quad (2.6)$$

La prise de décision définie par l'équation 2.5 est équivalente à la suivante :

$$\begin{cases} \log\Lambda > \tau' \Rightarrow \text{détection d'une cible} \\ \log\Lambda < \tau' \Rightarrow \text{non détection d'une cible} \end{cases} \quad (2.7)$$

où $\tau' = \log\tau$. Par la suite il nous faudra fixer le seuil τ' .

2.2.3 Modèle utilisé

Pour calculer ces quantités, il nous faut un modèle statistique décrivant la distribution des intensités des pixels de l'image. Grâce à l'observation de divers histogrammes pour le fond (figure 2.2), nous pouvons raisonnablement faire l'hypothèse que nous sommes en présence d'une distribution gaussienne pour le fond.

De plus, nous utiliserons l'hypothèse simplificatrice que les intensités des pixels sont indépendantes et uniformément distribuées (bruit blanc). Le lecteur pourra noter que si la fenêtre d'analyse est petite, cette hypothèse est peu restrictive pour les fonds inhomogènes de forte longueur de corrélation. En revanche, il est clair que ce modèle est théoriquement peu adapté lorsque le fond de la fenêtre présente une texture visible. Néanmoins, comme nous le verrons plus loin, cette hypothèse permet d'obtenir un modèle simple et très robuste. Pour la cible, nous choisirons le même modèle, ce qui est peu restrictif au vu des tailles concernées.

Le modèle retenu est donc le suivant :

- Chaque pixel de la cible suit une distribution gaussienne blanche de paramètres inconnus.
 - Chaque pixel du fond suit une distribution gaussienne blanche de paramètres inconnus.
-

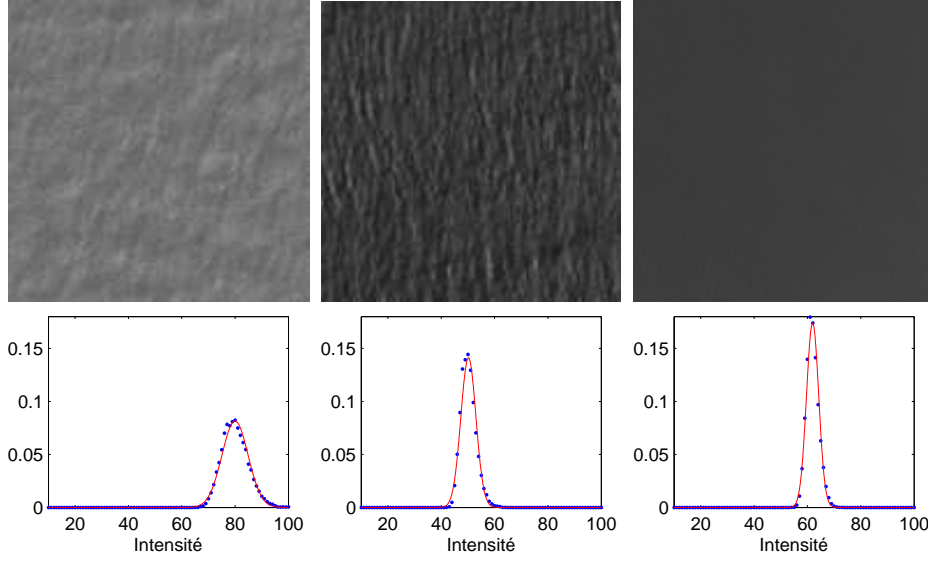


FIG. 2.2 – Exemples de texture de mer et histogrammes normalisés associés. La courbe en trait continu représente la distribution gaussienne la plus proche.

En utilisant ce modèle statistique on a :

- Pour un pixel s_i de la région R (ici R représente les différentes régions : $w/\bar{w}/A$) (figure 2.1) :

$$P(s_i) = \frac{1}{\sqrt{2\pi\sigma_R^2}} e^{-\frac{(s_i - m_R)^2}{2\sigma_R^2}} \quad (2.8)$$

où σ_R^2 est la variance de la région R et m_R est l'espérance de la région R.

- Les pixels étant indépendants, nous en déduisons que pour une région R :

$$P(R) = \prod_{i \in R} \frac{1}{\sqrt{2\pi\sigma_R^2}} e^{-\frac{(s_i - m_R)^2}{2\sigma_R^2}} \quad (2.9)$$

- Nous notons $L(R)$ le logarithme de la quantité précédente :

$$L(R) = -\frac{1}{2} \sum_{s_i \in R} \left[\log(2\pi\sigma_R^2) + \frac{(s_i - m_R)^2}{\sigma_R^2} \right] \quad (2.10)$$

Nous calculons les quantités $\log(P(s|H_0))$ et $\log(P(s|H_1))$ de la manière suivante :

$$P(s|H_0) = P(A) \Leftrightarrow \log(P(s|H_0)) = L(A) \quad (2.11)$$

$$P(s|H_1) = P(w) \times P(\bar{w}) \Leftrightarrow \log(P(s|H_1)) = L(w) + L(\bar{w}) \quad (2.12)$$

Le logarithme du rapport de vraisemblance défini par l'équation 2.13 est le suivant :

$$\log\Lambda = L(w) + L(\bar{w}) - L(A) \quad (2.13)$$

2.2.4 Rapport de vraisemblance généralisé

Les paramètres σ_R et m_R étant inconnus, une solution classique consiste à les estimer au sens du maximum de vraisemblance (dans ce cas, les estimateurs de moyenne et de variance classiques). Les σ_R et m_R peuvent être remplacés par leurs estimations $\hat{\sigma}_R$ et \hat{m}_R . On parle alors de **rapport de vraisemblance généralisé** ou *Generalized Likelihood Ratio Test* (GLRT).

A partir de l'équation 2.13 nous obtenons l'équation suivante (le détail des calculs se trouve dans l'annexe A) :

$$\log \Lambda = -\frac{1}{2}(\mathbf{N}_w \log(\hat{\sigma}_w^2) + \mathbf{N}_{\bar{w}} \log(\hat{\sigma}_{\bar{w}}^2) - \mathbf{N}_A \log(\hat{\sigma}_A^2)) \quad (2.14)$$

avec les notations suivantes :

- $N_w, N_{\bar{w}}, N_A$: nombre de pixels des régions w, \bar{w}, A
- $\hat{\sigma}_w^2, \hat{\sigma}_{\bar{w}}^2, \hat{\sigma}_A^2$: estimation de la variance des régions w, \bar{w}, A

Pour une image de N pixels nous avons une complexité en $O(N)$.

Nous nommerons **GLRT-Var** l'algorithme utilisant l'équation 2.14. Il est à noter qu'il s'agit de l'approche utilisée par Réfrégier et al [50].

2.2.5 Hypothèse sur les variances

Une petite originalité de nos travaux a consisté à faire l'hypothèse supplémentaire que le fond et les cibles ont la même variance ($\sigma_A = \sigma_{\bar{w}} = \sigma_w = \sigma$). Dans la partie 2.3, nous discuterons plus en détails de cette hypothèse et de sa validité.

Cette hypothèse présente quelques avantages : le premier est évidemment qu'elle simplifie les calculs ; le second est qu'elle a pour effet de faire disparaître les paramètres σ_R dont l'estimation est très bruitée pour des petites régions.

Ce qui conduit, après quelques simplifications calculatoires, à la formule suivante (le détail des calculs se trouve dans l'annexe A) :

$$\log \Lambda \propto N_w \hat{m}_w^2 + N_{\bar{w}} \hat{m}_{\bar{w}}^2 - N_A \hat{m}_A^2 \quad (2.15)$$

où \propto signifie « est proportionnel à ».

Puis simplifiée en :

$$\log\Lambda = N_w \hat{m}_w^2 + N_{\bar{w}} \hat{m}_{\bar{w}}^2 - N_A \hat{m}_A^2 \quad (2.16)$$

avec les notations suivantes :

- $N_w, N_{\bar{w}}, N_A$: nombre de pixels des régions w, \bar{w}, A
- $\hat{m}_w, \hat{m}_{\bar{w}}, \hat{m}_A$: estimation de la moyenne des régions w, \bar{w}, A

Du fait de l'hypothèse de variance identique sur la cible et le fond, la variance a disparu de cette équation. Du fait de l'estimation de moyenne, ce que nous calculons maintenant est un Rapport de Vraisemblance Généralisé.

Pour une image de N pixels nous avons une complexité en $O(N)$.

Nous noterons **GLRT** l'algorithme utilisant l'équation 2.16.

2.2.6 Taux de fausse alarme constant

Cette méthode est à taux de fausse alarme constant (TFAC ou CFAR pour *Constant False Alarm Rate*) [49]. Ceci signifie qu'en l'absence de cible, le rapport de vraisemblance généralisé est indépendant de la moyenne du fond. Cette propriété constitue un atout intéressant face aux problèmes d'illumination que présentent les acquisitions d'images satellitaires optiques.

Nous allons rapidement démontrer que la méthode est TFAC. Nous appliquons une transformation à la moyenne du fond : $\hat{m}'_A = \hat{m}_A + \delta$. Dans le cas où nous n'avons que du fond, cela revient à écrire : $\hat{m}'_w = \hat{m}_w + \delta$ et $\hat{m}'_{\bar{w}} = \hat{m}_{\bar{w}} + \delta$.

Nous remplaçons dans l'équation 2.16, nous obtenons :

$$\begin{aligned} \log\Lambda' &= N_w \hat{m}'_w{}^2 + N_{\bar{w}} \hat{m}'_{\bar{w}}{}^2 - N_A \hat{m}'_A{}^2 \\ \log\Lambda' &= N_w (\hat{m}_w + \delta)^2 + N_{\bar{w}} (\hat{m}_{\bar{w}} + \delta)^2 - N_A (\hat{m}_A + \delta)^2 \\ \log\Lambda' &= N_w \hat{m}_w^2 + N_{\bar{w}} \hat{m}_{\bar{w}}^2 - N_A \hat{m}_A^2 + \delta^2 \underbrace{(N_w + N_{\bar{w}} - N_A)}_{=0} + 2\delta \underbrace{(N_w \hat{m}_w + N_{\bar{w}} \hat{m}_{\bar{w}} - N_A \hat{m}_A)}_{=0} \\ \log\Lambda' &= \log\Lambda \end{aligned} \quad (2.17)$$

Ainsi, le rapport de vraisemblance généralisé est indépendant des paramètres de fond et nous avons bien démontré que la méthode est TFAC.

2.2.7 L'algorithme de pré-détection

Nous obtenons un algorithme de pré-détection simple et rapide :

1. Pour chaque pixel de l'image, nous calculons l'estimation du rapport de vraisemblance généralisé donnée par l'équation 2.14, nous obtenons une matrice que nous nommerons \mathbf{M}_{GLRT} .
2. Cette quantité est seuillée. Nous verrons dans 2.2.12 comment est choisi le seuil.

2.2.8 Comportement de l'algorithme en présence d'une cible

Nous voulons montrer le comportement de l'algorithme **GLRT** en présence d'une cible (petite et grosse) et également voir ce qui se passe dans les cas non couverts par H_0 et H_1 , pour cela nous utilisons une image de synthèse.

Cas d'une petite cible

Nous allons d'abord observer le comportement de l'algorithme dans le cas d'une petite cible. Lorsque nous appliquons l'algorithme sur l'image de synthèse de la figure 2.3(a), nous obtenons le résultat de la figure 2.3(b). Nous constatons que la présence de la cible est caractérisée par un pic de valeur.

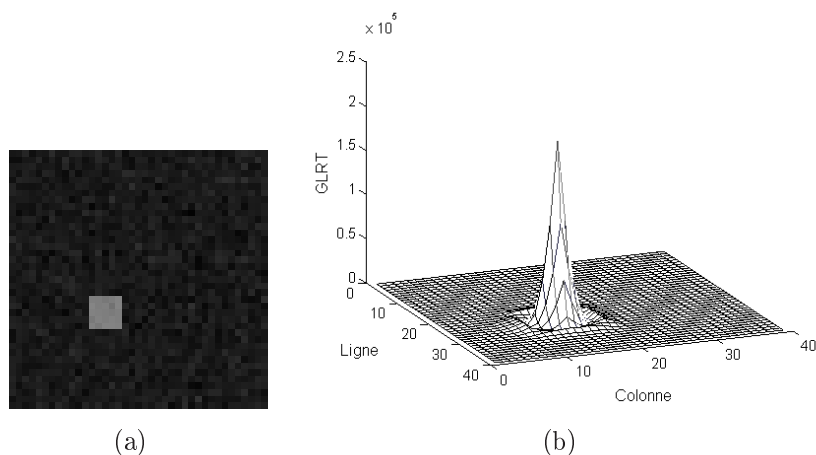


FIG. 2.3 – Résultat du calcul du rapport de vraisemblance sur une image de synthèse contenant une cible. Dans (a), l'image de synthèse de taille 39×39 pixels. La cible est de taille 5×5 pixels et suit une loi normale gaussienne d'espérance 0,5 et d'écart type 0,02. Le fond suit une loi normale gaussienne d'espérance 0,01 et d'écart type 0,03. Dans (b), le résultat de l'application de l'algorithme **GLRT** sur l'image (a). Nous avons utilisé les paramètres suivants : $L_w=5$ et $L_A=11$.

Dans la figure 2.4, nous allons étudier la réponse de l'algorithme sur la ligne qui passe par le milieu d'une cible. Nous constatons que lorsque la fenêtre représentant la cible

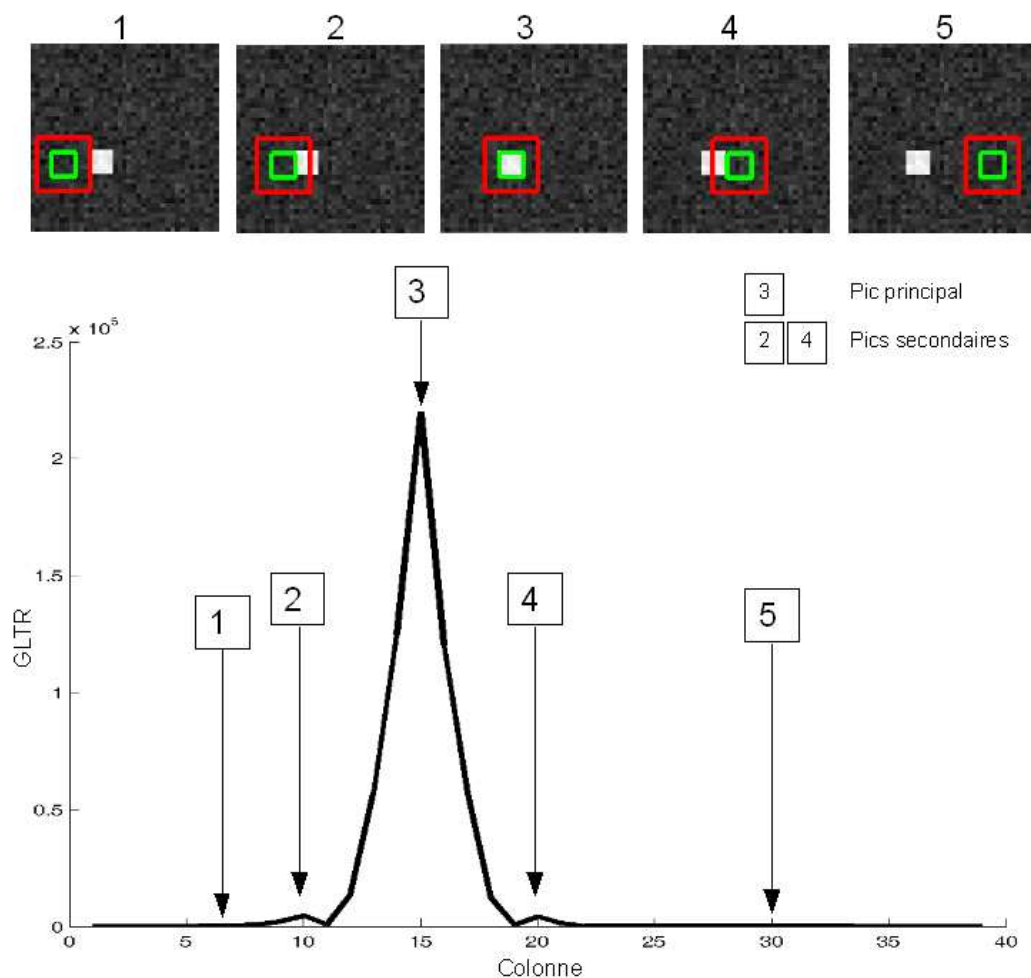


FIG. 2.4 – Résultat de l'algorithme sur l'image de synthèse sur la ligne qui passe par le milieu de la cible.

(carré vert) commence à recouvrir la cible, nous avons un pic secondaire assez faible puis, au fur et à mesure que la fenêtre va recouvrir la cible, la réponse va être de plus en plus forte pour être au maximum quand la fenêtre verte recouvre toute la cible. Au fur et à mesure que la fenêtre verte "quitte" la cible, la réponse diminue pour devenir nulle quand la fenêtre glissante (fenêtre rouge) ne recouvre que du fond.

Localisation d'une cible

La figure 2.5 illustre ce qui se passe lors du seuillage. Nous pourrions avoir plusieurs valeurs du GLRT au dessus du seuil pour une même cible, ce que nous allons nommer des réponses positives multiples. Pour nous affranchir de ces réponses multiples, nous considérons les réponses positives connexes comme une cible et nous gardons comme position de la cible le pixel ayant une réponse maximale. Pour la suite, nous associerons à chaque cible cette valeur maximum du rapport de vraisemblance.

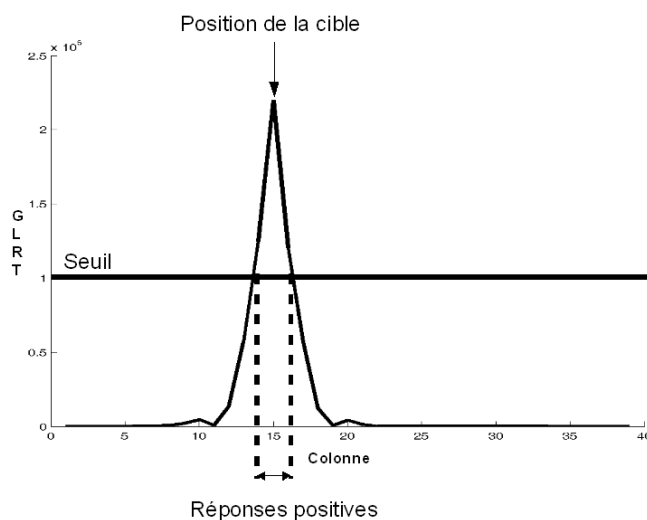


FIG. 2.5 - Résultat du seuillage.

Cas d'une grosse cible

Nous allons maintenant observer le comportement de l'algorithme dans le cas d'une grosse cible. Lorsque nous appliquons l'algorithme sur l'image de synthèse de la figure 2.6(a), nous obtenons le résultat de la figure 2.6(b).

Nous constatons que la présence de la grosse cible est caractérisée par la présence de quatre pics de valeur (un à chaque coin de la cible). Cette constatation nous laisse croire que les grands bateaux pourront être également détecté par cette méthode alors qu'elle n'est pas a priori prévue pour cela (nous verrons dans le chapitre 5 la confirmation de cette supposition). Ainsi, avec une même méthode, nous pourrions à la fois détecter les petits et les grand bateaux.

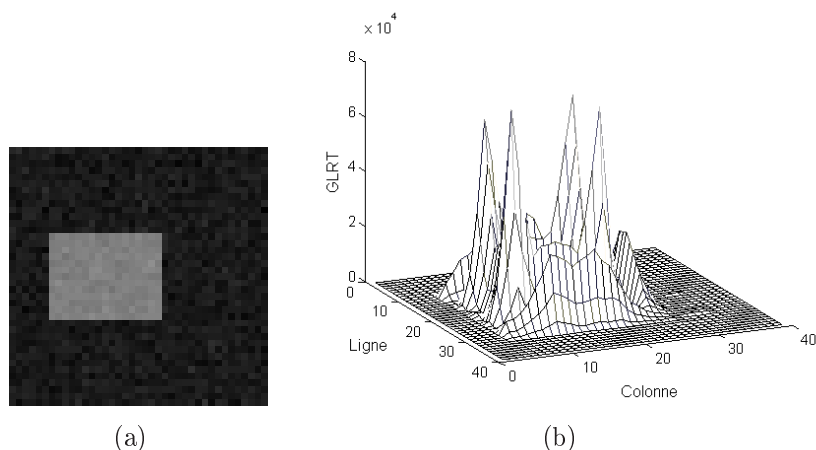


FIG. 2.6 – Résultat du calcul du rapport de vraisemblance (**GLRT**) sur une image de synthèse contenant une cible.

Dans (a), l'image de synthèse de taille 39×39 pixels. La cible est de taille 6×8 pixels et suit une loi normale gaussienne d'espérance 0,5 et d'écart type 0,02. Le fond suit une loi normale gaussienne d'espérance 0,01 et d'écart type 0,03.

Dans (b), le résultat de l'application de l'algorithme **GLRT** sur l'image (a). Nous avons utilisé les paramètres suivants : $L_w=5$ et $L_A=11$.

2.2.9 Les paramètres de la méthode

La méthode utilise trois paramètres :

1. L_w : la taille de la cible. Notre but étant de détecter des petits bateaux de taille comprise entre 15m et 30m, la résolution des images utilisées étant de 5m, nous avons fixé le paramètre de la taille de la cible $L_w=3$.
2. L_A : la taille de la fenêtre d'analyse. Dans la partie 2.2.11, nous décrivons la méthode pour fixer la taille de la fenêtre d'analyse.
3. Le seuil de détection. Dans la partie 2.2.12, nous expliquerons l'algorithme utilisé pour déterminer ce seuil de détection.

2.2.10 Méthode d'évaluation des algorithmes

Pour visualiser les performances globales d'un algorithme pour tous les seuils possibles, il est de coutume d'observer les **courbes ROC** (*Receiver Operating Characteristic*). Nous les utiliserons donc pour évaluer, caractériser et comparer les différents modèles, mais aussi pour choisir une taille de fenêtre d'analyse adaptée.

Pour chaque seuil de détection, nous calculons la probabilité de détection et la probabilité de fausse alarme.

La **probabilité de détection** $PD = P(\text{décider } H_1|H_1)$ est calculée de la manière suivante :

$$PD = \frac{\text{Nombre de cibles détectées}}{\text{Nombre de cibles présentes}}$$

La **probabilité de fausse alarme** $PFA = P(\text{décider } H_1|H_0)$ est calculée de la manière suivante :

$$PFA = \frac{N_{FA}}{N - N_{\text{Détection}}}$$

où N_{FA} est le nombre de fausses détections (détecté comme étant des cibles, mais qui n'en sont pas), N est la taille de la scène traitée et $N_{\text{Détection}}$ est le nombre de vraies détections dans la scène.

Le tracé du couple (PFA, PD) , pour $0 < \tau < +\infty$ produit une courbe nommée courbe ROC [19]. Cette courbe représente l'évolution de la probabilité de détection (PD) en fonction de la probabilité de fausse alarme (PFA). Ces deux probabilités sont liées à la valeur du seuil τ . Nous présentons dans la figure 2.7 un exemple de courbe ROC.

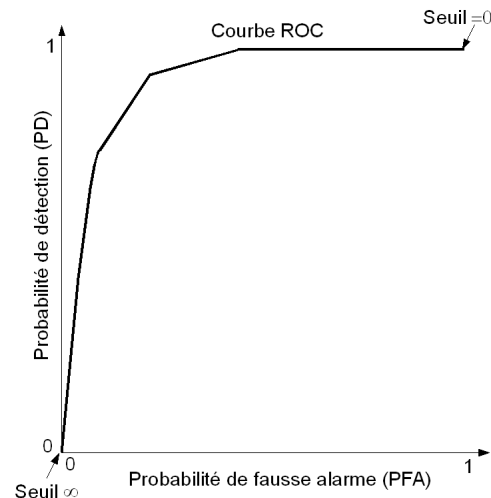


FIG. 2.7 - Exemple de courbe ROC.

Grâce à ces courbes ROC, nous allons comparer différents algorithmes. Pour chacun d'eux, nous produisons une courbe ROC. Pour une probabilité de fausse alarme (PFA) donnée, le meilleur algorithme sera celui qui a la probabilité de détection (PD) la plus élevée.

2.2.11 Choix d'une taille de fenêtre d'analyse

Pour réaliser nos expérimentations, nous avons utilisé la base d'images 1 décrit dans le chapitre 1.

Nous voulons choisir une taille de fenêtre d'analyse. Pour cela, nous testons différentes tailles de fenêtre :

$$\left\{ \begin{array}{l} 7 \times 7 \text{ pixels } (L_A=7) \\ 11 \times 11 \text{ pixels } (L_A=11) \\ 15 \times 15 \text{ pixels } (L_A=15) \end{array} \right.$$

Nous construisons une courbe ROC pour chaque taille de fenêtre en faisant varier le seuil de détection. L'observation des courbes ROC nous permet de sélectionner la taille de la fenêtre la plus adaptée.

Les différentes courbes ROC obtenues sur la base de test en utilisant l'algorithme **GLRT** sont représentées dans la figure 2.8. Nous constatons que quelque soit la valeur du seuil, la fenêtre d'analyse de 7×7 pixels (courbe en rouge) donne les meilleurs résultats (les valeurs PD de la fenêtre 7×7 pixels sont toujours supérieures (ou égales) aux valeurs de PD des deux autres fenêtres quelque soit la PFA).

De façon évidente, une fenêtre de 7×7 pixels ($L_A=7$) sera donc utilisée dans la suite de l'étude.

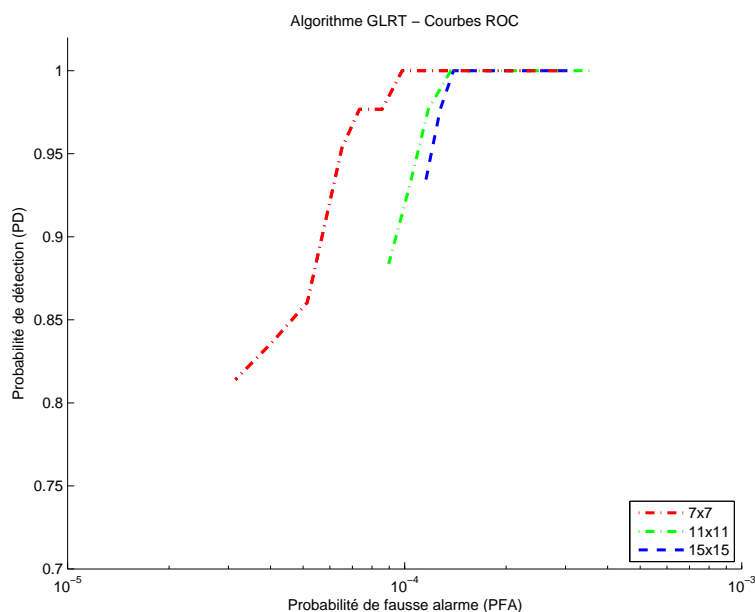


FIG. 2.8 - Courbes ROC pour l'algorithme **GLRT** et pour des tailles de fenêtre 7×7 ($L_A=7$) 11×11 ($L_A=11$) et 15×15 ($L_A=15$) pixels. L'algorithme a été exécuté sur l'ensemble de la base d'images 1.

2.2.12 Choix du seuil de détection

Nous allons maintenant définir un seuil de détection. Celui-ci, comme le montre les courbes ROC, relève d'un compromis entre PD (la plus haute possible) et PFA (que l'on voudrait la moins haute possible). Le choix du seuil relève alors exclusivement du point de vue opérationnel. Dans notre cas, l'objectif industriel fixé par notre partenaire est d'avoir un algorithme capable de détecter toutes les cibles (si possible) de façon à fournir une pré-estimation de la présence de bateaux à un photo-interprète humain. Un seuil qui fasse cela nous donne une PFA de 10^{-4} d'après les courbes ROC de la figure 2.8.

Nous avons vu dans la partie 2.2.8 que la présence d'une cible est caractérisée par un pic dans la matrice M_{GLRT} (figure 2.3(b)). Nous avons constaté que les valeurs pour la matrice M_{GLRT} en présence d'une cible peuvent fortement varier (de 580 à $\sim 10^4$) suivant les différentes situations rencontrées : un fond homogène, un fond non-homogène ou nuageux.

Il est donc difficile de fixer un seuil unique valable pour toutes les images. Si l'on fixe un seuil bas, la probabilité de fausse alarme sera importante et si l'on fixe un seuil haut on risque de faire baisser la probabilité de détection des cibles. Il est donc intéressant de fixer un seuil dépendant des différentes situations rencontrées.

Dans une image complète, nous pouvons avoir différents cas de figure simultanément. En effet, nous pouvons avoir à la fois des zones de fond homogène, des zones de fond non-homogène et des zones de fond nuageux. Pour éviter d'avoir des zones trop hétérogènes à traiter nous allons découper l'image en sous-images (dalles) de taille 1000×1000 pixels, comme cela est illustré dans la figure 2.9.

Nous avons réalisé des tests avec des tailles de dalles variant de 500×500 pixels à 3000×3000 pixels et nous avons sélectionné une taille de 1000×1000 pixels. En effet c'est un bon compromis : la taille de la dalle est suffisamment petite pour avoir une situation homogène, mais pas trop grande pour ne pas augmenter le temps de calcul en travaillant sur des données trop importantes. Un autre avantage du découpage en dalle est de pouvoir paralléliser les traitements, nous en discuterons plus en détail dans le chapitre 5.

Pour caractériser les différentes situations, l'utilisation de paramètres de texture serait bien adaptée. Etant données nos contraintes de temps, nous avons préféré être pragmatique et travailler sur une quantité déjà calculée : la matrice M_{GLRT} . Nous verrons, par la suite, qu'il est aisé de caractériser les différentes situations à partir de M_{GLRT} .

Le principe va donc être de fixer le seuil en fonction des situations rencontrées et des pics trouvés dans la matrice M_{GLRT} .

Pour chaque dalle, nous calculons pour chaque position l'estimation de rapport de vraisemblance généralisé donnée par l'équation 2.16. Nous obtenons la matrice M_{GLRT} ,

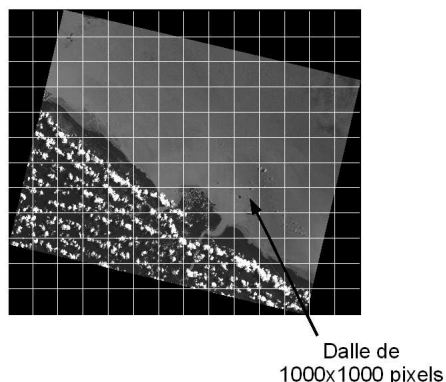


FIG. 2.9 – Découpage de l'image en dalles de 1000×1000 pixels.

sur laquelle nous allons appliquer l'algorithme de recherche de seuil.

Nous allons donc fixer le seuil de détection pour une image à l'aide d'un algorithme adaptatif en prenant en compte les pics présents dans M_{GLRT} . Précédemment, nous avons distingué plusieurs cas de figure. Nous allons maintenant étudier le traitement de ces différentes situations.

Nous avons fixé empiriquement un seuil τ_{Min} . Ce seuil a été déterminé à partir d'un cas « difficile » où la cible et le fond étaient très peu contrastés.

Fond homogène

Lorsque nous sommes en présence d'un fond homogène, les valeurs de M_{GLRT} sont faibles comme cela est illustré dans la figure 2.10.

Si la moyenne des valeurs de M_{GLRT} est inférieure à $\tau_{FondHomogène}$ alors nous considérons que nous sommes en présence d'un fond homogène. $\tau_{FondHomogène}$ a été fixé empiriquement et a une valeur faible (5).

Nous pouvons avoir deux cas possibles :

- si toutes les valeurs de M_{GLRT} sont inférieures à τ_{min} alors il n'y a pas de cible,
- s'il existe des valeurs de M_{GLRT} supérieures à τ_{min} alors une ou des cibles sont présentes comme cela est illustré dans la figure 2.11.

Dans le cas d'un fond homogène, nous appliquons comme seuil τ_{min} .

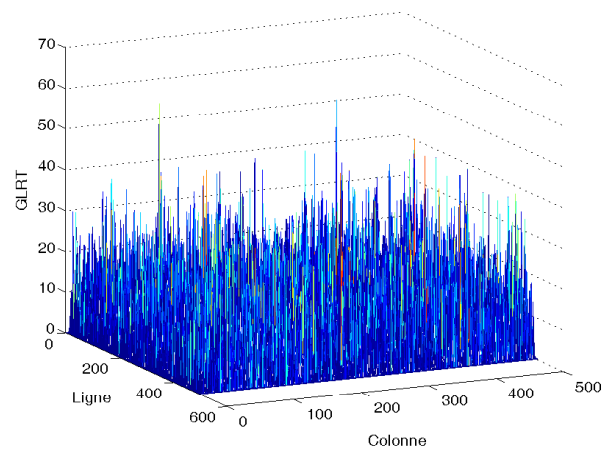


FIG. 2.10 – Exemple de M_{GLRT} en présence d'un fond homogène sans cible.

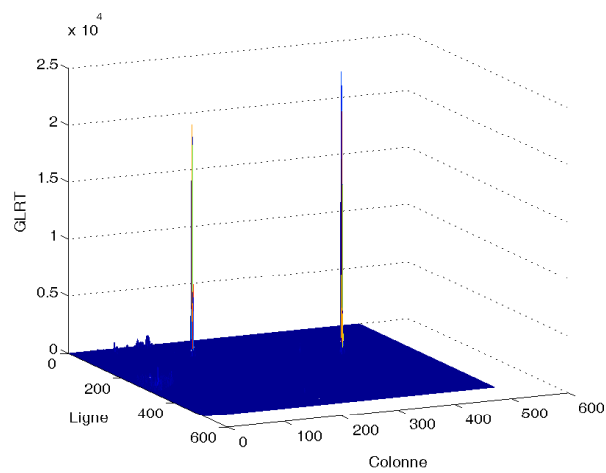


FIG. 2.11 – Exemple de M_{GLRT} en présence d'un fond homogène avec deux cibles.

Fond non-homogène ou nuageux

Nous définissons la notion de pic significatif : un pic sera considéré significatif lorsqu'il sera suffisamment important pour se distinguer du fond.

Pour déterminer si un pic est significatif, nous procédons de la manière suivante :

1. Nous recherchons le pic en cherchant la valeur maximum dans M_{GLRT} (MAX_{GLRT}).
2. Nous éliminons le pic en mettant les valeurs à -1 dans une petite fenêtre de taille $2L_A \times 2L_A$ autour du maximum (figure 2.12). Ainsi nous éliminons le pic principal et les pics secondaires (figure 2.4).
3. Nous calculons la moyenne m_{GLRT} et l'écart type σ_{GLRT} sur la nouvelle matrice M_{GLRT} obtenue après l'élimination du pic.
4. Nous décidons qu'il y a un pic significatif si :
 $MAX_{GLRT} - (m_{GLRT} + \sigma_{GLRT}) > Seuil_{pic}$
 où $Seuil_{pic}$ est un seuil fixé empiriquement.

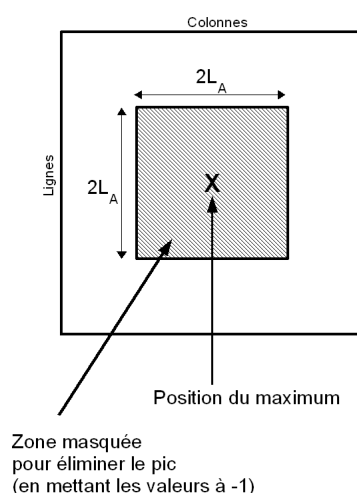


FIG. 2.12 – *Elimination d'un pic : en mettant les valeurs à -1 dans une petite fenêtre de taille $2L_A \times 2L_A$ autour du maximum.*

Dans le cas d'un fond non-homogène sans cible nous n'avons pas de pic significatif, comme cela est illustré dans la figure 2.13.

Pour le traitement de ce type de zones, nous procédons de la manière suivante :

1. Nous déterminons si un pic significatif est présent dans M_{GLRT} .
2. S'il existe un pic significatif alors nous fixons comme seuil $\tau = MAX_{GLRT} - 1$ et nous recommençons (1).
3. S'il n'y a pas de pic significatif nous considérons qu'il n'y a pas (ou plus) de cibles dans la dalle.

S'il s'agit du premier pic recherché alors il n'y a pas de cible et nous ne seuillons pas la dalle sinon nous appliquerons τ .

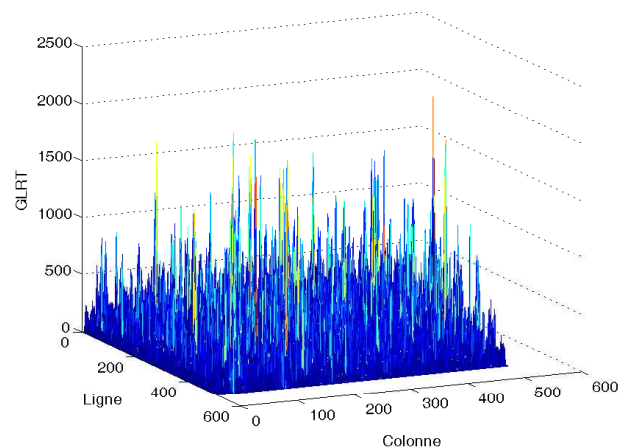


FIG. 2.13 – Exemple de M_{GLRT} en présence d'un fond non-homogène sans cible.

Dans le cas d'un fond nuageux, nous pouvons avoir de nombreux pics dans la dalle comme cela est illustré dans la figure 2.14.

Pour le traitement de ces zones, nous allons comme précédemment rechercher les pics significatifs. Si dans une dalle nous avons plus de dix pics significatifs alors nous découpons en quatre la dalle et nous appliquons à nouveau le seuillage sur les 4 sous-dalles.

Le but étant de détecter d'éventuelles cibles qui seraient au milieu des nuages.

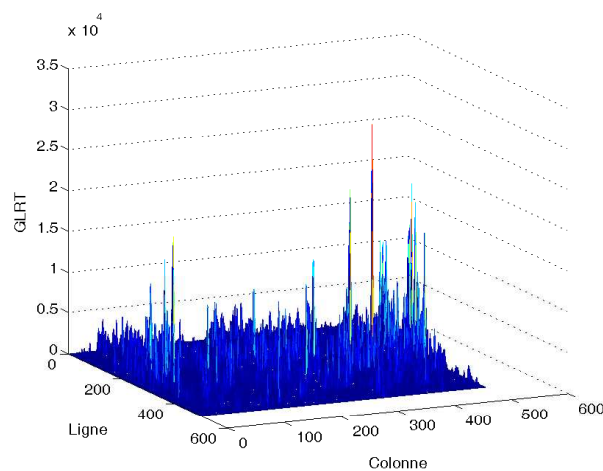


FIG. 2.14 – Exemple de M_{GLRT} en présence d'un fond nuageux.

2.3 Discussion sur notre modèle

Dans cette partie, nous voulons discuter des hypothèses utilisées dans notre modèle afin de classer leur importance et de savoir s'il est intéressant d'envisager un modèle plus complexe.

Nous rappelons ces hypothèses :

- une distribution gaussienne pour le fond et la cible,
- une variance identique pour le fond et la cible,
- les intensités des pixels sont indépendantes et uniformément distribuées (bruit blanc).

La variance identique pour le fond et la cible

La première hypothèse que nous testons est celle qui nous distingue de Réfrégier et al : variance identique pour le fond et la cible. Pour cela nous allons d'abord comparer les performances des deux algorithmes **GLRT** et **GLRT-Var**, à l'aide des courbes ROC, sur des images réelles (base d'images 1 décrite dans la chapitre 1).

Nous avons utilisé les paramètres suivants :

- $L_w=3$
- Pour la taille de la fenêtre d'analyse (L_A), nous avons choisi pour chaque algorithme la taille de fenêtre donnant les meilleurs résultats :
 - pour **GLRT** nous avons $L_A=7$
 - pour **GLRT-Var** nous avons $L_A=15$

Les résultats sont présentés dans la figure 2.15.

Les courbes ROC font clairement apparaître que les performances de l'algorithme **GLRT** (courbe rouge) sont meilleures que celles de l'algorithme **GLRT-Var** (courbe bleue) : pour une probabilité de détection de 1 nous avons une probabilité de fausse alarme beaucoup plus faible avec l'algorithme **GLRT**. Cette hypothèse sur la variance est donc primordiale.

La gaussianité

Nous voudrions savoir si ces moins bonnes performances sont dues à la qualité du modèle utilisé (distribution gaussienne) ou à une mauvaise estimation des paramètres. Nous allons donc prendre un modèle peu réaliste (loi de distribution Gamma) et comparer les résultats des différents modèles.

Nous utilisons donc le modèle peu réaliste suivant :

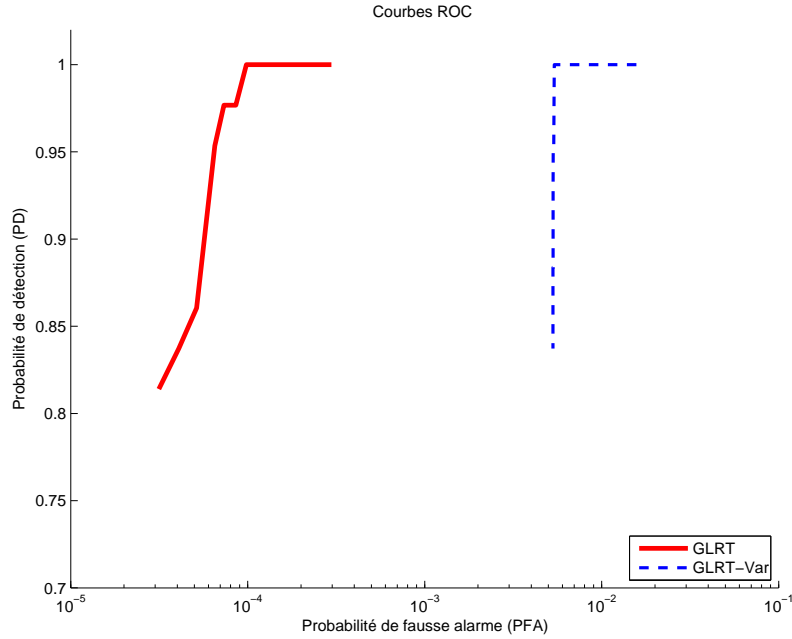


FIG. 2.15 – Courbes ROC pour les algorithmes **GLRT** et **GLRT-Var** sur des images réelles de la base d'images 1.

Pour chaque algorithme on utilise la taille de fenêtre la mieux adaptée :

- pour **GLRT**, $L_A=7$ (partie 2.2.11)

- pour **GLRT-Var**, $L_A=15$ (annexe B.1) et

Pour les deux algorithmes le paramètre $L_w=3$.

- chaque pixel du bateau suit une loi Gamma de moyenne m et d'ordre L ,
- chaque pixel du fond suit une loi Gamma de moyenne m et d'ordre L .

Pour un pixel s_i nous avons :

$$P^{m,L}(s_i) = \frac{1}{\Gamma(L)} \left(\frac{L}{m}\right)^L s_i^{L-1} e^{-L\frac{s_i}{m}} \quad (2.18)$$

Sous cette hypothèse, l'estimation du rapport de vraisemblance généralisé est le suivant :

$$\log \Lambda = -L[N_w \log(\hat{m}_w) + N_{\bar{w}} \log(\hat{m}_{\bar{w}}) - N_A \log(\hat{m}_A)] \quad (2.19)$$

avec les notations suivantes :

- $N_w, N_{\bar{w}}, N_A$: nombre de pixels des régions w, \bar{w}, A
- $\hat{m}_w, \hat{m}_{\bar{w}}, \hat{m}_A$: estimation de la moyenne des régions w, \bar{w}, A

Nous noterons **GLRT-Gamma** l'algorithme utilisant l'équation 2.19. Nous sélectionnons pour cet algorithme la taille de fenêtre d'analyse la mieux adaptée à l'aide des courbes ROC qui sont présentées dans l'annexe B.2.

Les courbes ROC des différents algorithmes sont présentées dans la figure 2.16.

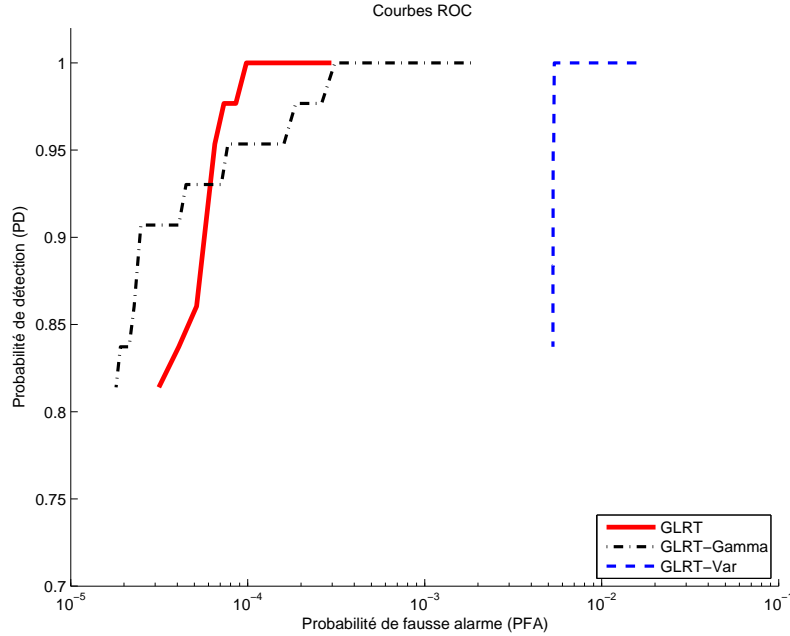


FIG. 2.16 - Comparaison des courbes ROC des algorithmes **GLRT**, **GLRT-Var** et **GLRT-Gamma**.

Pour chaque algorithme on utilise la taille de fenêtre la mieux adaptée :

- pour **GLRT**, $L_A=7$ (partie 2.2.11)
- pour **GLRT-Var**, $L_A=15$ (annexe B.1) et
- pour **GLRT-Gamma**, $L_A=11$ (annexe B.2).

Pour les trois algorithmes le paramètre $L_w=3$.

La comparaison entre les algorithmes **GLRT**, **GLRT-Var** et **GLRT-Gamma** à l'aide des courbes ROC fait clairement apparaître que **GLRT** donne les meilleurs résultats. Nous constatons également que **GLRT-Gamma** a de meilleures performances que **GLRT-Var**. Or **GLRT-Gamma** possède un modèle peu réaliste. Ainsi avec un modèle faux, nous obtenons de meilleurs résultats qu'avec un modèle plus réaliste. A ce stade, force est de supposer que c'est probablement l'estimation des paramètres plus que la qualité du modèle qui va dégrader les performances.

Pour confirmer cela, nous allons tester les deux algorithmes **GLRT** et **GLRT-Var** sur 3000 images de synthèses générées aléatoirement et correspondant au modèle théorique de bruit blanc gaussien. Dans ces images, les moyennes des cibles et du fond sont représentatives de ce que nous avons dans les vraies images et l'écart de variance est suffisamment conséquente.

La comparaison des courbes ROC est présentée dans la figure 2.17. Encore une fois nous constatons que les performances sont meilleures pour l'algorithme **GLRT**. Ce qui confirme que l'estimation des paramètres a tendance à dégrader les performances.

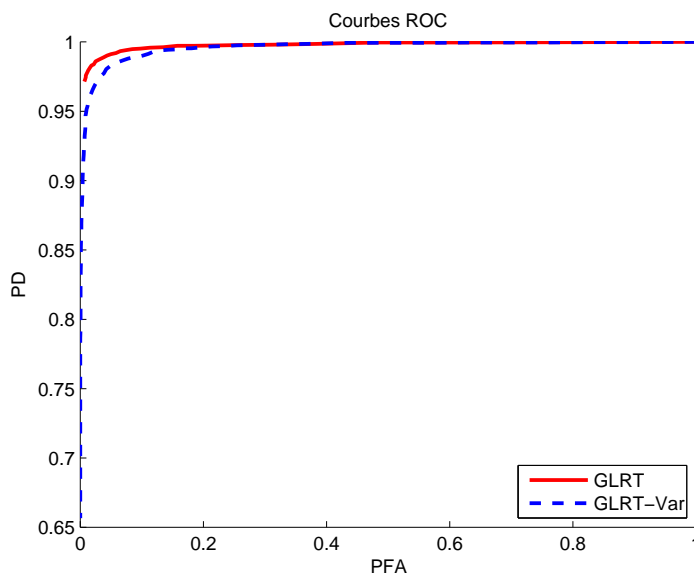


FIG. 2.17 – Comparaison des courbes ROC des algorithmes **GLRT** et **GLRT-Var** sur 3000 images de synthèse.

Les paramètres des images sont les suivants :

- pour le fond, la moyenne est de 60 et la variance est de 20,
- pour la cible, la moyenne est de 100 et la variance est de 30.

Pour les deux algorithmes, la taille de la fenêtre d'analyse $L_A=7$ et $L_w=3$.

Nous avons montré que pour des petites cibles l'estimation des paramètres prime sur la qualité du modèle en terme de performances.

Bruit blanc

La dernière hypothèse de notre modèle (bruit blanc) pourrait être traitée avec des études plus complexes (champ de markov, par exemple). Néanmoins, les résultats précédents inciteraient à penser que les performances seraient moins bonnes en utilisant un modèle plus réaliste théoriquement et donc plus complexe (utilisant plus de paramètres).

En conclusion, un meilleur modèle théorique ne donne pas forcément les meilleurs résultats dans le contexte de la détection de petites cibles. Ce résultat nous semble un point assez important au sein de ces travaux de recherches dans un cadre opérationnel.

2.4 Résultats des expérimentations

2.4.1 Résultats sur la base d'images 1

Nous caractérisons maintenant les performances avec l'algorithme **GLRT** dont nous avons montré l'intérêt. Nous avons testé la méthode de pré-détection sur les images réelles de la base d'images 1 décrite dans la partie 1.6.1. Elle est composée de quatre sous-bases, classées selon les situations que nous pouvons rencontrer :

- 1.A : mer calme.
- 1.B : mer agitée.
- 1.C : présence de nuages.
- 1.D : peu de contraste entre la cible et le fond.

Les résultats de la pré-détection sont présentés dans la table 2.1.

| | 1.A | 1.B | 1.C | 1.D |
|----------------------------------|-----|-----------------------|-------------------------|------------------------|
| Nombre de bateaux présents | 32 | 5 | 16 | 14 |
| Nombre de pré-détections | 32 | 64 | 128 | 15 |
| Nombre de fausses pré-détections | 0 | 59 | 112 | 1 |
| PD | 1 | 1 | 1 | 1 |
| PFA | 0 | $2,32 \times 10^{-5}$ | $2,7542 \times 10^{-5}$ | $5,775 \times 10^{-7}$ |

TAB. 2.1 - Résultats de la méthode de pré-détection pour les quatre sous-bases de la base d'images 1. Les paramètres sont : $L_A=7$ pixels et $L_w=3$ pixels.

Conformément à l'intuition, les situations de mer calme (1.A) sont faciles et toutes les cibles sont détectées sans apparition de fausse alarme.

Les résultats dans les situations de mer agitée (1.B) sont moins bons. Quelques fausses alarmes apparaissent lorsque la mer est très forte et qu'il y a des crêtes de vagues comme dans la figure 2.18(a).

La présence de nuages (1.C) est plus problématique. En effet, l'observation des résultats de l'algorithme montre que les bords des nuages ont tendance à générer des fausses alarmes comme dans la figure 2.18(b).

2.4.2 Exemples

Dans les figures 2.19, 2.20, 2.21 et 2.22, nous illustrons le résultat de la méthode de pré-détection sur quatre exemples.

Dans ces exemples, les pré-détections entourées d'un carré blanc correspondent à des

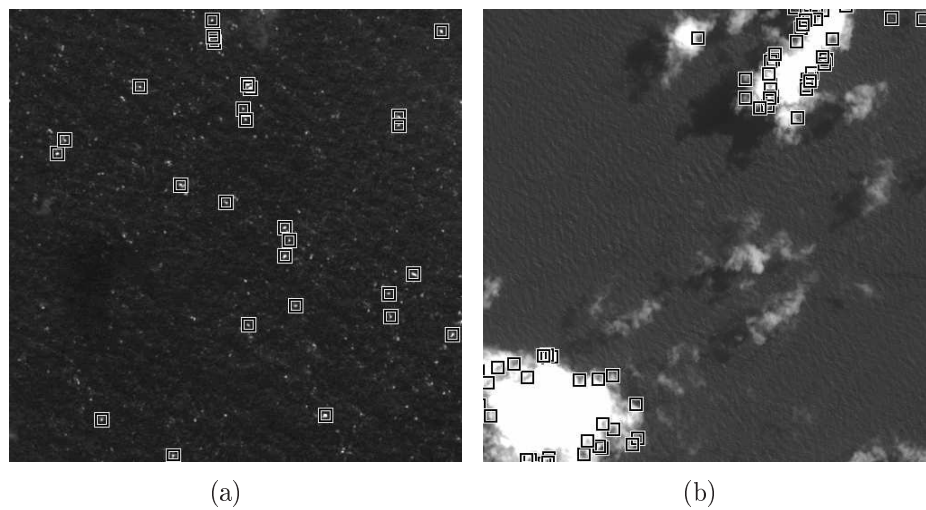


FIG. 2.18 – Cette figure illustre les différentes fausses alarmes. Les pré-détections sont entourées par un carré blanc et noir.

(a) montre des fausses alarmes dues aux crêtes de vagues en présence d'une mer agitée.

(b) montre des fausses alarmes dues à la présence de nuage.

vraies cibles et les pré-détections entourées par un carré blanc et noir correspondent à des fausses alarmes.

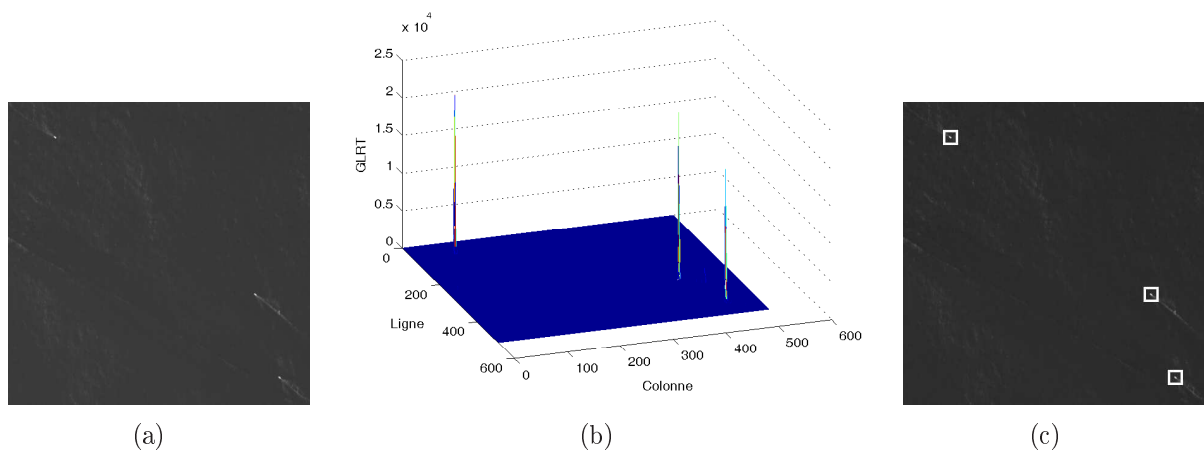


FIG. 2.19 – Résultat de la pré-détection sur une image contenant trois cibles dans une mer calme (a). En (b), nous avons le résultat du calcul du GLRT et en (c) le résultat après le seuillage adaptatif. Nous avons trois pré-détections qui correspondent à des vraies cibles.

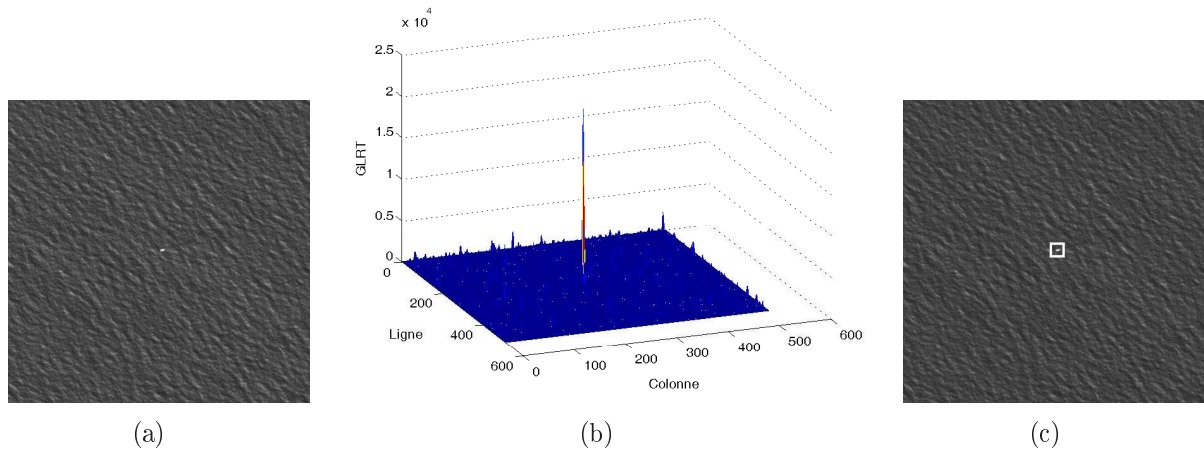


FIG. 2.20 – Exemple d'image extraite de la base d'images 1.B. (a) l'image traitée contient une cible et une mer agitée . En (b), nous avons le résultat du calcul du GLRT et en (c) le résultat après le seuillage adaptatif. Nous avons une pré-détection qui correspond à une vraie cible.

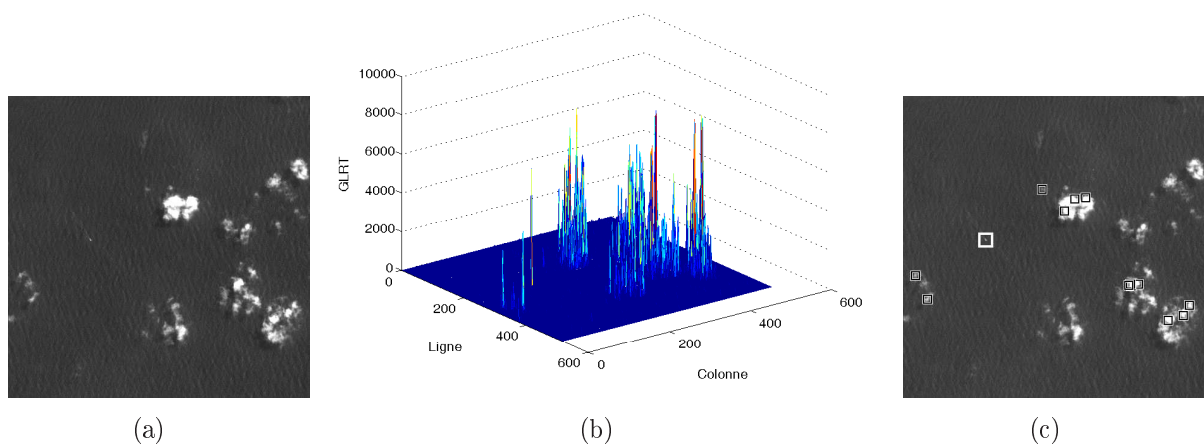


FIG. 2.21 – Résultat de la pré-détection sur une image contenant une cible et des nuages (a). En (b), nous avons le résultat du calcul du GLRT et en (c) le résultat après le seuillage adaptatif. Nous avons une pré-détection qui correspond à une vraie cible et onze fausses alarmes localisées sur les bords de certains nuages.

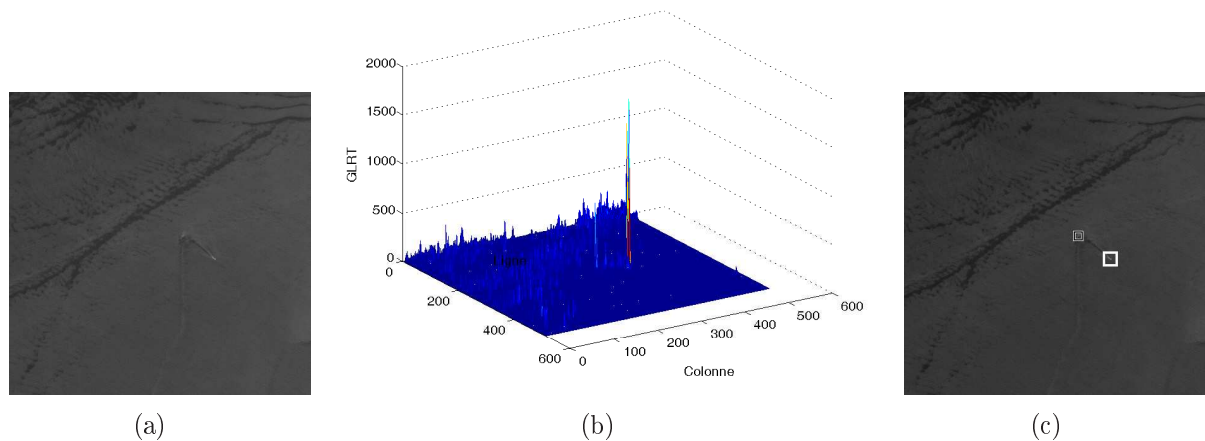


FIG. 2.22 – Résultat de la pré-détection sur une image contenant une cible qui est peu contrastée avec le fond (a). En (b), nous avons le résultat du calcul du GLRT et en (c) le résultat après le seuillage adaptatif. Nous avons une pré-détection qui correspond à une vraie cible et une fausse alarme localisée au niveau du sillage.

2.5 Conclusion

Dans ce chapitre, nous nous sommes intéressés à la pré-détection des petites cibles à l'aide d'une méthode basée sur la théorie de la décision bayésienne. Nous avons appliqué cette méthode sur des images satellitaires optiques de type SPOT 5 panchromatique de résolution 5m.

Elle présente l'avantage d'être à taux de fausse alarme constant (TFAC) ce qui constitue un atout intéressant face aux problèmes d'illumination que présentent les acquisitions d'images satellitaires optiques.

Les paramètres de la méthode sont la taille de la fenêtre d'analyse et le seuil de détection. La taille de la fenêtre d'analyse a été fixé à l'aide des courbes ROC.

Le seuil de détection est un seuil adaptatif qui s'adapte à l'image traitée.

L'algorithme est relativement rapide : la complexité de l'algorithme est en $O(N)$.

Nous avons évalué les performances de la méthode en terme de probabilité de détection et de probabilité de fausse alarme. Les performances sont très satisfaisantes : la probabilité de détection est excellente pour un taux de fausse alarme d'environ 10^{-5} dans les cas les plus difficiles (mer agitée et présence de nuages).

Nous avons également évalué les limites de la méthode qui sont les fausses alarmes dues à la présence des nuages d'une part et une mer très agitée avec des crêtes de vague d'autre part. Cela représente un nombre de fausses alarmes non négligeable sur une image

complète.

Nous avons également vu que cette méthode, adaptée à la détection de petites cibles, détecte également des grosses cibles. Ainsi, avec la même méthode nous pourrions détecter les petits et les grands bateaux. Cela constitue un aspect très intéressant.

Grâce à cette première étape nous avons défini des régions d'intérêt (ROI). Nous allons maintenant traiter ces pré-détections en faisant une segmentation et une classification dans le but de diminuer le nombre de fausses alarmes et de caractériser les bateaux.

Chapitre 3

Segmentation et classification

3.1 Introduction

La première étape présentée dans le chapitre 2 fournit un ensemble de pré-détections. Parmi ces pré-détections, un certain nombre sont des fausses alarmes. Notre but dans l'étape suivante sera non seulement de réduire ce nombre de fausses détections, mais aussi de trouver leur forme pour caractériser les bateaux détectés.

Pour cela, nous allons réaliser une segmentation afin d'obtenir le contour du bateau. cette étape sera présentée dans la partie 3.2. Ensuite, nous réalisons une classification de ces contours afin de les classer en trois groupes : petit bateau, grand bateau et tout ce qui n'est pas un bateau. La méthode de classification sera présentée dans la partie 3.3 et dans la partie 3.4 nous concluons sur les résultats obtenus.

3.2 La segmentation

S'il existe de nombreuses méthodes de segmentation, dans ce contexte précis, nous avons choisi d'utiliser des contours actifs [45]. La raison de ce choix est simple : les contours actifs sont par nature adaptés à la segmentation d'un objet sur un fond. Etant donné que les objets apparaissent comme un ensemble de pixels connexes, nous n'avons pas besoin de contours actifs géodésiques ou de contours actifs utilisant des courbes de niveaux (level sets) [57]. Nous avons utilisé un algorithme rapide et simple également basé sur la théorie de la décision bayésienne car les résultats, que nous avons obtenus par ces techniques dans le chapitre traitant de la détection, étaient relativement bons et les algorithmes rapides.

3.2.1 Les contours actifs

Cette méthode présentée par Kass et al [45] est, par nature, bien adaptée à la segmentation d'objets simples.

Elle consiste à utiliser un contour déformable. Les déformations du contour sont guidées par une stratégie de minimisation d'une énergie. Cette énergie est fonction d'une énergie interne E_{int} et d'une énergie externe E_{ext} :

$$E = f(E_{int}, E_{ext})$$

On peut prendre par exemple :

$$E = \alpha E_{int} + \beta E_{ext} \quad (3.1)$$

L'énergie interne E_{int} est liée aux caractéristiques de la courbe qui forme le contour et elle contrôle la régularité de la courbe.

L'énergie externe E_{ext} est liée aux caractéristiques de l'image.

3.2.2 Notre approche

Nous avons utilisé des contours actifs basés régions plutôt que des contours actifs basés contours. En effet, ces derniers sont plus sensibles à l'initialisation du contour.

La méthode utilisée pour la pré-détection étant basée sur des algorithmes très simples et rapides, nous avons voulu utiliser les mêmes concepts pour la partie de segmentation. Cette approche par contours actifs a d'abord été utilisée par Réfrégier et al dans [25] puis généralisée dans [6]. Elle consiste à tester différentes hypothèses de formes et à sélectionner la meilleure. Nous avons légèrement adapté l'énergie et l'algorithme d'optimisation, mais le coeur de la méthode reste le même que celui présenté dans ces articles.

Pour chacune des pré-détections, nous allons extraire une sous-image centrée sur cette pré-détection et de taille $L_B \times L_B$ pixels puis nous allons tenter de segmenter en deux régions : la cible et le fond.

Nous gardons toujours deux hypothèses H_0 et $H_{1,w}$ (figure 3.1) :

$$\left\{ \begin{array}{l} H_0 : \text{la sous-image est composée uniquement du fond.} \\ \quad \text{la région correspondante est notée } B. \\ H_{1,w} : \text{la sous-image est composée de deux parties :} \\ \quad - \text{ une cible de forme } w, \\ \quad - \text{ une région de fond } \bar{w} \text{ qui est le complémentaire de } w \text{ dans la sous-image.} \end{array} \right.$$

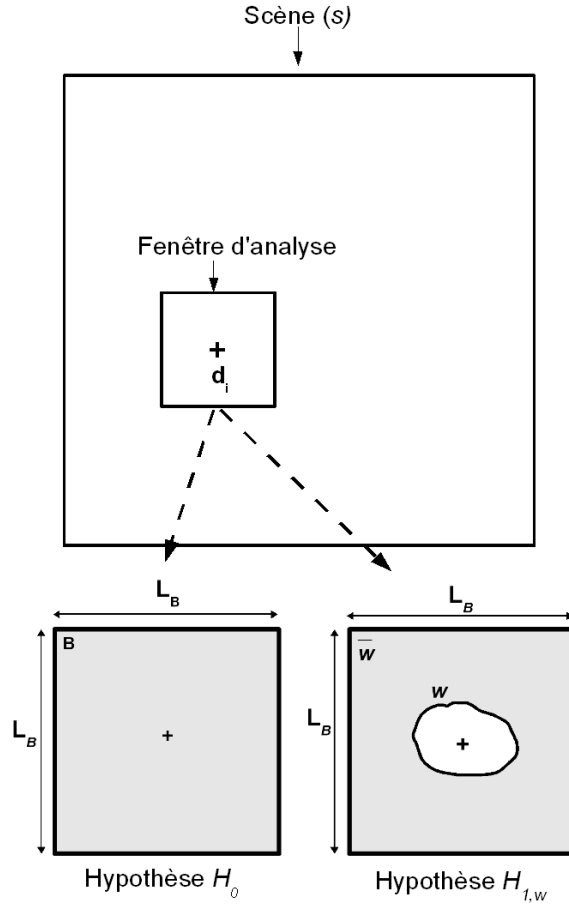


FIG. 3.1 – Modèle de la scène (s) autour d'une pré-détection. La position de la pré-détection est marquée par la croix. Une sous-image de taille $L_B \times L_B$ pixels est extraite autour de cette position. La cible est matérialisée par la région blanche et le fond par la région grisée. Le fond et la cible sont modélisés par des distributions de probabilité, des niveaux de gris, blanches et gaussiennes.

Ainsi, nous pouvons tester différentes formes w pour la cible (figure 3.2).

Le modèle statistique utilisé est le même que dans la partie 2.2.3 : le fond et la cible suivent une distribution gaussienne blanche de moyenne différente, mais de même variance. Comme le modèle statistique de la scène est le même que dans la partie 2.2.3, le calcul du rapport de vraisemblance est identique.

Pour chaque forme w , nous calculons $\log\Lambda = L(w) + L(\bar{w}) - L(B)$ comme dans l'équation 2.16 que nous rappelons ici en prenant en compte les nouvelles notations :

$$\log\Lambda(w) = N_w \hat{m}_w^2 + N_{\bar{w}} \hat{m}_{\bar{w}}^2 - N_B \hat{m}_B^2 \quad (3.2)$$

avec les notations suivantes :

- $N_w, N_{\bar{w}}, N_B$: nombre de pixels des régions w, \bar{w}, B .
- $\hat{m}_w, \hat{m}_{\bar{w}}, \hat{m}_B$: estimation de la moyenne des régions w, \bar{w} et B .

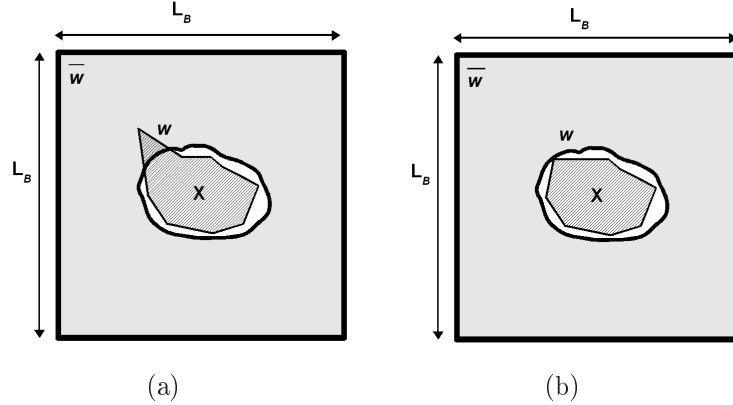


FIG. 3.2 – Le fond est représenté par la région en grisée et la cible par la région blanche. La forme candidate est représentée par la zone hachurée. Les deux sous-figures montrent deux possibilités de forme candidate w pour la même sous-image.

On considère que la forme qui donne la valeur maximale pour cette quantité est la forme réelle de la cible.

3.2.3 Énergie utilisée

Énergie externe

Nous utiliserons comme énergie externe le rapport de vraisemblance. Dans les concepts des contours actifs, la recherche du contour est guidée par une minimisation de l'énergie. Précédemment, nous avons vu que le rapport de vraisemblance généralisé doit être maximisé. Nous prendrons donc comme énergie, l'opposé du rapport de vraisemblance :

$$E_{Ext}(w) = -(N_w \hat{m}_w^2 + N_{\bar{w}} \hat{m}_{\bar{w}}^2 - N_B \hat{m}_B^2) \quad (3.3)$$

avec les notations suivantes :

- $N_w, N_{\bar{w}}, N_B$: nombre de pixels des régions w, \bar{w}, B .
- $\hat{m}_w, \hat{m}_{\bar{w}}, \hat{m}_B$: estimation de la moyenne des régions w, \bar{w} et B .

Par rapport aux travaux de Réfrégier et al [25, 6], nos contributions sont simples, mais particulièrement efficaces :

- nous utilisons l'hypothèse supplémentaire (partie 2.2.5) que la variance est identique entre le fond et la cible. Comme dans le chapitre sur la pré-détection, nos tests (non présentés ici) ont montré que l'estimation des paramètres de variance dégrade également les performances
- nous avons ajouté le terme qui correspond à l'hypothèse H_0 ($L(B)$) qui normalise l'énergie par rapport à l'intensité de la fenêtre. Nous verrons l'intérêt de cet ajout dans le chapitre 4.

Energie Interne

Dans la publication initiale de Kass [45], l'énergie totale fait intervenir également une énergie interne qui est liée aux caractéristiques de la courbe (notamment sa courbure) qui constitue le contour. Energie interne et énergie externe pouvant être sommées de façon pondérée pour guider au mieux la segmentation, le facteur de pondération étant fixé de façon empirique.

Dans le chapitre précédent, nous avons vu que nous pouvions avoir des pré-détections au niveau des nuages. Cela implique que lors de la segmentation, nous aurons à la fois des bateaux, mais aussi d'autres objets à segmenter. Dans ces conditions, il n'est pas simple de définir des contraintes sur le contour et nous avons choisi de lui donner la plus grande liberté possible (afin qu'il puisse s'adapter à différents objets) en n'utilisant pas d'énergie interne. L'avantage supplémentaire est qu'il n'est plus nécessaire de fixer les deux coefficients α et β de l'équation 3.1.

Energie totale

L'énergie totale est donc égale à l'énergie externe : $E(w) = E_{Ext}(w)$.

L'énergie totale pour une forme w est donc définie par :

$$\mathbf{E}(\mathbf{w}) = -(\mathbf{N}_w \hat{m}_w^2 + \mathbf{N}_{\bar{w}} \hat{m}_{\bar{w}}^2 - \mathbf{N}_B \hat{m}_B^2) \quad (3.4)$$

avec les notations suivantes :

- $N_w, N_{\bar{w}}, N_B$: nombre de pixels des régions w, \bar{w}, B .
- $\hat{m}_w, \hat{m}_{\bar{w}}, \hat{m}_B$: estimation de la moyenne des régions w, \bar{w} et B .

3.2.4 Algorithme d'optimisation

Un problème majeur pour notre recherche de minimum est que l'espace des formes possibles est vaste. De plus, pour notre application, le temps de calcul est une préoccupation importante. La réduction de ce temps de calcul est réalisée en réduisant l'espace des formes possibles et en testant uniquement une petite partie d'entre elles.

Nous avons considéré uniquement des formes polygonales. Ce qui permet un contrôle très simple des formes en utilisant les noeuds du polygone.

Nous avons donc besoin d'un algorithme d'optimisation qui explorera une petite partie des formes polygonales et trouvera la solution qui minimise l'énergie proposée dans

l'équation 3.4.

L'algorithme utilisé s'appuie sur deux méthodes connues : descente de gradient et algorithme de Monte Carlo de température nulle. Les noeuds du contour sont déplacés un par un pour changer la forme candidate et nous mettons à jour la forme si une meilleure forme est trouvée. Le contour actif est initialisé avec un carré de 3×3 pixels, qui correspond à la taille de la cible (L_w) que nous avons utilisé dans l'étape de pré-détection (partie 2.2.9).

On a donc l'algorithme suivant :

1. $w_{current} = w_0$, w_0 est un carré de 3×3 pixels défini par 4 noeuds.
2. Calcul de $E(w_{current})$.
3. Sélection d'un noeud et déplacement de ce noeud. Nous obtenons une nouvelle forme w_{test} .
4. Calcul de $E(w_{test})$.
5. Si $E(w_{test}) < E(w_{current})$ alors $w_{current} = w_{test}$.
6. Nous réitérons les étapes 3,4,5

Nous avons évalué plusieurs variantes de cet algorithme dans lequel l'étape 3 peut être réalisée de manière stochastique ou déterministe. Par exemple, la sélection du noeud à déplacer peut être faite au hasard ou dans un ordre séquentiel. Le déplacement pour le noeud peut être tiré au hasard (en direction et en distance) ou nous pouvons tester tous les déplacements possibles dans une petite fenêtre autour de la position actuelle du noeud.

Parmi les méthodes évaluées, nous avons des résultats très intéressants et un temps de calcul correct avec la méthode suivante :

- On tire un déplacement au hasard (direction et distance (≤ 2 pour privilégier les petits déplacements)).
- On applique successivement ce déplacement à chaque noeud du contour ce qui fournit un ensemble de nouveaux contours.
- Pour chaque nouveau contour, on calcule l'énergie.
- Parmi tous les contours qui induisent une diminution de l'énergie, on choisit celui qui donne le meilleur résultat.

Nous avons également ajouté une procédure entre deux itérations qui ajoute un noeud au milieu de deux noeuds successifs si la distance est trop longue entre eux. Ainsi, nous obtenons des formes plus précises. Le seuil pour lequel on considère la distance trop grande a été fixé à 10 pixels. C'est un compromis qui permet d'obtenir des formes précises sans avoir trop de points pour le contour.

Comme nous pouvons le constater dans la figure 3.5, de très bons résultats sont obtenus avec moins de 100 itérations.

3.2.5 Couplage de la pré-détection et de la segmentation

Nous allons d'abord décrire comment la pré-détection et la segmentation sont couplées. Le couplage est évident dans le cas d'une cible réelle : la position de la pré-détection est utilisée comme centre du contour initial (w_0). En revanche, quand la cible pré-détectée fait partie du bord d'un nuage ou d'un gros bateau ce couplage s'avère plus complexe. En effet, les fausses alarmes sont alors très près les unes des autres et chacune est considérée comme une cible potentielle.

Nous avons associé à chaque cible pré-détectée le rapport de vraisemblance généralisé obtenu à l'étape de pré-détection (partie 2.2.8). Cette mesure est étroitement liée à la probabilité que la cible soit un petit bateau de taille $L_w \times L_w$ pixels. La liste des cibles pré-détectées est donc classée par ordre décroissant du rapport de vraisemblance généralisé.

On peut alors segmenter la première cible pour obtenir sa forme. Pour accélérer le traitement, toutes les autres cibles pré-détectées, dont la position est incluse à l'intérieur du contour de cette première cible, peuvent être éliminées car il s'agit probablement du même objet. On peut alors réitérer ce processus jusqu'à ce que la liste des cibles à traiter soit vide.

De plus, dans le cas des nuages ou des gros bateaux, les détections peuvent être proches et les résultats des différentes segmentations auront tendance à se superposer. Il faut donc détecter ces superpositions et les fusionner en réalisant une union des contours, pour former un seul objet comme cela est illustré dans la figure 3.3.

3.2.6 Exemples

Segmentation des petites cibles

Pour la segmentation des petits bateaux le nombre d'itérations est assez faible. En effet, nous pouvons constater dans la figure 3.4 qu'avec 17 itérations nous obtenons un résultat très satisfaisant.

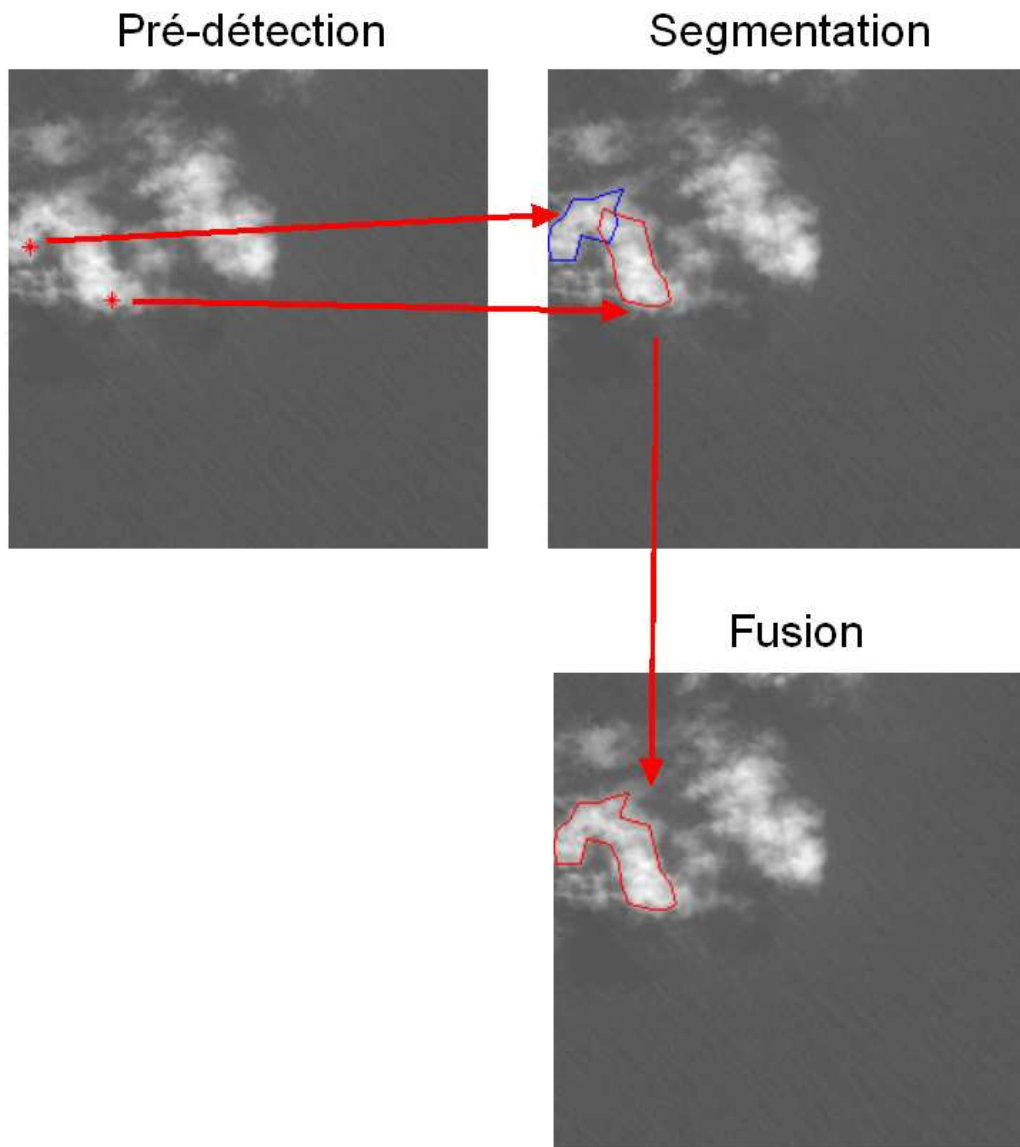


FIG. 3.3 – Exemple de segmentation d'un nuage avec fusion des contours.

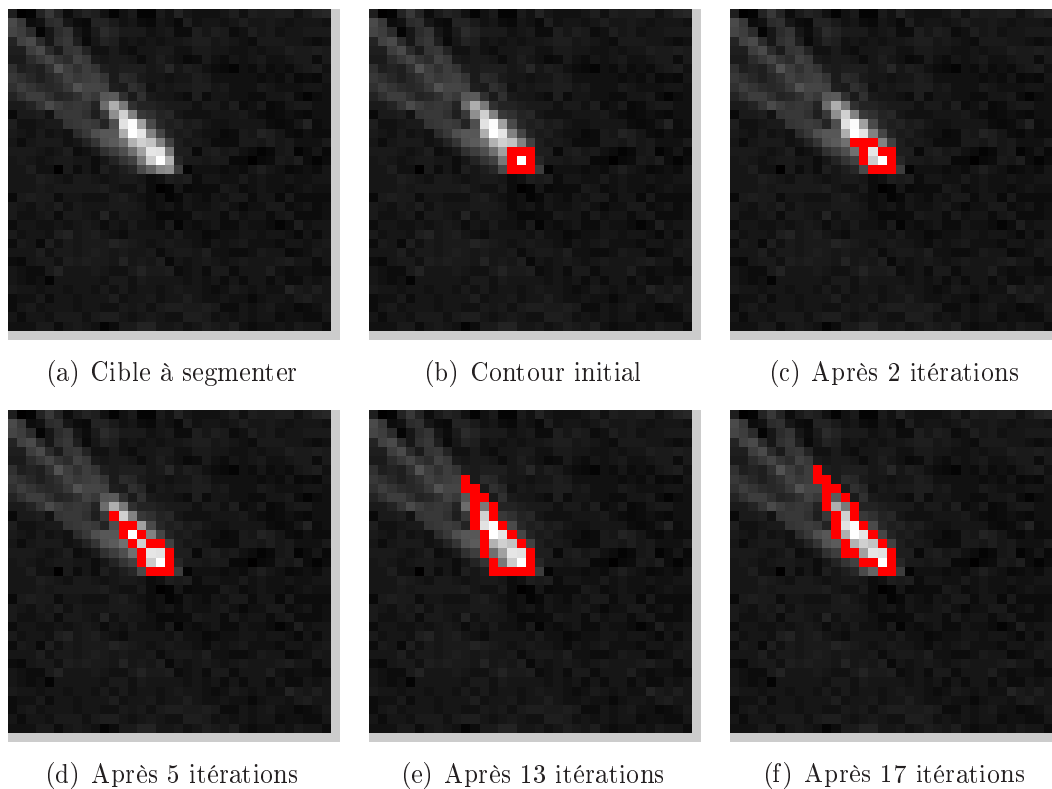


FIG. 3.4 – Exemple de segmentation d'un petit navire. 17 itérations ont été nécessaires pour segmenter cette cible.

Segmentation des grandes cibles

Pour la segmentation des grands bateaux, le nombre d'itérations doit être plus important. Nous voyons, dans l'exemple de la figure 3.5, qu'il faut au moins 80 itérations pour obtenir une segmentation complète de ce navire de grande taille.

Nous fixerons donc à 100 le nombre maximum d'itérations dans la suite de nos expérimentations.

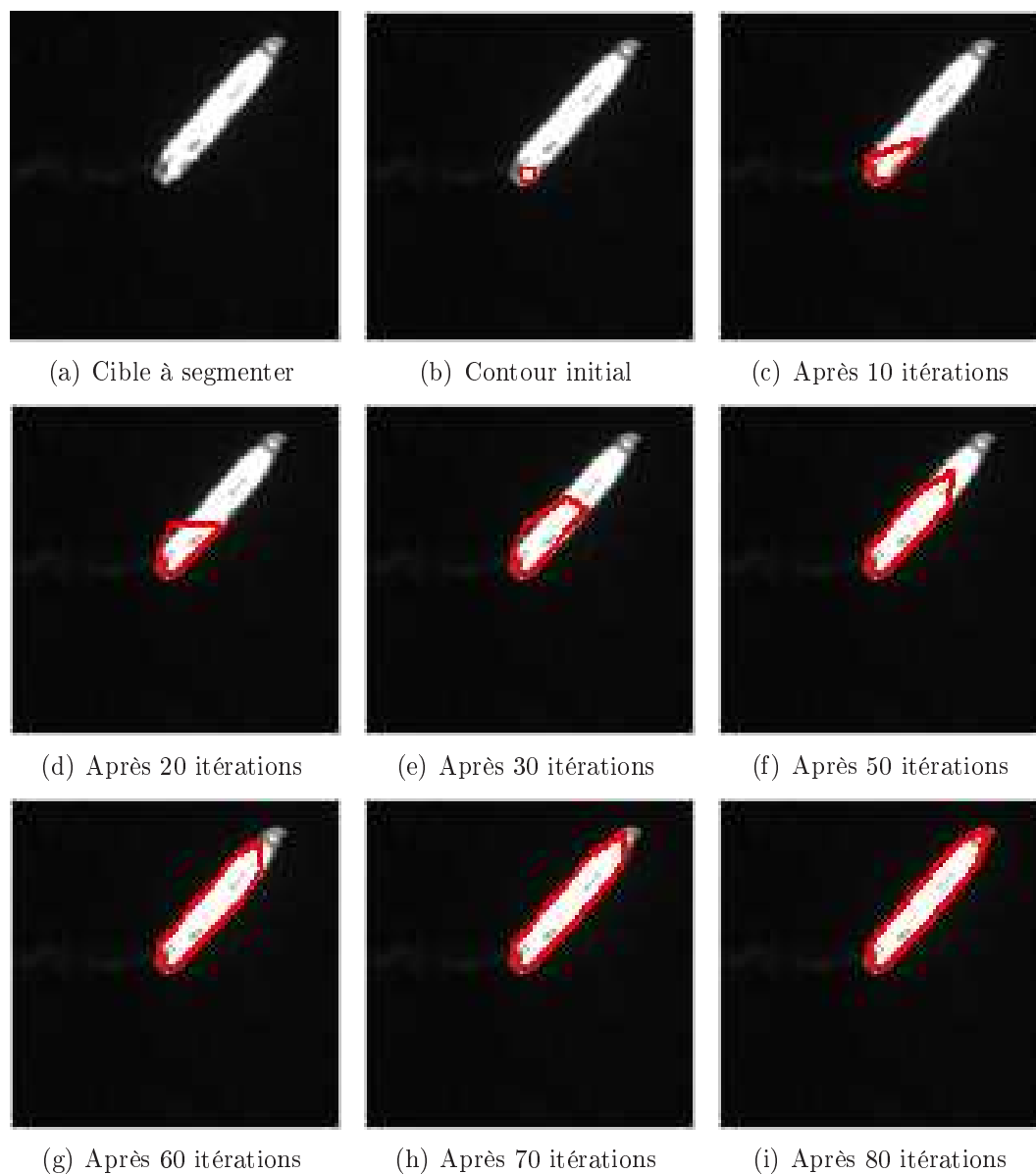


FIG. 3.5 – Exemple de segmentation d'un grand navire. 80 itérations sont nécessaires pour segmenter la cible.

3.3 La classification

L'étape de segmentation nous fournit un ensemble de contours localisés qui correspond à des détections potentielles de bateaux. Nous voulons maintenant associer à chaque détection une classe : petit bateau, grand bateau ou tout ce qui n'est pas un bateau.

Pour classifier les contours issus de la segmentation et de l'image, nous allons d'abord extraire un vecteur de caractéristiques à partir du contour et de l'image. Puis nous allons prendre une décision sur la base de ces caractéristiques (figure 3.6).

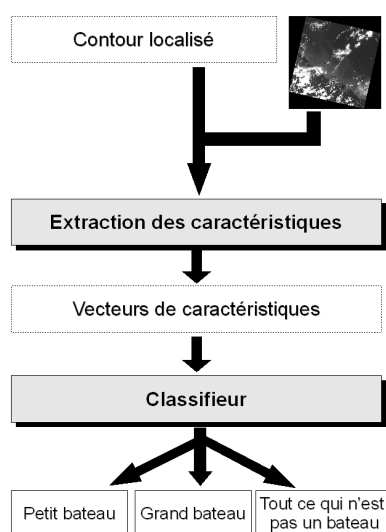


FIG. 3.6 - Les étapes de la classification.

3.3.1 Extraction des caractéristiques

Nous avons extrait l'ensemble des caractéristiques à partir des contours et du contenu de l'image. Nous présentons ici toutes les caractéristiques extraites et utilisées lors de la recherche d'un classifieur adapté à notre problématique (pour chaque classifieur testé nous avons sélectionné les caractéristiques les mieux adaptées), mais toutes ne seront pas utilisées dans le classifieur final.

Afin de s'affranchir dans cette partie de la variabilité des conditions d'acquisition, il nous a paru important de limiter les caractéristiques liées à l'intensité des pixels et de privilégier les caractéristiques liées à la taille et à la forme de l'objet.

Nous avons constitué trois groupes de caractéristiques :

- celles liées à l'intensité des pixels,
- celles liées à la forme du contour,

– celles liées à la texture.

Nous noterons C le contour obtenu par l'étape de segmentation et R la région correspondante.

Les caractéristiques liées aux pixels de la région R :

1. Le nombre de pixels de la région R .
Cette caractéristique donne une indication de taille pour la région.
2. La moyenne des pixels de la région R .
Cette caractéristique donne une indication sur l'intensité des pixels de la région.
3. L'écart type de la région R .
Cette caractéristique donne une indication sur l'homogénéité dans la région.
4. La différence de moyenne des pixels entre l'intérieur de la région et l'extérieur de la région. Nous prenons pour cela un rectangle englobant autour du contour. Cette caractéristique donne une indication sur le contraste entre les pixels de la région et les pixels à l'extérieur de la région.

Les caractéristiques liées à la forme du contour C :

1. L'aire de la région R (A).
Cette caractéristique est redondante avec le nombre de pixels de la région mais elle est utilisée pour le calcul d'autres caractéristiques.
2. Le périmètre du contour C (P).
Cette caractéristique donne une indication sur la taille de la région et elle est utilisée pour le calcul d'autres caractéristiques.
3. Le coefficient de compacité (K) de la région R .
Cette caractéristique donne une indication de forme de la région en caractérisant l'aspect compact ou allongé de celle-ci.

Le coefficient de compacité de la région proposé par Gravelius en 1914 [27] est défini par le rapport entre le périmètre d'une région et celui du cercle de même surface que la région :

$$K = \frac{P}{2\sqrt{\pi A}} \quad (3.5)$$

où P est le périmètre de la région R et A est l'aire de la région R .

Plus la région sera allongée, plus le coefficient de compacité sera élevé.

Le tableau 3.1 présente le coefficient de compacité de quelques objets présents dans les images étudiées.

4. L'excentricité (e) de l'ellipse englobante du contour C .

Ce paramètre va caractériser l'aplatissement du contour. Pour une ellipse, elle est comprise entre 0 et 1. Plus elle est proche de 1 plus la forme est aplatie.

Le tableau 3.1 présente l'excentricité de l'ellipse englobante de quelques objets présents dans les images étudiées.

5. La longueur (L) et la largeur (l) du contour C .

Ces deux caractéristiques donnent des indications sur la taille du contour. Elles sont utilisées pour le calcul d'autres caractéristiques.

Pour déterminer la longueur et la largeur nous allons d'abord définir l'orientation du contour C (θ). Ensuite, à partir de la droite (D) passant par le centre du contour C et d'orientation θ , nous calculons la longueur (L). En prenant la perpendiculaire à (D) nous calculons la largeur (l) comme cela est illustré dans la figure 3.7.

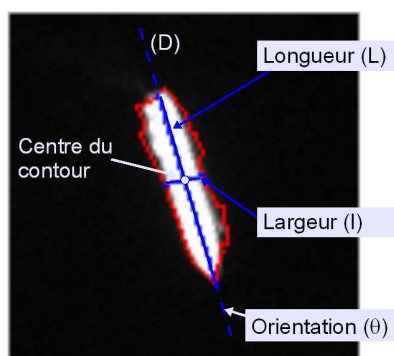


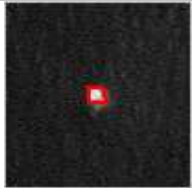
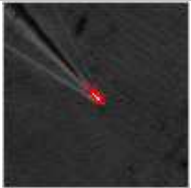
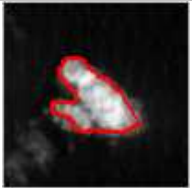

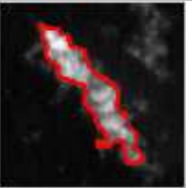
FIG. 3.7 – La longueur (L) et la largeur (l) du contour C (en rouge).

6. Le rapport largeur/longueur du contour C ($R_{l/L}$).

Ce rapport caractérise l'allongement de l'objet. Le tableau 3.1 présente le rapport largeur/longueur de quelques objets présents dans les images étudiées.

7. Le rapport entre l'aire de la région R et l'aire de la zone convexe de cette région (R_c).

Ce rapport met en évidence les formes qui sont irrégulières (avec des creux). La figure 3.8 illustre deux exemples de ce rapport : pour un grand bateau 3.8(a) et un nuage de forme allongée 3.8(b). Plus on va avoir des creux dans la forme, plus le

| | | | | | |
|-----------|---|---|---|--|---|
| |  |  |  |  |  |
| K | 1,0708 | 1,2647 | 1,4026 | 1,5220 | 2,0023 |
| e | 0,6052 | 0,9355 | 0,7742 | 0,9710 | 0,9799 |
| $R_{l/L}$ | 0,7961 | 0,3533 | 0,6329 | 0,2390 | 0,1997 |

TAB. 3.1 – Coefficient de compacité (K), excentricité (e) et rapport largeur/longueur ($R_{l/L}$) sur quelques exemples.

rapport sera faible. Cela devrait permettre de distinguer les nuages qui ont souvent une forme chaotique alors qu'un grand navire aura des contours plus réguliers donc un rapport plus proche de 1.

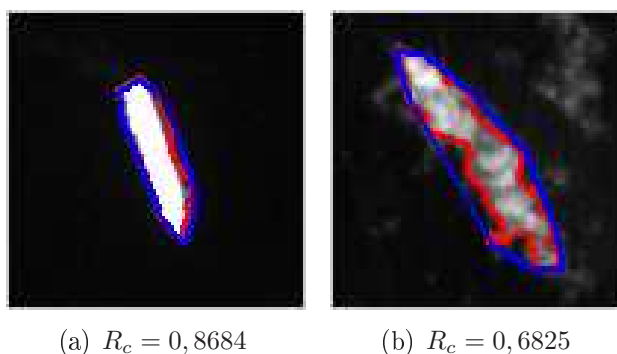


FIG. 3.8 – Valeur de R_c pour deux exemples de contours. Le contour C est en rouge et le contour convexe est en bleu.

8. Le recouvrement pour différents cercles.

Cette caractéristique est empirique. Elle caractérise la forme du contour (plutôt allongé ou arrondi) en prenant également en compte un critère de taille (en prenant plusieurs rayons possibles).

Soit C_r le cercle de rayon r et de centre : le centre de la région R (figure 3.9(a)).

Soit I l'intersection entre le cercle C_r et la région R .

Le taux de recouvrement est défini par :

$$\frac{\text{Aire}(I)}{\text{Aire}(C_r)}$$

Le rayon r varie entre 1 et 10.

9. Le recouvrement pour le cercle englobant la région R .

Cette caractéristique est également empirique. Elle caractérise la forme du contour : plutôt allongé ou arrondi.

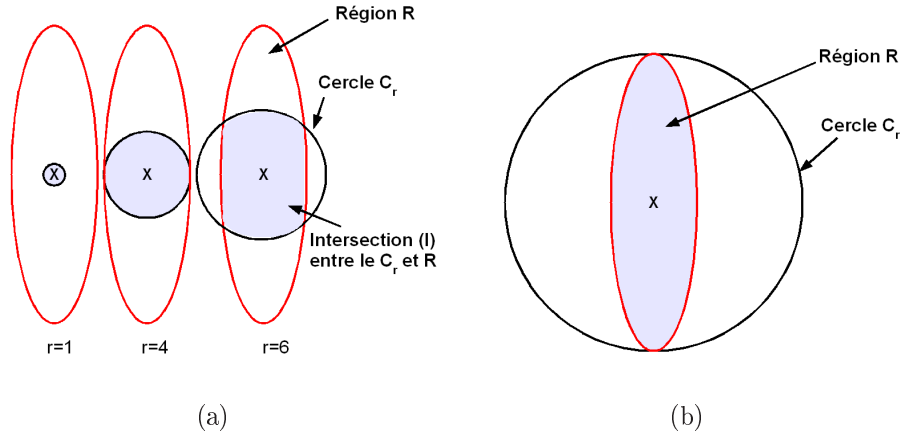


FIG. 3.9 – Recouvrement pour différents cercles.

Soit C_E le cercle englobant la région R (figure 3.9(b)).

Le recouvrement est défini par :

$$\frac{\text{Aire}(R)}{\text{Aire}(C_E)}$$

10. Les moments de Hu.

Hu [36], en 1962, est le premier à présenter des travaux sur l'analyse d'images en utilisant les moments géométriques. Les moments de Hu sont au nombre de sept et sont invariants par rotation, translation et changement d'échelle. Le changement d'échelle aurait pu poser des problèmes pour la distinction entre un petit et un grand bateau, mais en combinant avec des caractéristiques de taille, nous pouvons facilement faire cette distinction.

Ils sont calculés de la façon suivante :

Soit $f(x, y)$ la valeur d'un pixel de l'image, on définit le moment d'ordre $(p+q)$ ($p, q > 0$) par :

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad (3.6)$$

On définit le moment central d'ordre $(p+q)$ ($p, q > 0$) par la fonction suivante :

$$v_{pq} = \sum_x \sum_y (x - x_0)^p (y - y_0)^q f(x, y) \quad (3.7)$$

$$\text{où } x_0 = \frac{m_{10}}{m_{00}} \text{ et } y_0 = \frac{m_{01}}{m_{00}}$$

Puis on introduit le moment normalisé d'ordre $(p+q)$ ($p, q > 0$) par la fonction suivante :

$$\mu_{pq} = \frac{v_{pq}}{v_{00}^{1+(p+q)/2}} \quad (3.8)$$

Enfin on définit les sept moments de Hu de la manière suivante :

$$M1 = \mu_{20} + \mu_{02} \quad (3.9)$$

$$M2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \quad (3.10)$$

$$M3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \quad (3.11)$$

$$M4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \quad (3.12)$$

$$\begin{aligned} M5 = & (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\ & + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \end{aligned} \quad (3.13)$$

$$M6 = (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}) \quad (3.14)$$

$$\begin{aligned} M7 = & (3\mu_{21} - \mu_{03})(\mu_{03} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\ & - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \end{aligned} \quad (3.15)$$

Les moments de Hu ont été utilisés pour la reconnaissance d'objets dans [48] et plus particulièrement pour la reconnaissance de bateaux dans des images de type FLIR (Forward Looking Infrared) dans [1, 68].

Nous aurions pu utiliser les moments de Zernike [59] qui donnent de bons résultats pour des formes complexes et bien définies. Nous avons réalisé des tests avec les moments de Zernike, mais les résultats étant moins bons qu'avec les moments de Hu, nous avons donc préféré ne pas les utiliser. Nous pensons que ces moins bons résultats sont liés au fait qu'ils sont trop précis par rapport à nos petits bateaux et leur forme imprécise.

Les caractéristiques liées à la texture

Les paramètres de texture vont permettre de caractériser les zones nuageuses ou la mer par rapport aux bateaux.

Haralick [34, 33] propose une caractérisation des textures basée sur les matrices de cooccurrence des niveaux de gris dont les éléments représentent l'estimation de la probabilité de transition d'un niveau de gris à un autre selon une direction donnée avec une distance entre les pixels donnée. Elles ont été utilisées pour la classification des nuages dans [67].

Nous avons utilisé des paramètres calculés à partir de la matrice de cooccurrence des niveaux de gris. Les objets à classifier ne présentant pas de textures très complexes nous avons uniquement utilisé les paramètres de base.

La matrice de cooccurrence des niveaux de gris est définie de la manière suivante :

L'élément $G_{\theta,d}(i, j)$ représente le nombre de pixels de niveau de gris i ayant pour voisin un pixel de niveau de gris j à la distance d dans la direction θ .

Nous avons utilisé la matrice de cooccurrence de paramètre $\theta=0$ et $d=1$ que nous noterons G dans la suite.

soit $G(i, j)$ l'élément (i, j) de la matrice de co-occurrence G .

On peut alors définir les paramètres de texture suivants :

1. Le contraste défini par :

$$G_c = \sum_{i,j} G(i, j)(i - j)^2 \quad (3.16)$$

2. L'homogénéité définie par :

$$G_h = \sum_{i,j} \frac{G(i, j)}{1 + (i - j)^2} \quad (3.17)$$

3. La corrélation normalisée définie par :

$$G_{cn} = \sum_{i,j} \frac{G(i, j)(i - \mu_x)(j - \mu_y)}{\sigma_x \sigma_y} \quad (3.18)$$

où μ_x et μ_y représentent respectivement les moyennes des lignes et des colonnes de la matrice G ; σ_x et σ_y représentent respectivement les écarts types des lignes et des colonnes de la matrice G .

4. L'énergie définie par :

$$G_e = \sum_{i,j} G(i, j)^2 \quad (3.19)$$

5. L'entropie définie par :

$$E = - \sum_i p_i \ln(p_i) \quad (3.20)$$

où p_i est la probabilité du niveau de gris i .

3.3.2 Les échantillons d'apprentissage et de généralisation

A partir de la base d'images 2 décrite dans la partie 1.6.2, nous avons constitué les échantillons d'apprentissage et de généralisation (figure 3.10). Nous avons attribué une classe aux 533 éléments issus du traitement pré-détection + segmentation.

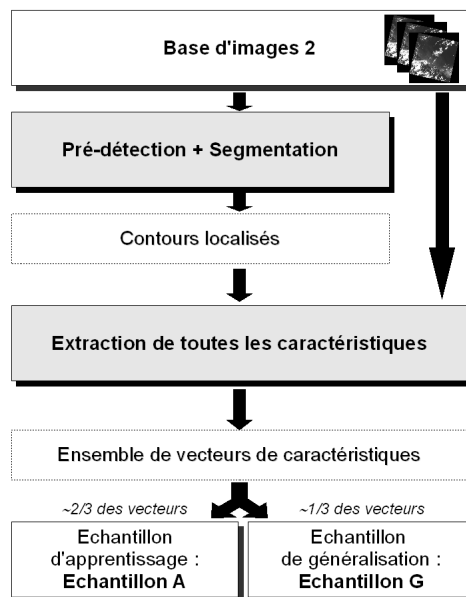


FIG. 3.10 – Démarche pour la constitution des échantillons d'apprentissage et de généralisation.

Dans un premier temps, nous avons utilisé trois classes : petit bateau, grand bateau, tout ce qui n'est pas un bateau. Les résultats n'étant pas satisfaisants, nous avons supposé que c'était lié au fait que la troisième classe contient des éléments disparates (petits nuages, gros nuages, crêtes des vagues). Nous avons donc choisi de diviser cette dernière classe en neuf classes homogènes. Finalement nous avons onze classes décrites dans le tableau 3.2 et illustrées dans la figure 3.11. Pour la classification finale, nous regrouperons les classes de 3 à 11 en une seule classe 0 (tout ce qui est autre qu'un bateau).

Puis, pour chaque classe, nous avons tiré aléatoirement des éléments pour constituer l'échantillon d'apprentissage (approximativement 2 éléments sur 3) et l'échantillon de généralisation (approximativement 1 élément sur 3). Nous noterons **Echantillon A** l'échantillon d'apprentissage et **Echantillon G** l'échantillon de généralisation.

3.3.3 Le classifieur

Pour rechercher une méthode efficace pour notre échantillon d'apprentissage, nous avons utilisé le logiciel **Weka** (Waikato Environment for Knowledge Analysis) qui est composé d'outils de manipulation et d'analyse des fichiers de données et qui implémente de nombreux algorithmes de fouille de données. Weka s'appuie sur [69].

A l'aide du logiciel Weka nous avons testé plusieurs classifieurs afin d'en retenir un. Nous avons testé plusieurs méthodes à base d'arbres de décision : *Logistic Model Trees* [39, 40], *Functional Trees* [23, 54], *unpruned or pruned C4.5 decision trees* [52] et une méthode

| Classe | Contenu | Illustration | Taille Echantillon A | Taille Echantillon G |
|--------|--------------------------------------|--------------|-------------------------|-------------------------|
| 1 | petit bateau | 3.11(a) | 143 | 69 |
| 2 | grand bateau | 3.11(b) | 8 | 4 |
| 3 | mer calme | 3.11(c) | 8 | 4 |
| 4 | grosse portion d'un nuage | 3.11(d) | 16 | 8 |
| 5 | nuage plutôt arrondi | 3.11(e) | 76 | 37 |
| 6 | nuage allongé | 3.11(f) | 3 | 1 |
| 7 | nuage irrégulier | 3.11(g) | 6 | 3 |
| 8 | petite portion d'un nuage | 3.11(h) | 34 | 16 |
| 9 | fond au milieu d'un nuage | 3.11(i) | 27 | 13 |
| 10 | mer agitée avec des moutons | 3.11(j) | 24 | 11 |
| 11 | mer agitée avec des crêtes de vagues | 3.11(k) | 15 | 7 |

TAB. 3.2 – Inventaire des différentes classes et nombre d'éléments pour chacune.

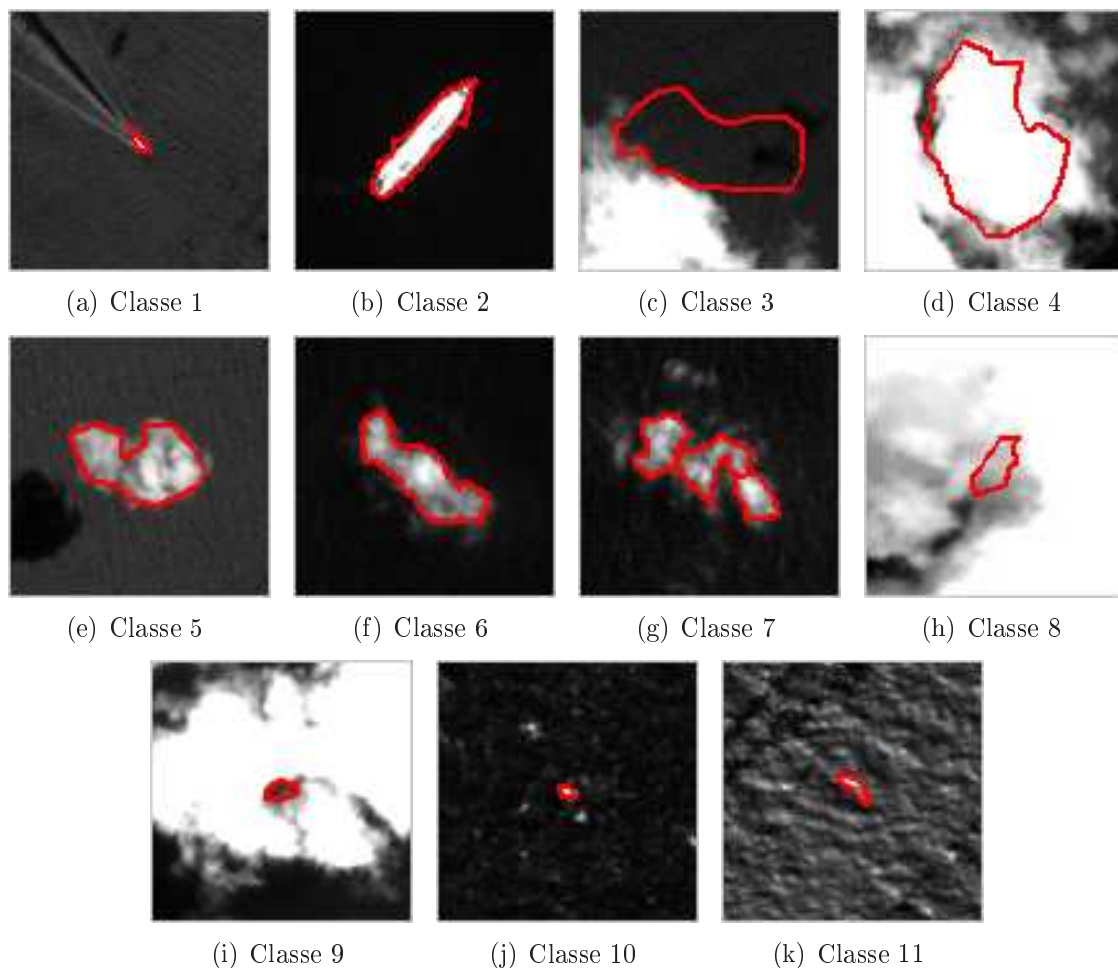


FIG. 3.11 – Un exemple pour chaque classe.

de classification bayésienne : *Bayes network classifier* [7, 4].

Nous avons également testé, de façon extensive, des combinaisons de caractéristiques afin de déterminer celles qui sont les mieux adaptées pour chaque classifieur.

Pour évaluer la qualité du classifieur, nous avons utilisé la matrice de confusion. Chaque colonne représente le nombre d'occurrences d'une classe estimée, tandis que chaque ligne représente le nombre d'occurrences d'une classe réelle (tableau 3.3). L'intérêt de la matrice de confusion est que nous pouvons voir facilement si le classifieur classe correctement.

| | | Classe estimée | | |
|---------------|--------------|----------------|--------------|------------|
| | | Petit bateau | Grand bateau | Pas bateau |
| Classe réelle | Petit bateau | | | |
| | Grand bateau | | | |
| | Pas bateau | | | |

TAB. 3.3 - *Matrice de confusion.*

Pour caractériser les performances par rapport à notre problématique de détection de bateaux, nous avons également utilisé :

- La probabilité de détection des petits bateaux calculée de la manière suivante :

$$PD_{petit} = \frac{N_{D_{petit}}}{N_{P_{petit}}}$$

où $N_{D_{petit}}$ est le nombre de petits bateaux détectés et $N_{P_{petit}}$ est le nombre de petits bateaux présents.

- La probabilité de détection des grands bateaux calculée de la manière suivante :

$$PD_{grand} = \frac{N_{D_{grand}}}{N_{P_{grand}}}$$

où $N_{D_{grand}}$ est le nombre de grands bateaux détectés et $N_{P_{grand}}$ est le nombre de grands bateaux présents.

- Le nombre de fausses alarmes Nb_{FA} qui est le nombre d'éléments classés comme étant des petits ou des grands bateaux, mais qui n'en sont pas.

Choix du classifieur

Nous avons testé les méthodes de classification citées précédemment, avec de nombreuses combinaisons de caractéristiques. A l'issue de ces tests, nous avons retenu la méthode qui a donné les meilleurs résultats sur notre échantillon de généralisation : *Functional Trees* (FT) [23, 54].

Cette méthode s'adapte à des paramètres nominaux ou continus. Elle peut être utilisée en régression comme en classification. Dans notre cas, nous allons l'utiliser pour des paramètres continus et en classification.

C'est un arbre de décision qui s'appuie en chaque noeud sur un classifieur que nous allons nommer classifieur de base. Ce classifieur de base est un classifieur linéaire bayésien [22]. Comme tous les classifieurs bayésiens, il utilise la probabilité conditionnelle :

$$P(C_i|x) = \frac{P(C_i)P(x|C_i)}{P(x)}$$

comme fonction discriminante. Pour estimer cette probabilité et éviter l'hypothèse d'indépendances des paramètres qui caractérise le classifieur bayésien naïf, le classifieur linéaire bayésien suppose que $P(x|C_i)$ suit une loi normale multivariée de la forme :

$$P(x|C_i) = \frac{1}{\sqrt{|\Sigma|}(2\pi)^{p/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma^{-1}(x-\mu_i)}$$

où μ_i est la moyenne de la classe C_i , Σ est la matrice de covariance et p est le nombre de paramètres du vecteur x . Le logarithme de cette probabilité est :

$$\log(P(C_i)) - \frac{1}{2}(x - \mu_i)^T \Sigma^{-1}(x - \mu_i)$$

à une constante près. Il est notable que, dans cette équation, la matrice de covariance est considérée identique pour l'ensemble des classes. Après quelques simplifications calculatoires, nous obtenons :

$$\log(P(C_i)) - \frac{1}{2}x^T \Sigma^{-1}x + \frac{1}{2}x^T \Sigma^{-1}\mu_i + \frac{1}{2}\mu_i^T \Sigma^{-1}x - \frac{1}{2}\mu_i^T \Sigma^{-1}\mu_i$$

Le terme $x^T \Sigma^{-1}x$ ne dépendant pas de la classe C_i , il peut être ignoré. En prenant en compte la symétrie de Σ , nous obtenons :

$$\log(P(C_i)) - \frac{1}{2}\mu_i^T \Sigma^{-1}\mu_i + x^T \Sigma^{-1}\mu_i$$

Les différents μ_i sont estimés sur l'ensemble des échantillons entrant dans chaque noeud et correspondant à la classe C_i . La matrice Σ est estimée sur l'ensemble des échantillons entrant dans le noeud.

En conclusion, le classifieur de base est un classifieur très simpliste. C'est la combinaison de ces classifieurs sous forme d'arbre qui fait la force du classifieur *Functional Trees*.

Un autre avantage des *Functional Trees* est qu'ils bénéficient des techniques de *boosting*. En effet, ils utilisent la méthode de *Additive logistic regression* [21]. Cette méthode reprend les mêmes concepts que la méthode *adaboost (adaptive boosting)* [20].

A la première étape, un sous-ensemble des échantillons est tiré aléatoirement avec la même probabilité. Nous construisons un *Functionnal Tree* sur la base de ces exemples. Une erreur de classification est calculée. Les échantillons mal classés voient leur probabilité d'être sélectionnés augmentée alors que les échantillons bien classés voient leur probabilité d'être sélectionnés diminuée. Nous réitérons ensuite ces étapes pour construire tout un ensemble de classifieurs. Tous ces classifieurs sont finalement recombinaison linéairement avec des poids décroissants au fur et à mesure des itérations.

Les détails des calculs des modifications de poids entre deux étapes ainsi que le poids de chaque classifieur dans la décision finale relèvent de considérations statistiques qui sont bien au-delà du cadre de cette thèse et qui pourront être trouvés dans [20]. Retenons simplement que la combinaison des arbres de décision et du *boosting* permet de dépasser les performances du classifieur bayésien linéaire de base. Cette méthode est rapide et simple à mettre en place.

Les caractéristiques finalement retenues sont : la moyenne des pixels de la région R, l'écart type de la région R, la différence de moyenne des pixels entre l'intérieur et l'extérieur de la région R, l'aire de la région R, le périmètre du contour C, le coefficient de compacité de la région R, la longueur et la largeur du contour C, le rapport largeur/longueur, le rapport entre l'aire de la région R et l'aire de la zone convexe de cette région, les recouvrements par différents cercle et le recouvrement par le cercle englobant la région R, les moments de Hu et les paramètres de textures : le contraste, la corrélation et l'entropie.

A partir de cette méthode de classification, nous avons envisagé deux approches :

1. **FT-1** : nous utilisons un classifieur défini pour classifier directement en 11 classes (figure 3.12(a)) .
2. **FT-2** : nous utilisons 9 « sous-classifieurs » en cascade (figure 3.12(b)) : Soit 9 classifieurs *Classifieur n°i* où $i \in [1, 2, 3, 4, 5, 6, 7, 8, 9]$ et C_i la classe obtenue avec ce classifieur. Chaque classifieur réalise la classification des classes 1,2 et d'une classe c où $c \in [3, 4, 5, 6, 7, 8, 9, 10, 11]$. Nous avons pour chacun sélectionné les caractéristiques qui donnaient les meilleurs résultats. Pour réaliser la classification d'un vecteur de caractéristiques v , nous procédons de la manière suivante :
 - Pour $i \in [1, 2, 3, 4, 5, 6, 7, 8]$ on classe v en utilisant le *Classifieur n° i* :
 - si $C_i \neq 1$ et $C_i \neq 2$ alors on arrête la classification et la classe est C_i ,
 - si $C_i = 1$ ou $C_i = 2$ alors on classe v en utilisant le *Classifieur n° i+1*.
 Ainsi, à chaque étape nous « éliminons » ce qui n'est pas un petit bateau ou un grand bateau.

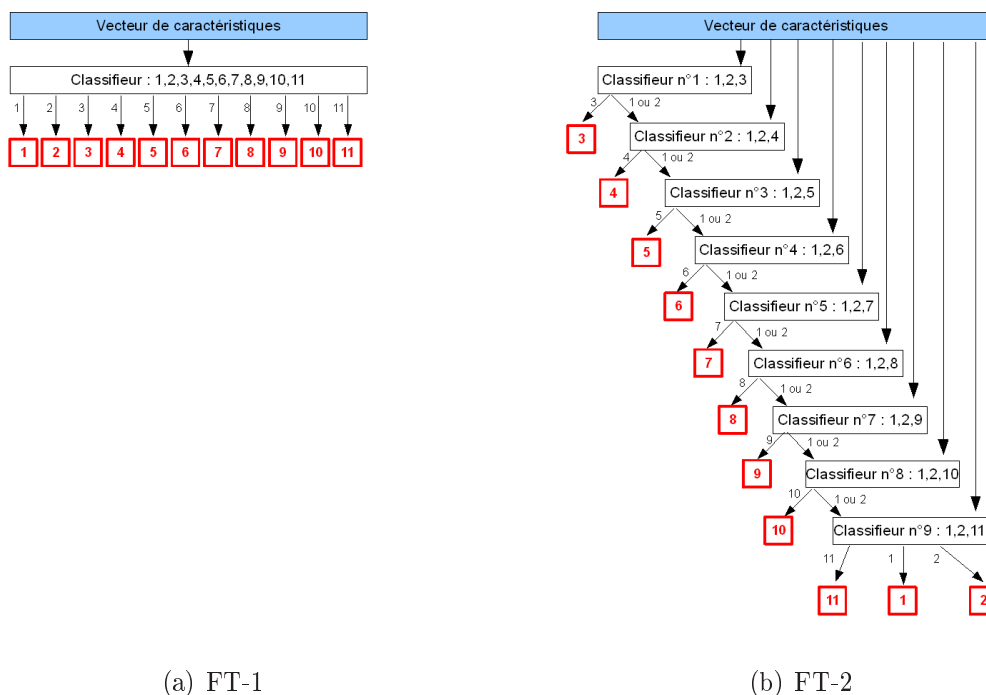


FIG. 3.12 – Description des deux approches : FT-1 et FT-2.

Pour nos deux approches **FT-1** et **FT-2**, nous donnons les résultats dans les tables 3.4 et 3.5.

| | | | FT-1 | | | FT-2 | | |
|----------------------|---------------|--------------|----------------|--------------|------------|----------------|--------------|------------|
| | | | Classe estimée | | | Classe estimée | | |
| | | | Petit bateau | Grand bateau | Pas bateau | Petit bateau | Grand bateau | Pas bateau |
| Echantillon A | Classe réelle | Petit bateau | 143 | 0 | 0 | 143 | 0 | 0 |
| | | Grand bateau | 0 | 8 | 0 | 0 | 8 | 0 |
| | | Pas bateau | 0 | 1 | 208 | 0 | 1 | 208 |
| Echantillon G | Classe réelle | Petit bateau | 64 | 1 | 4 | 67 | 0 | 2 |
| | | Grand bateau | 0 | 3 | 1 | 0 | 4 | 0 |
| | | Pas bateau | 14 | 1 | 85 | 14 | 2 | 84 |

TAB. 3.4 – Matrices de confusion pour **FT-1** et **FT-2**.

Sur l'échantillon A, les deux approches donnent les mêmes résultats. Par contre, si nous observons les résultats sur l'échantillon G, nous constatons que **FT-2** donne un meilleur résultat pour la classification des petits bateaux (67 pour **FT-2** contre 64 pour **FT-1**) et également pour la classification des grands bateaux (4 pour **FT-2** contre 3 pour **FT-1**). Par contre le nombre de fausses alarmes est légèrement supérieur pour **FT-2**. En ne perdant pas de vue que l'objectif final est de détecter le plus possible de bateaux, nous allons privilégier l'approche **FT-2** qui nous donne de meilleurs résultats en terme de probabilité de détection aussi bien pour les petits que pour les grands bateaux (tableau 3.5).

| | | FT-1 | FT-2 |
|----------------------|--------------|-------------|-------------|
| Echantillon A | PD_{petit} | 1 | 1 |
| | PD_{grand} | 1 | 1 |
| | Nb_{FA} | 1 | 1 |
| Echantillon G | PD_{petit} | 0,9275 | 0,9710 |
| | PD_{grand} | 0,7500 | 1 |
| | Nb_{FA} | 15 | 16 |

TAB. 3.5 – Performances des deux approches : FT-1 et FT-2.

Dans le tableau 3.6, nous allons maintenant montrer l'intérêt de la segmentation et la classification en terme de réduction des fausses alarmes. Les résultats sont présentés pour l'approche **FT-2** sur l'échantillon G. Nous constatons une réduction significative du nombre de fausses alarmes entre la phase de pré-détection et la phase de classification. Cela nous confirme l'intérêt de segmenter puis de classifier les pré-détections obtenues lors de la première étape.

| | FT-2 sur l'échantillon G |
|---|-------------------------------------|
| Nombre de pré-détections | 311 |
| Nombre de fausses pré-détections | 192 |
| Nombre de contours après ségmentation | 173 |
| Nombre de fausses détections après classification | 16 |

TAB. 3.6 – Synthèse des résultats en terme de nombre de fausses alarmes, avec l'approche FT-2 sur l'échantillon G.

3.4 Conclusion

Dans ce chapitre, nous avons étudié les parties segmentation et classification de notre approche décrite dans la chapitre 1 (figure 1.5.2).

A partir des pré-détections obtenues dans la première étape, nous réalisons une segmentation en utilisant des contours actifs dont l'énergie s'appuie sur la théorie de la décision bayésienne.

Après cette étape de segmentation, nous réalisons la classification des contours obtenus en trois classes : petit bateau, grand bateau, tout ce qui n'est pas un bateau. Pour cela, nous utilisons un arbre de décision multivarié nommé *Functional Tree*.

Nous avons tenté de classifier directement dans ces trois classes. Les résultats étant peu concluants, nous avons opté pour une autre solution dans laquelle nous utilisons neuf classes pour représenter « tout ce qui n'est pas un bateau » (nous avons ainsi des

classes plus homogènes). Nous avons donc à classifier en onze classes. Puis à l'issue de la classification, nous avons regroupé les neuf classes représentant les objets qui ne sont pas des bateaux en une seule, pour nous retrouver avec les trois classes désirées.

Nous avons testé deux approches : dans la première, nous avons classifié directement les onze classes (FT-1) et dans la seconde approche (FT-2), nous classifions par étape en utilisant à chaque étape un classifieur à trois classes (petit bateau, grand bateau et une des neuf classes représentant "tout ce qui n'est pas un bateau") .

Nous avons comparé les deux approches à l'aide des matrices de confusion et des probabilités de détection et de fausse alarme. Nous avons retenu l'approche **FT-2**.

Nous avons également montré l'intérêt de cette étape de segmentation et de classification. En effet, nous avons une réduction significative du nombre de fausses alarmes entre la phase de pré-détection et la phase de classification. Les résultats obtenus montrent que la méthode de segmentation est globalement satisfaisante.

Pour valider ces résultats, nous avons testé l'approche **FT-2** sur la base d'images complètes (la base d'images 3 décrite dans le chapitre 1). Nous présentons les résultats dans le chapitre 5.

Chapitre 4

Détection des sillages

4.1 Introduction

Le sillage d'un bateau est la propagation à la surface de la mer des perturbations induites par celui-ci. Il apparaît comme une traînée à l'arrière du navire. La détection du sillage du bateau fournit des informations complémentaires concernant la direction de déplacement du navire. Comme nous l'avons vu dans le chapitre 1, la forme générale du sillage est triangulaire (figure 1.5). L'ouverture du sillage est variable et peut (si elle est suffisamment faible) amener à considérer le sillage principal comme une droite.

Actuellement, dans OCEANWAY, l'opérateur humain ne fait aucune mesure sur le sillage des bateaux. Il l'utilise juste pour déterminer une approximation de la direction de déplacement du navire.

A partir de l'observation des différents sillages, nous avons fait plusieurs constatations :

- Les sillages peuvent être soit clairs comme dans les figures 4.1(a) 4.1(b) 4.1(c) 4.1(d), soit foncés comme dans les figures 4.1(e) et 4.1(f).
- Les vagues d'étrave ne sont pas toujours visibles comme nous pouvons l'observer dans les figures 4.1(c) 4.1(d) 4.1(e) 4.1(f).
- Les sillages ne sont pas forcément rectilignes et peuvent présenter des courbures comme dans les figures 4.1(d) et 4.1(e).
- Les sillages se présentent parfois comme une zone homogène à l'arrière du bateau qui s'estompe rapidement comme dans la figure 4.1(c).
- La largeur des sillages est variable.

Nous allons maintenant présenter différentes approches possibles. L'approche classique [11] détecte dans un premier temps la cible puis à partir de celle-ci recherche dans son très proche voisinage son sillage (figure 4.2(a)). L'avantage de cette approche est qu'elle limite la zone de recherche du sillage à une zone proche du bateau, mais cela peut

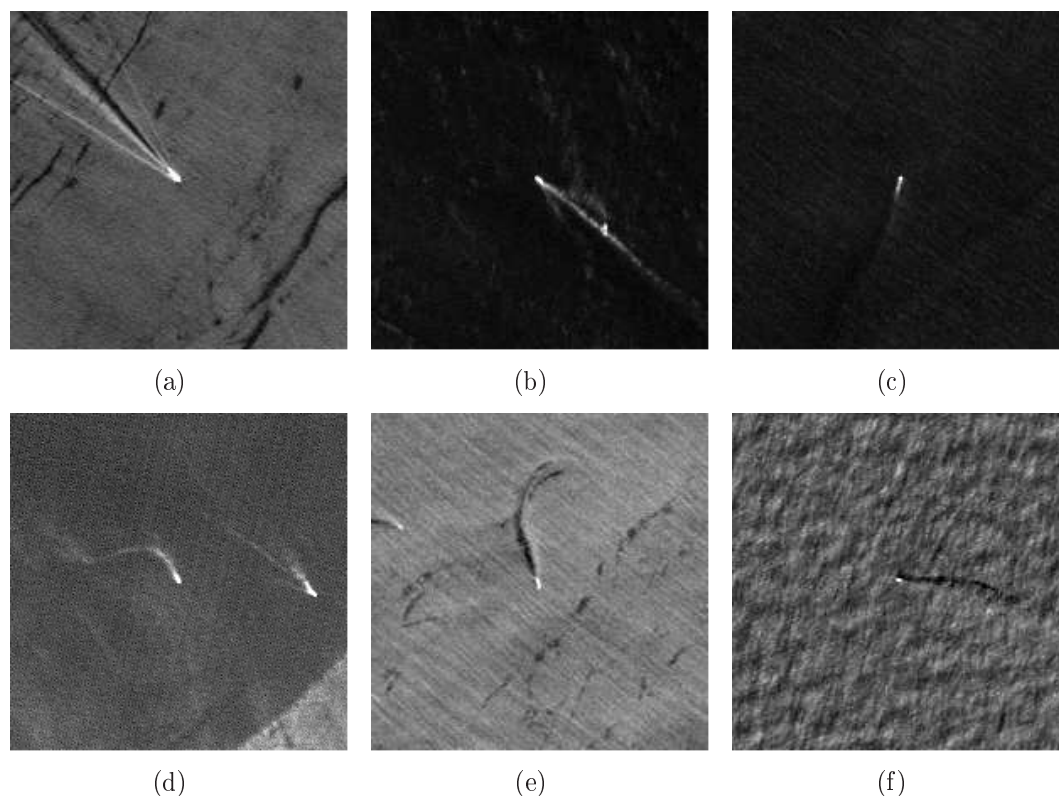


FIG. 4.1 – Exemples de différents types de sillages présents dans les images étudiées.

être un inconvénient lorsque le début du sillage est éloigné du bateau.

Une autre approche consiste à détecter les bateaux d'une part et les sillages potentiels d'autre part, puis de coupler les bateaux avec les sillages candidats (figure 4.2(b)). Dans cette approche, la détection du sillage ne se limite plus à une zone très proche du bateau (cela est intéressant lorsque le début du sillage est éloigné du bateau). Pour des raisons de temps de calcul, nous avons préféré considérer la détection du bateau comme le centre d'une région d'intérêt et de limiter la recherche des sillages à cette région comme cela est illustré dans la figure 4.2(c).

Dans [2], différentes méthodes de détection de sillage dans des images radars ont été testées et comparées. Il est indiqué que les méthodes classiques utilisant la transformée de Radon sont les plus efficaces, ce qui se confirme par une abondante littérature sur le sujet [9, 10, 42, 5, 66, 53].

Toutefois, d'autres méthodes ont été utilisées dans les images radars comme les ondelettes [38] ou une combinaison d'ondelettes et de transformée de radon [37] ou encore l'utilisation d'opérateurs morphologiques [24].

Nous avons réalisé des tests sur nos images en utilisant la transformée de radon, mais la méthode est coûteuse en temps de calcul. Or, le temps de traitement est un paramètre

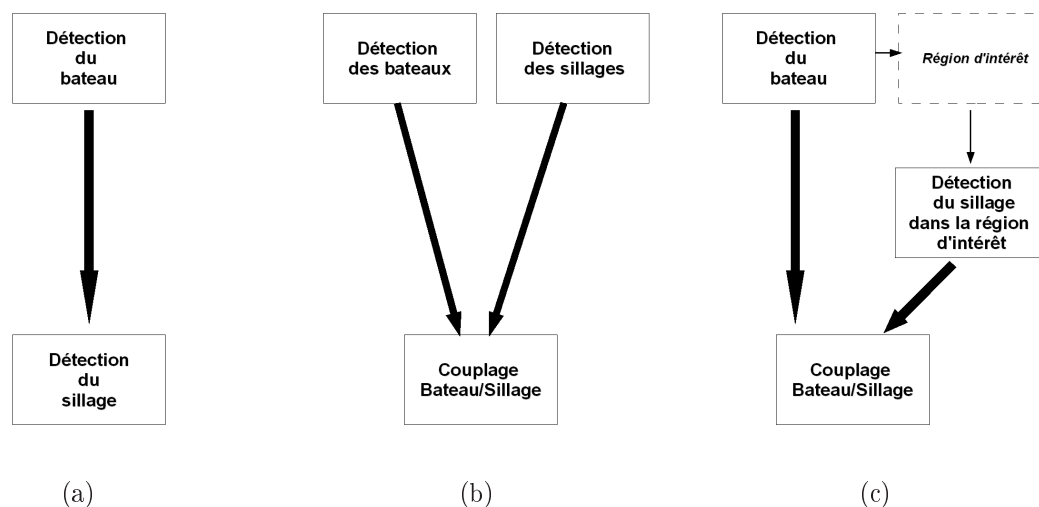


FIG. 4.2 – Les différentes approches possibles pour la détection des sillages.

très important de notre problématique.

Nous avons également réalisé quelques tests en utilisant des ondelettes. Là encore, les résultats ont été peu satisfaisants, principalement pour des sillages peu contrastés avec le fond. Nous avons donc orienté notre recherche vers d'autres méthodes en gardant toujours en tête notre contrainte de temps de traitement et la taille des images traitées.

Nous avons vu précédemment que le sillage peut être souvent considéré comme une droite ou une combinaison de droites. Nous nous sommes donc intéressés aux méthodes de détection de segment. Nous avons opté pour une approche *a contrario* introduite par Desolneux et al [14] pour la détection d'alignement dans les images. C'est une approche basée sur la théorie de la perception visuelle qui présente l'avantage d'être non paramétrique et de prendre en compte dans le processus de décision des informations contextuelles.

Nous pouvons également considérer le sillage comme une zone homogène sur un fond et nous orienter vers la même approche bayésienne que pour la pré-détection (chapitre 2) et la segmentation des bateaux (chapitre 3).

Nous avons donc retenu deux approches afin de les comparer sur un échantillon d'expérimentation qui sera défini dans la partie 4.2.

La première approche consiste à partir des bateaux qui sont détectés lors des étapes précédentes (pré-détection, segmentation et classification), de rechercher le sillage dans le voisinage proche de la cible, puis de segmenter le sillage en utilisant des contours actifs (figure 4.2(a)). Nous nommerons cette méthode **approche contours actifs**. Elle sera présentée dans la partie 4.3.

La seconde approche consiste à considérer la détection du bateau comme le centre d'une

région d'intérêt, de rechercher dans cette région les sillages potentiels par une approche *a contrario*, puis de coupler le bateau avec les sillages candidats (figure 4.2(c)). Nous nommerons cette méthode **approche a contrario**, elle sera étudiée dans la partie 4.4.

Dans la partie 4.5, nous donnerons les résultats expérimentaux de chaque approche. Puis dans la partie 4.6 nous comparerons les résultats des deux approches pour enfin conclure sur la détection de sillage dans la partie 4.7.

4.2 Echantillon d'expérimentation

La première étape de notre travail a consisté à créer un échantillon « d'observation » des sillages et d'expérimentation. Il a été réalisé à partir de la base d'images 2 (chapitre 1.6) et contient différents types de sillages (figure 4.1).

L'échantillon est constitué de 106 extraits d'images contenant à chaque fois un bateau avec un sillage (96) ou sans sillage (10).

Pour chaque image nous avons déterminé par inspection visuelle :

- la présence ou non d'un sillage,
- la position du sillage,
- la direction du sillage.

4.3 L'approche contours actifs

Cette approche se décompose en plusieurs parties. A partir de la position du bateau, nous extrayons une sous-image centrée sur le bateau dans laquelle nous masquons le bateau à l'aide du contour obtenu lors de la segmentation (partie 3.2) afin ne pas perturber la détection du sillage. Puis nous cherchons à détecter le sillage.

La détection du sillage comporte deux parties. Dans un premier temps nous allons déterminer si un sillage est présent ou non. Puis, s'il existe un sillage, nous déterminerons sa direction. La recherche de la direction du sillage utilise les mêmes concepts que la partie pré-détection (chapitre 2).

Ensuite, nous segmentons le sillage à l'aide d'une méthode basée sur l'utilisation des contours actifs pour obtenir le contour précis du sillage. Le contour actif sera initialisé à partir de la direction du sillage obtenue précédemment.

L'ensemble de la démarche est illustrée dans la figure 4.3.

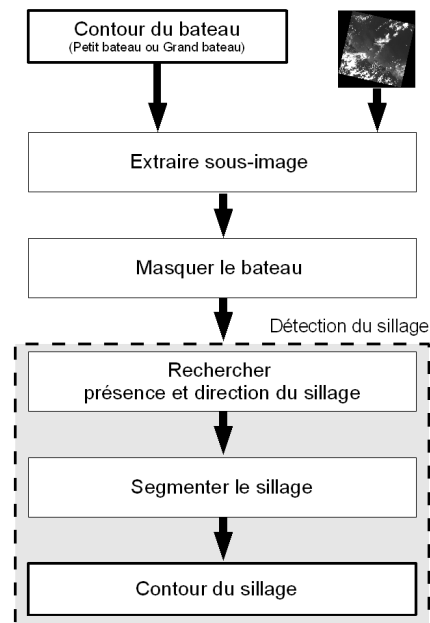


FIG. 4.3 – Démarche de l'approche contours actifs.

4.3.1 Recherche de la direction du sillage

Pour la recherche de la présence et de la direction d'un sillage, nous avons utilisé la même approche que pour la pré-détection des bateaux ; c'est-à-dire une méthode qui s'appuie sur le théorème bayésien de la décision.

Nous réalisons un pré-traitement qui consiste à appliquer un masque sur le bateau pour ne pas prendre en compte les pixels du bateau. Ce masque est réalisé à partir du contour obtenu lors de l'étape de segmentation (partie 3.2).

Nous définissons deux hypothèses :

- $$\left\{ \begin{array}{l} H_0 : \text{la fenêtre ne couvre que du fond sans sillage.} \\ \quad \text{La région correspondante est notée } B. \\ \quad \text{Cette hypothèse est illustrée par la figure 4.4(a).} \\ \\ H_1 : \text{Un sillage de longueur } L_w, \text{ de largeur } l_w \text{ et} \\ \quad \text{de direction } \theta \text{ par rapport à la cible est présent.} \\ \quad \text{La région correspondante est notée } w \text{ et} \\ \quad \text{la région notée } \bar{w} \text{ correspond à l'extérieur du sillage.} \\ \quad \text{Cette hypothèse est illustrée par la figure 4.4(b).} \end{array} \right.$$

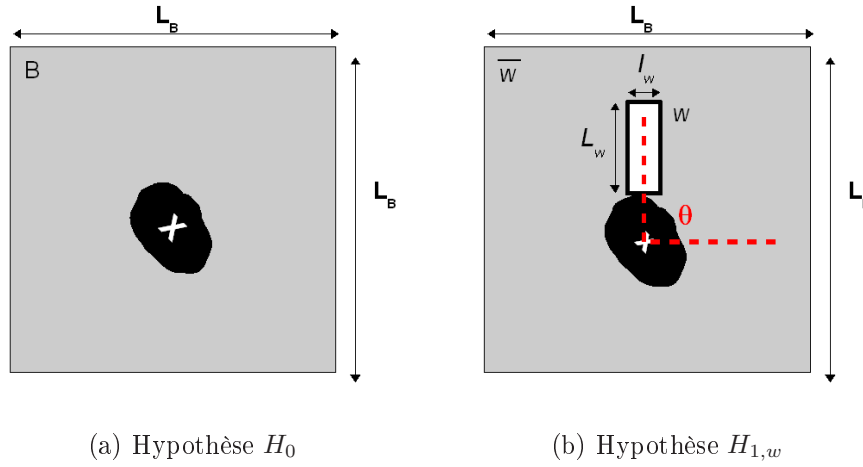


FIG. 4.4 – Scène (s) sous les deux hypothèses H_0 et $H_{1,w}$. La fenêtre d'analyse est un carré de taille $L_B \times L_B$ pixels. Le sillage est considéré comme un rectangle de taille $L_w \times l_w$ et de direction θ par rapport à la cible. On peut remarquer que la cible est masquée pour ne pas être prise en compte dans les calculs.

Le modèle statistique est le même que dans la partie 2.2.3 : le fond et le sillage sont supposés suivre une distribution gaussienne blanche de moyennes différentes, mais de variance identique. Comme le modèle statistique de la scène est identique, le calcul du rapport de vraisemblance est identique.

Pour chaque forme w , nous calculons $\log\Lambda = L(w) + L(\bar{w}) - L(B)$ comme dans l'équation 2.16 que nous rappelons ici en prenant en compte les nouvelles notations :

$$\log\Lambda(\mathbf{w}) = N_w \hat{m}_w^2 + N_{\bar{w}} \hat{m}_{\bar{w}}^2 - N_B \hat{m}_B^2 \quad (4.1)$$

avec les notations suivantes :

- $N_w, N_{\bar{w}}, N_B$: nombre de pixels des régions w, \bar{w}, B .
- $\hat{m}_w, \hat{m}_{\bar{w}}, \hat{m}_B$: estimation de la moyenne des régions w, \bar{w} et B .

Nous considérons deux cas :

- La cible est allongée : on détermine son orientation puis on recherche le sillage aux extrémités (figure 4.5(a)).
- La cible n'est pas allongée (petite cible) : on recherche le sillage partout autour de la cible (figure 4.5(b)).

Pour déterminer la présence d'un sillage, nous allons tester plusieurs longueurs possibles en faisant varier L_w entre 4 et 40 pixels et nous fixons la largeur l_w à 3 pixels.

Soit $w_{L_w, \theta}$ le contour de longueur L_w et de direction θ . La figure 4.5 illustre les différentes valeurs de θ testées.

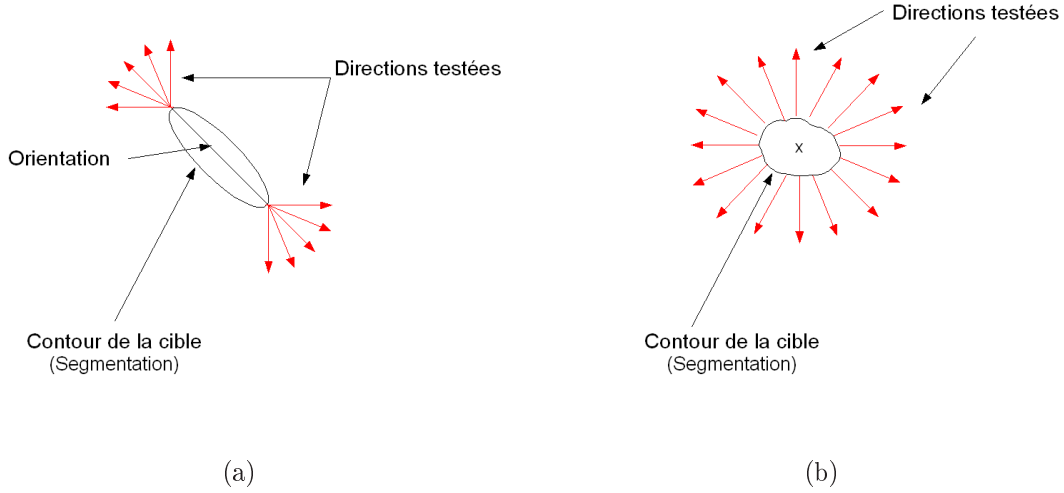


FIG. 4.5 – Recherche de la direction du sillage dans le cas d'une grande cible (a) et dans le cas d'une petite cible (b).

Pour chaque longueur L_w , nous gardons le meilleur candidat : c'est celui qui donne la valeur maximale pour $\log\Lambda(w_{L_w, \theta})$ (équation 4.1). On note θ_{L_w} la direction du meilleur sillage de longueur L_w .

Nous définissons deux critères pour déterminer l'absence de sillage :

- $\log\Lambda(w_{L_w, \theta_{L_w}}) < s1$
- le contraste entre le sillage ($w_{L_w, \theta_{L_w}}$) et le fond ($\bar{w}_{L_w, \theta_{L_w}}$) est inférieur à un seuil $s2$

Les seuils $s1$ et $s2$ sont fixés empiriquement.

Nous pouvons seuiller la valeur $\log\Lambda(w_{L_w, \theta_{L_w}})$ car, à la différence de Réfrégier et al [25], nous avons ajouté dans notre approche le terme qui correspond à l'hypothèse H_0 ($L(B)$). Ce terme normalise l'énergie par rapport à l'intensité de la fenêtre. Il n'a pas d'intérêt pour la segmentation du sillage. En revanche, dans la décision de la présence d'un sillage, il permet d'utiliser la valeur du GLRT pour prendre cette décision.

A l'issue de ce traitement, nous obtenons un ensemble de contours candidats ($E_{contours}$) et un ensemble de directions possibles pour le sillage (E_θ).

Nous conservons l'angle qui a le plus d'occurrences dans l'ensemble E_θ (nous nommons cet angle $\theta_{sillage}$). Puis, nous sélectionnons dans $E_{contours}$ le contour de direction $\theta_{sillage}$ ayant la plus grand longueur (L_w). C'est ce contour qui est utilisé pour initialiser la segmentation du sillage à l'aide des contours actifs.

4.3.2 Segmentation du sillage

Nous utilisons la même approche que pour la segmentation des bateaux décrite dans la partie 3.2, la forme w représentant maintenant le sillage et non le bateau.

Energie utilisée

L'énergie utilisée est la même que celle définie dans la partie 3.2.3, nous rappelons ici la formule :

$$\mathbf{E}(\mathbf{w}) = -(\mathbf{N}_w \hat{m}_w^2 + \mathbf{N}_{\bar{w}} \hat{m}_{\bar{w}}^2 - \mathbf{N}_B \hat{m}_B^2) \quad (4.2)$$

avec les notations suivantes :

- $N_w, N_{\bar{w}}, N_B$: nombre de pixels des régions w, \bar{w}, B .
- $\hat{m}_w, \hat{m}_{\bar{w}}, \hat{m}_B$: estimation de la moyenne des régions w, \bar{w} et B .

Algorithme d'optimisation

L'algorithme utilisé ici est différent de celui utilisé pour la segmentation des bateaux. Nous allons privilégier les déplacements de points du contour dans la direction du contour et dans son voisinage. En effet, il n'est pas nécessaire de tester des configurations où le point est déplacé d'un grand angle par rapport à la direction du contour, cela est peu réaliste.

Soit θ l'orientation du sillage, nous déplaçons chaque point comme cela est illustré dans la figure 4.6 :

- dans la direction θ
- dans la direction $\theta + \frac{\pi}{32}$
- dans la direction $\theta - \frac{\pi}{32}$

Nous pouvons ainsi suivre les éventuelles changements de trajectoire du sillage.

Le nombre maximum d'itérations est fixé à 100.

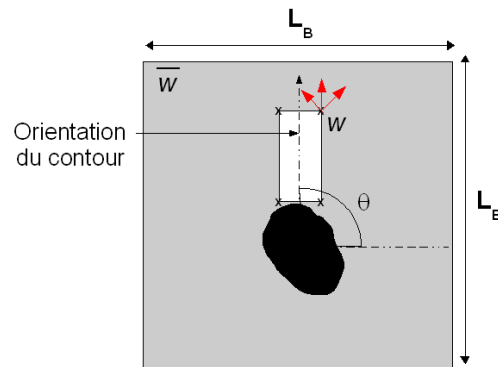


FIG. 4.6 – Déplacements d'un noeud du contour : ils sont représentés par les flèches en rouge.

4.3.3 Exemples

Nous présentons dans la figure 4.7 quelques exemples de résultats de la détection de sillage par l'**approche contours actifs**.

Cette approche détecte des sillages ayant une trajectoire non-rectiligne comme nous pouvons le constater dans la figure 4.7(a). Cela fournit une information supplémentaire qui est de savoir si un bateau est en train d'exécuter une manoeuvre.

Les sillages clairs (figure 4.7(b)) ou foncés (figure 4.7(d)) sont aussi bien détectés.

Par contre, la détection des sillages composés de plusieurs éléments (sillage principal et vagues d'étrave) comme dans la figure 4.7(c) est plus problématique. En effet, nous ne détectons qu'un seul sillage dans notre approche. Nous pouvons envisager de masquer le sillage déjà détecté et de recommencer le processus pour détecter les autres éléments du sillage.

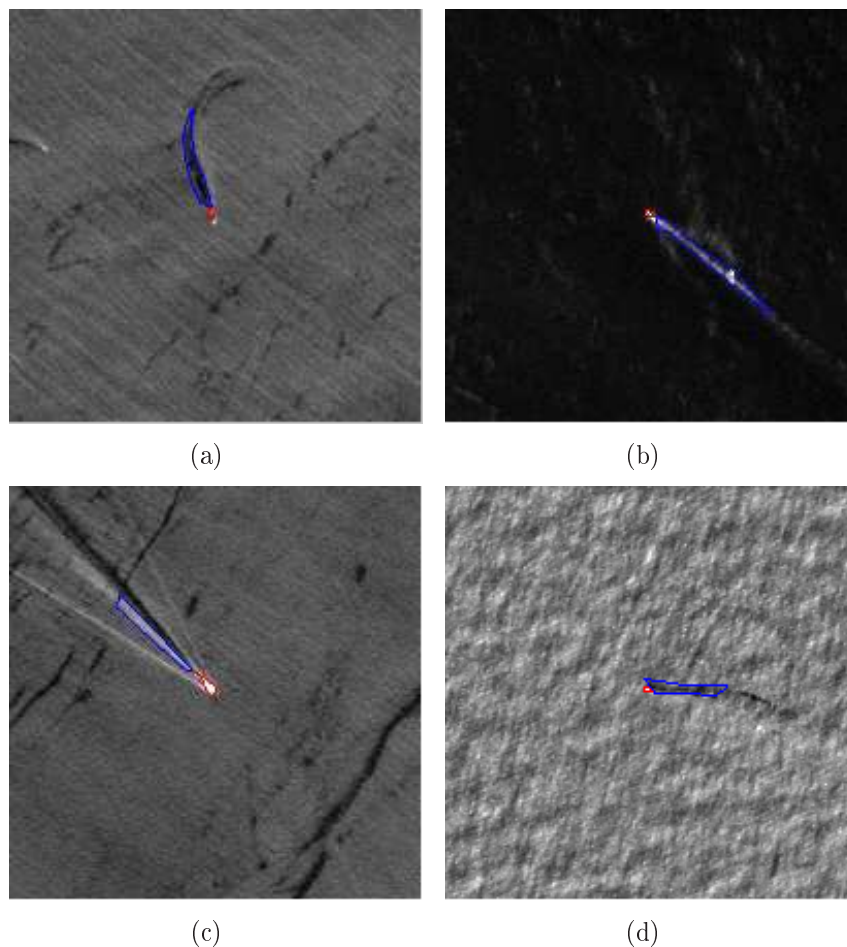


FIG. 4.7 – Exemples de résultats de la détection de sillages par l'approche contours actifs.

4.4 L'approche *a contrario*

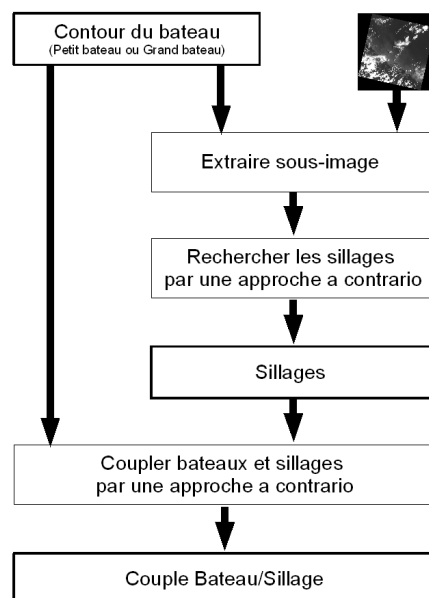
Cette approche se décompose en deux parties. Dans un premier temps, nous détectons les segments dans une sous-image centrée sur le bateau. Ensuite, nous couplons ces segments avec le bateau afin de déterminer les segments qui appartiennent au sillage. Nous avons utilisé une approche *a contrario* pour les deux étapes.

L'ensemble de la démarche est illustrée dans la figure 4.8.

4.4.1 Principes de l'approche *a contrario*

L'approche *a contrario* introduite par Desolneux et al en 1999 est une approche statistique qui s'inspire de recherches en psychologie de la vision [17, 16].

Cette approche détecte des structures géométriques telles que des alignements [14],

FIG. 4.8 – Démarche de l'approche *a contrario*.

des contours ou des lignes contrastées [15].

Nous pouvons résumer son principe par « chercher ce qui ne devrait pas être là ». Ainsi, une structure géométrique sera dite significative si son observation dans une image aléatoire (au sens intuitif du terme) est très peu probable.

Supposons que l'on veuille détecter une structure géométrique dans une image, le modèle a priori consiste à supposer qu'il n'y a pas de structure géométrique (il n'y a que du bruit). La détection *a contrario* consiste alors à déterminer le seuil à partir duquel on considère que ce n'est pas le modèle a priori qui est observé, mais un événement. Un événement est détecté comme un écart par rapport au modèle a priori. L'avantage de cette approche est qu'elle est robuste au bruit.

Soit l'hypothèse H_0 (modèle de fond) : variables aléatoires indépendantes et uniformément distribuées. Soit E : l'observation d'une structure géométrique. Soit $P(E|H_0)$: la probabilité d'observer E sous l'hypothèse H_0 .

Le critère de décision est donc :

Si $\begin{cases} P(E|H_0) \text{ est « fort » alors on est en présence de l'hypothèse } H_0. \\ P(E|H_0) \text{ est « faible » alors on rejette } H_0 \text{ et donc on accepte } H_1 \text{ (non défini).} \end{cases}$

Cela permet de faire l'économie d'un modèle de données pour H_1 qui peut être complexe et d'avoir uniquement le modèle de H_0 qui est simple le plus souvent : des données indépendantes et uniformément distribuées.

Nous fixons un seuil ϵ et au lieu de seuiller $P(E|H_0)$, nous seuillons le nombre de fausses alarmes $NFA(E)$ (qui correspond au nombre d'occurrences de l'observation E).

Le critère de décision devient alors : si $NFA(E) \leq \epsilon$ alors l'observation E est dite ϵ -significative. Dans ce cas, nous décidons que nous sommes en présence de la structure géométrique recherchée.

4.4.2 Détection des sillages par une approche *a contrario*

L'approche *a contrario* a été utilisée pour la détection d'alignements [14] et de lignes [16] dans des images. Nous allons rapidement en décrire son principe.

Direction des points

En chaque point d'une image de taille $N \times N$, nous calculons une direction (définie comme étant orthogonale au gradient calculé sur un voisinage de taille 2×2).

Soit $I(x, y)$ les pixels d'une image I de taille $N \times N$

Soit $dir(x, y)$ la direction du point (l'orthogonal au gradient sur un voisinage de taille 2×2) est calculée de la manière suivante :

$$dir(x, y) = \frac{1}{\|\vec{D}\|} \vec{D} \quad (4.3)$$

où

$$\vec{D} = \frac{1}{2} \begin{pmatrix} -[I(x, y+1) + I(x+1, y+1)] + [I(x, y) + I(x+1, y)] \\ [I(x+1, y) + I(x+1, y+1)] - [I(x, y) + I(x, y+1)] \end{pmatrix} \quad (4.4)$$

On dit qu'un point $P = (x, y)$ est aligné avec un segment S à la précision ρ près (figure 4.9) si

$$|Angle(dir(x, y), S)| \leq \rho\pi \quad (4.5)$$

Dans la littérature, le choix traditionnel est de fixer la précision $\rho = \frac{1}{16}$ [14].

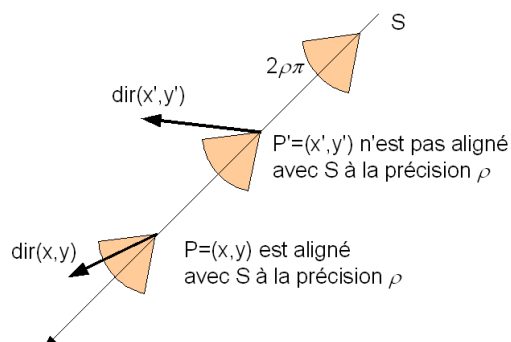


FIG. 4.9 – Notion d'alignement d'un point avec un segment.

Hypothèse H_0

Soit l'hypothèse H_0 : les directions des points de l'image sont indépendantes et uniformément distribuées sur $[0, 2\pi]$.

Principe de groupement

Pour chaque segment de l'image, on calcule le nombre de points alignés avec ce segment.

Soit S un segment de longueur ℓ et x_1, x_2, \dots, x_ℓ les ℓ points de S . Nous comptons le nombre k de points parmi les ℓ ayant la propriété d'avoir leur direction alignée avec celle du segment S à la précision ρ près. Si le nombre k est suffisamment grand pour être très peu probable sous l'hypothèse H_0 , on retient le segment.

Soit X_i la variable aléatoire définie par :

$$\begin{cases} X_i = 1 & \text{si le point } x_i \text{ est aligné avec la direction de } S \text{ à la précision } \rho \\ X_i = 0 & \text{sinon} \end{cases}$$

Nous définissons les probabilités suivantes :

$$P_{H_0}[X_i = 1] = p \text{ et } P_{H_0}[X_i = 0] = 1 - p \quad (4.6)$$

Soit S_ℓ la variable aléatoire représentant le nombre de points x_i ayant la « bonne » direction :

$$S_\ell = X_1 + X_2 + \dots + X_\ell \quad (4.7)$$

Les points étant supposés indépendants et leur direction uniformément distribuée, la probabilité de l'événement « k points alignés parmi ℓ , à la précision ρ près » est égale à :

$$P_{H_0}[S_\ell = k] = C_\ell^k \cdot p^k \cdot (1 - p)^{\ell - k} \quad (4.8)$$

Soit $P_{H_0}[S_\ell \geq k]$ la probabilité de l'événement « au moins k points alignés parmi ℓ , à la précision ρ près ». La probabilité de cet événement est donnée par la formule suivante :

$$P_{H_0}[S_\ell \geq k] = \sum_{i=k}^{\ell} C_\ell^i \cdot p^i \cdot (1 - p)^{\ell - i} \quad (4.9)$$

Dans [14], Desolneux et al donne la définition suivante : un segment S_ℓ de longueur ℓ est dit ϵ -significatif dans une image $N \times N$ s'il contient au moins $k(\ell)$ points alignés avec la direction du segment, où

$$k(\ell) = \min\{k \in \mathbb{N}, P_{H_0}[S_\ell \geq k] \leq \frac{\epsilon}{N^4}\} \quad (4.10)$$

Le nombre total de segments possibles dans une image de taille $N \times N$ est $N^2(N-1)^2 \approx N^4$.

Soit e_i définie par :

$$\begin{cases} e_i = 1 & \text{si le } i\text{-ème segment de l'image est } \epsilon\text{-significatif} \\ e_i = 0 & \text{sinon} \end{cases}$$

Soit R le nombre total de segments ϵ -significatif dans l'image.

$$\mathbb{E}_{H_0}[R] = \sum_{i=1}^{N^4} \mathbb{E}[e_i] = \sum_{i=1}^{N^4} P_{H_0}[S_{\ell_i} \geq k(\ell_i)] \leq \epsilon \quad (4.11)$$

L'espérance sous l'hypothèse H_0 du nombre de segment ϵ -significatif est inférieure ou égale à ϵ .

Le nombre de fausses alarmes

Dans [14], Desolneux et al définit le nombre de fausses alarmes. Soit un segment S de longueur ℓ_0 et ayant au moins k_0 points alignés avec la direction de S . Nous lui associons

le nombre de fausses alarmes défini par :

$$\begin{aligned} NFA(k_0, \ell_0) &= N^4 \cdot P[S_{\ell_0} \geq k_0] \\ &= N^4 \cdot \sum_{i=k_0}^{\ell_0} C_{\ell_0}^i \cdot p^i \cdot (1-p)^{\ell_0-i} \end{aligned} \quad (4.12)$$

Critère de décision

si $NFA(k_0, \ell_0) < \epsilon$ alors le segment S de longueur ℓ_0 et ayant au moins k_0 points alignés est ϵ -significatif.

Choix de la taille de la sous-image

Le but étant de rechercher le sillage d'un bateau il n'est pas nécessaire de rechercher celui-ci trop loin du bateau. Une taille trop élevée serait inutile et coûteuse en temps de calcul. Nous avons fixé la taille N à 201 pixels.

Choix du seuil ϵ

Dans la littérature, le choix traditionnel est de fixer $\epsilon = 1$ [14]. Nous avons réalisé des tests sur notre échantillon d'expérimentation (partie 4.2) avec cette valeur. La probabilité de détection obtenue était de 0.263, ce qui est assez médiocre : les segments ayant un contraste faible avec le fond n'ont pas été détectés.

Nous allons donc déterminer une valeur de ϵ mieux adaptée. Nous utilisons pour cela une courbe ROC (*Receiver Operating Characteristic*), comme dans la partie 2.2.10.

Pour un seuil ϵ donné, nous calculons la probabilité de détection des sillages (PD) et la probabilité de fausse alarme (PFA). Le tracé du couple (PFA, PD) , pour $1 < \epsilon < +\infty$ produit la courbe ROC. La figure 4.12 représente la courbe ROC obtenue.

La probabilité de détection est calculée de la manière suivante :

$$PD = \frac{N_D}{N_P}$$

où N_D est le nombre de sillages détectés et N_P est le nombre de sillages présents. La figure 4.10 représente la courbe de probabilité de détection en fonction de la valeur de ϵ .

La probabilité de fausse alarme est calculée de la manière suivante :

$$PFA = \frac{N_{FS}}{N_{img} - N_{VS}}$$

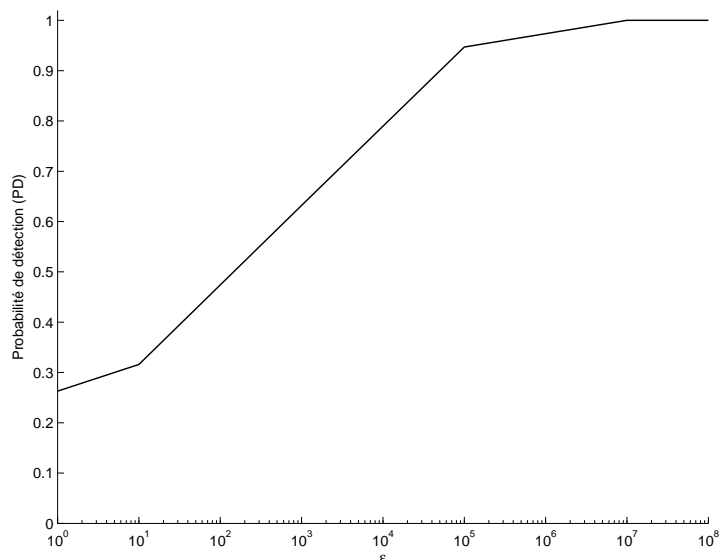


FIG. 4.10 – Probabilité de détection en fonction de ϵ .

où N_{FS} est le nombre de faux sillages (détectés comme étant des sillages, mais qui n'en sont pas), N_{img} est le nombre de sillages possibles pour l'échantillon et N_{VS} est le nombre de détections correspondant à de vrais sillages. La figure 4.11 représente la courbe de probabilité de fausse alarme en fonction de la valeur de ϵ .

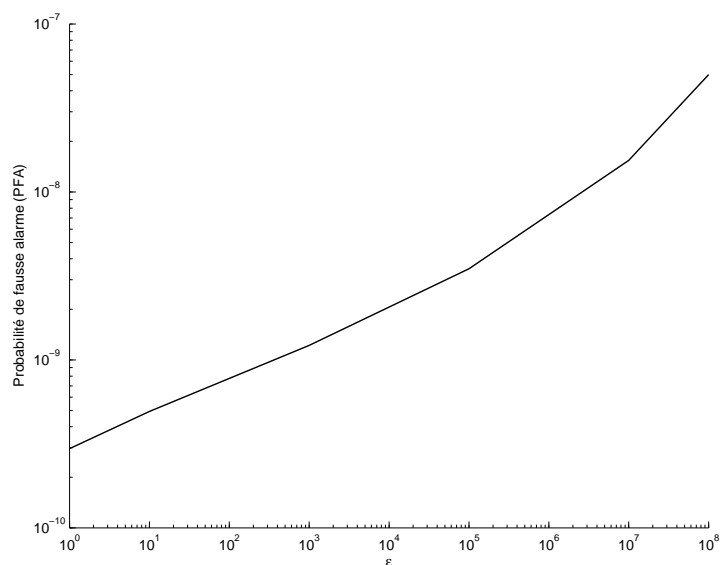


FIG. 4.11 – Probabilité de fausse alarme en fonction de ϵ .

La courbe ROC est utilisée pour fixer une valeur de ϵ qui permet de maximiser la probabilité de détection en gardant un taux de fausse alarme le plus bas possible. Nous utiliserons donc $\epsilon = 10^7$.

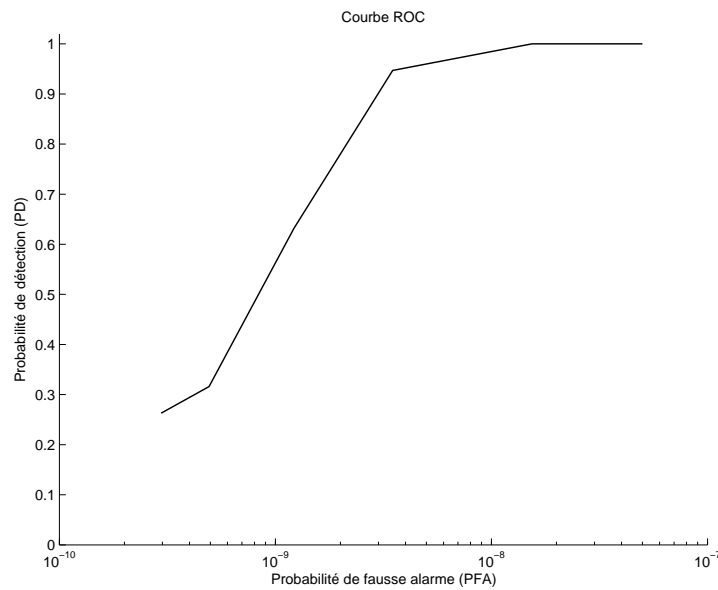


FIG. 4.12 - Courbe ROC.

Conclusion

En utilisant une valeur élevée pour ϵ , les sillages sont bien détectés comme nous pouvons le constater dans les figures 4.13 et 4.15.

Lorsque la trajectoire n'est pas rectiligne, un sillage peut être composé de plusieurs segments comme dans les figures 4.13(b) et 4.13(d).

Lorsque le sillage est assez large, plusieurs segments peuvent être associés à un même sillage comme on peut voir dans la figure 4.14(b).

Dans le cas d'un sillage en « V » les différentes parties du sillage (sillage principal et les vagues d'étrave) sont détectées comme nous pouvons le constater dans la figure 4.15. La phase de couplage bateau/sillage se chargera d'associer ces différents segments au sillage d'un même bateau.

Dans la figure 4.16, nous constatons qu'une mer agitée engendre de nombreuses fausses alarmes et le sillage peut être « noyé » au milieu des segments générés par l'état de la mer. Le couplage présenté dans la partie 4.4.3 va éliminer ces fausses alarmes.

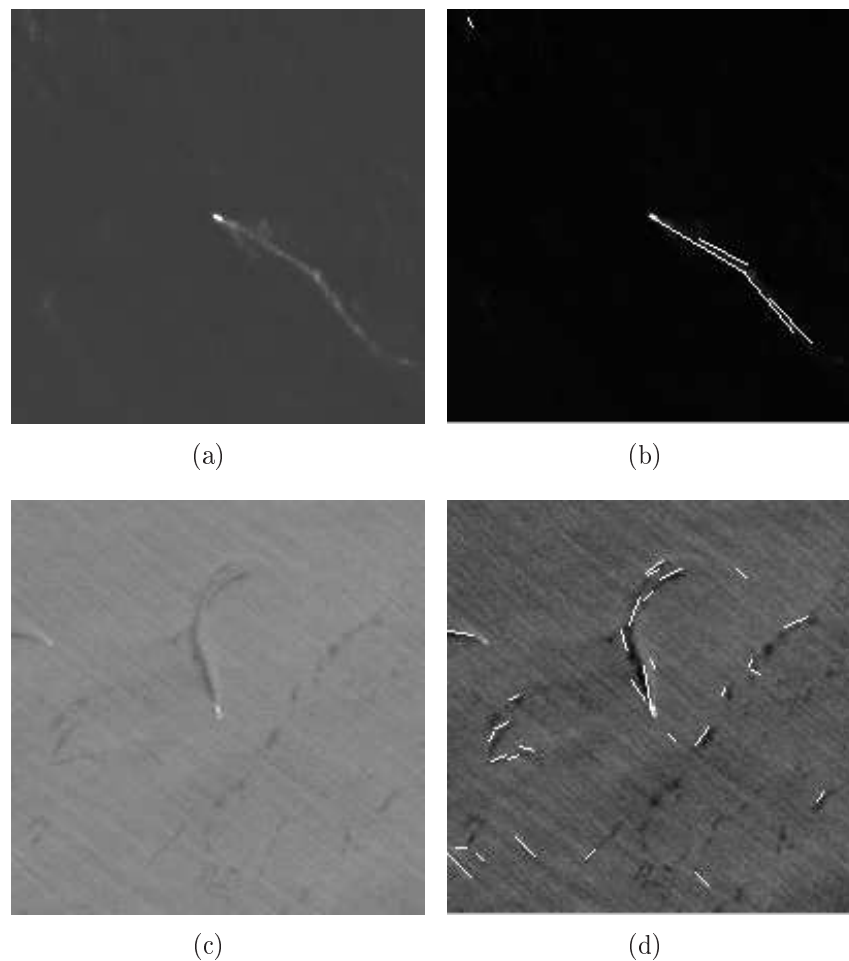


FIG. 4.13 – Exemple de résultats de la détection de segments par une approche *a contrario* dans le cas où le sillage n'est pas rectiligne. Nous avons plusieurs segments qui vont suivre la trajectoire du sillage. (a)(c) sont les images traitées et (b)(d) sont les résultats de la détection des segments. Les segments détectés sont en blanc dans les images.

4.4.3 Le couplage bateau/sillage

Principe

Nous voulons mettre en relation un bateau et son sillage grâce à une approche *a contrario*. Celle-ci permet, à partir d'un bateau et d'un ensemble de détections de sillages, de créer des couples bateau/sillage.

Nous disposons :

- des bateaux issus des étapes de pré-détection (chapitre 2), segmentation et classification (chapitre 3),
- des sillages issus de la détection de segments décrite dans la partie 4.4.2.

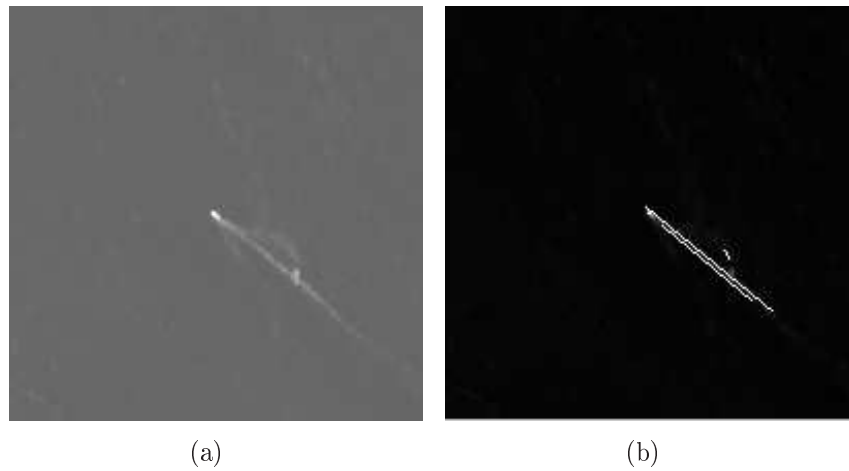


FIG. 4.14 – Exemple de résultats de la détection de segments par une approche *a contrario* dans le cas où le sillage est large. Nous avons deux segments détectés sur les bords du sillage. (a) est l'image traitée et (b) est le résultat de la détection des segments. Les segments détectés sont en blanc dans les images.

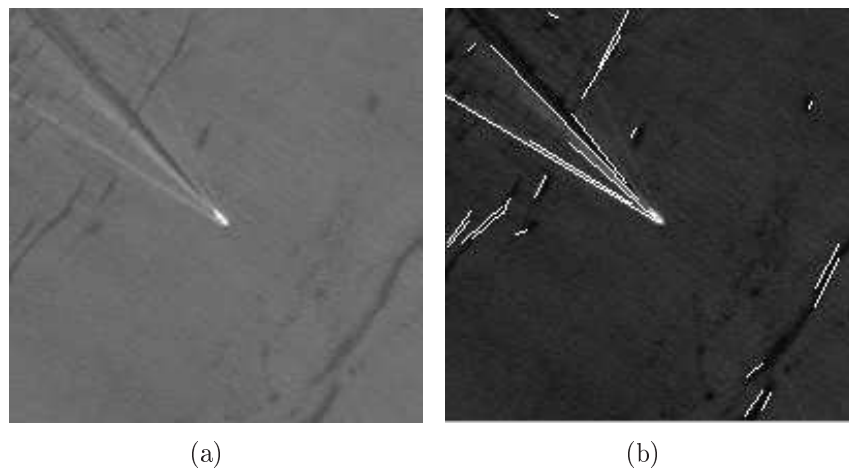


FIG. 4.15 – Exemples de résultats de la détection de segments par une approche *a contrario* dans le cas où le sillage est composé de plusieurs éléments (sillage principal et vagues d'étrave). (a) est l'image traitée et (b) est le résultat de la détection des segments. Les segments détectés sont en blanc dans les images.

A priori, si un vrai couple est présent (H_1), le bateau se trouve :

- près d'une extrémité du sillage,
- dans son alignement,
- à une distance variable du sillage.

Cela constitue un modèle de données non trivial.

Dans le cas de fausses alarmes (H_0), les positions du bateau et du sillage sont indépendantes.

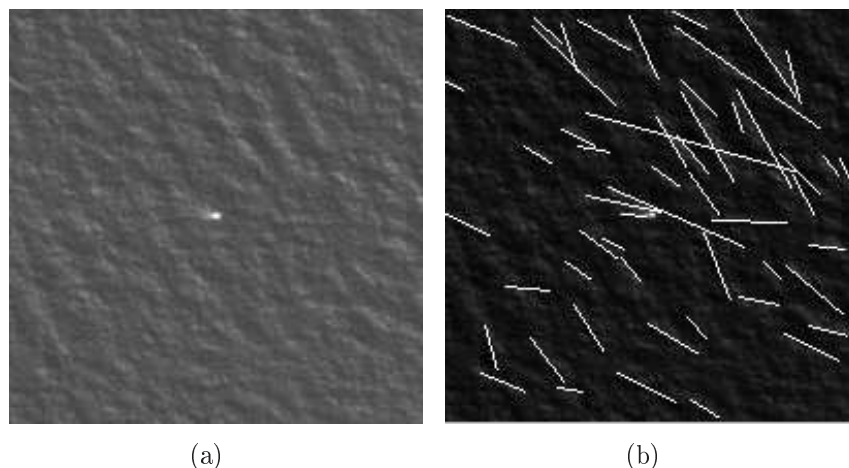


FIG. 4.16 – Exemple de résultats de la détection de segments par une approche a contrario dans le cas d'une mer agitée. (a) est l'image traitée et (b) est le résultat de la détection des segments. Les segments détectés sont en blanc dans les images.

Modélisation

Soit un bateau (B) représenté par un point $P_B = (x, y)$ et un sillage (S) représenté par un point $P_S = (x', y')$ et une orientation (dx, dy) .

Le bateau (B) se situe à la distance r du sillage (S) et le vecteur $\vec{P_B P_S}$ fait un angle θ avec l'orientation du sillage.

Un couple bateau/sillage est illustré dans la figure 4.17.

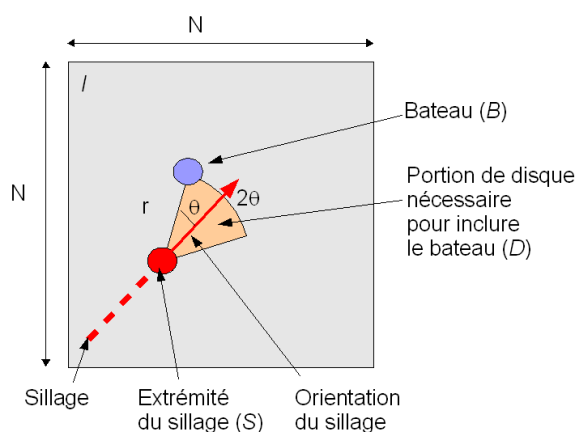


FIG. 4.17 – Modélisation d'un couple bateau/sillage. Le sillage est en rouge et le bateau est en bleu.

Soit $P_{H_0}(B, S)$ la probabilité que pour un sillage (S, θ) tiré au hasard selon une distri-

bution uniforme sur l'image (I), le point (B) tombe dans la portion de disque (D) centrée sur S et d'angle 2θ . Cette quantité est calculée de la manière suivante :

$$P_{H_0}(B, S) = \frac{\text{Aire}(D)}{\text{Aire}(I)} = \frac{\theta r^2}{N^2} \quad (4.13)$$

Au lieu de seuiller $P_{H_0}(B, S)$, nous voulons seuiller le nombre de fausses alarmes possibles $NFA(B, S)$ qui est donné par la formule suivante :

$$NFA(B, S) = N_{\text{Couples}} \cdot P_{H_0}(B, S) \quad (4.14)$$

où N_{Couples} est le nombre de couples bateau/sillage possibles.

Le critère de décision

Si $NFA(B, S) \leq \epsilon$ alors l'observation est dite ϵ -significative et nous décidons qu'il y a un couplage entre le bateau (B) et le sillage (S).

Algorithme

Nous obtenons un algorithme simple et rapide. Pour chaque couple bateau/sillage (B, S) de l'image :

1. Calculer $P_{H_0}(B, S)$ (équation 4.13)
2. Calculer $NFA(B, S)$ (équation 4.14)
3. Déterminer si l'observation est ϵ -significative
4. Prendre la décision : c'est un couple ou pas

Nous cherchons maintenant un seuil ϵ correct pour $NFA(B, S)$.

Choix du seuil ϵ

Dans la partie 4.4.2, nous avons fixé la taille de la sous-image à $N = 201$ pixels.

Nous allons maintenant étudier les performances de la méthode en fonction de ϵ et du nombre de sillages potentiels présents dans la sous-image. Cela va nous permettre de fixer une valeur adaptée pour ϵ . Nous testons plusieurs valeurs pour $\epsilon = \{1, 0.5, 0.2, 0.1\}$.

Nous générons des configurations synthétiques sur lesquelles nous allons tester l'algorithme. Pour chaque configuration :

- Nous fixons trois « vrais » couples (en bleu sur la figure 4.18), la distance entre le sillage et le bateau varie entre 3 et 20 pixels et l'angle varie entre $\frac{\pi}{18}$ et $\frac{\pi}{32}$.
- Nous générons aléatoirement $Nb_{fauxSillages}$ faux sillages (en rouge sur la figure 4.18). Nous faisons varier $Nb_{fauxSillages}$ entre 1 et 140.

Un exemple de configuration synthétique est présenté dans la figure 4.18 et le résultat de l'algorithme pour les différents ϵ est présenté dans la figure 4.19.

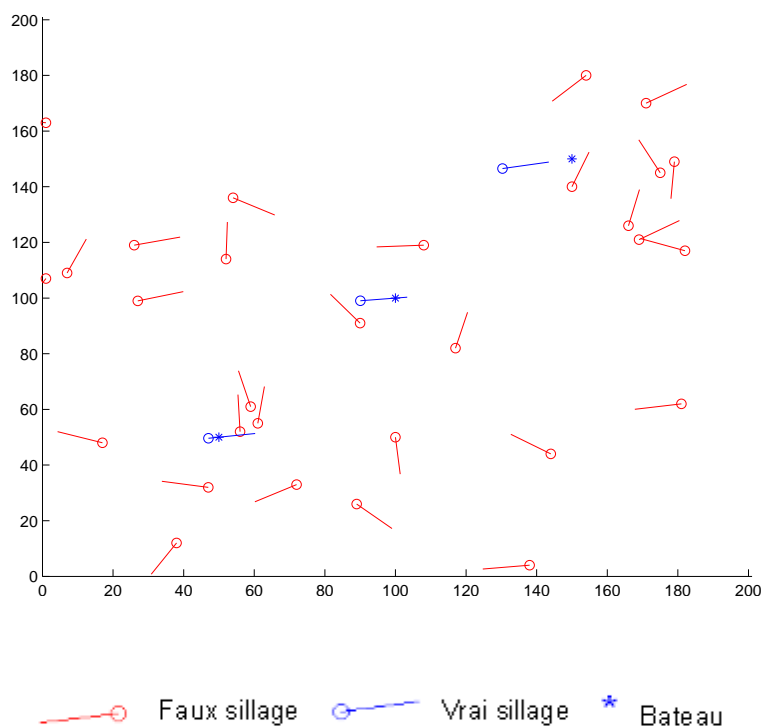


FIG. 4.18 – Exemple d'une configuration synthétique de test. Les bateaux sont représentés par les astérisques bleus, les vrais sillages sont les segments bleus et les faux sillages générés aléatoirement sont les segments en rouge. Dans cet exemple $Nb_{fauxSillages}=30$.

Pour chaque couple $(\epsilon, Nb_{fauxSillages})$:

- Nous générons 100 configurations.
- Nous calculons la probabilité de détecter les vrais couples :

$$PD = \frac{N_D}{N_P}$$

où N_D est le nombre de vrais couples détectés et N_P est le nombre de vrais couples présents.

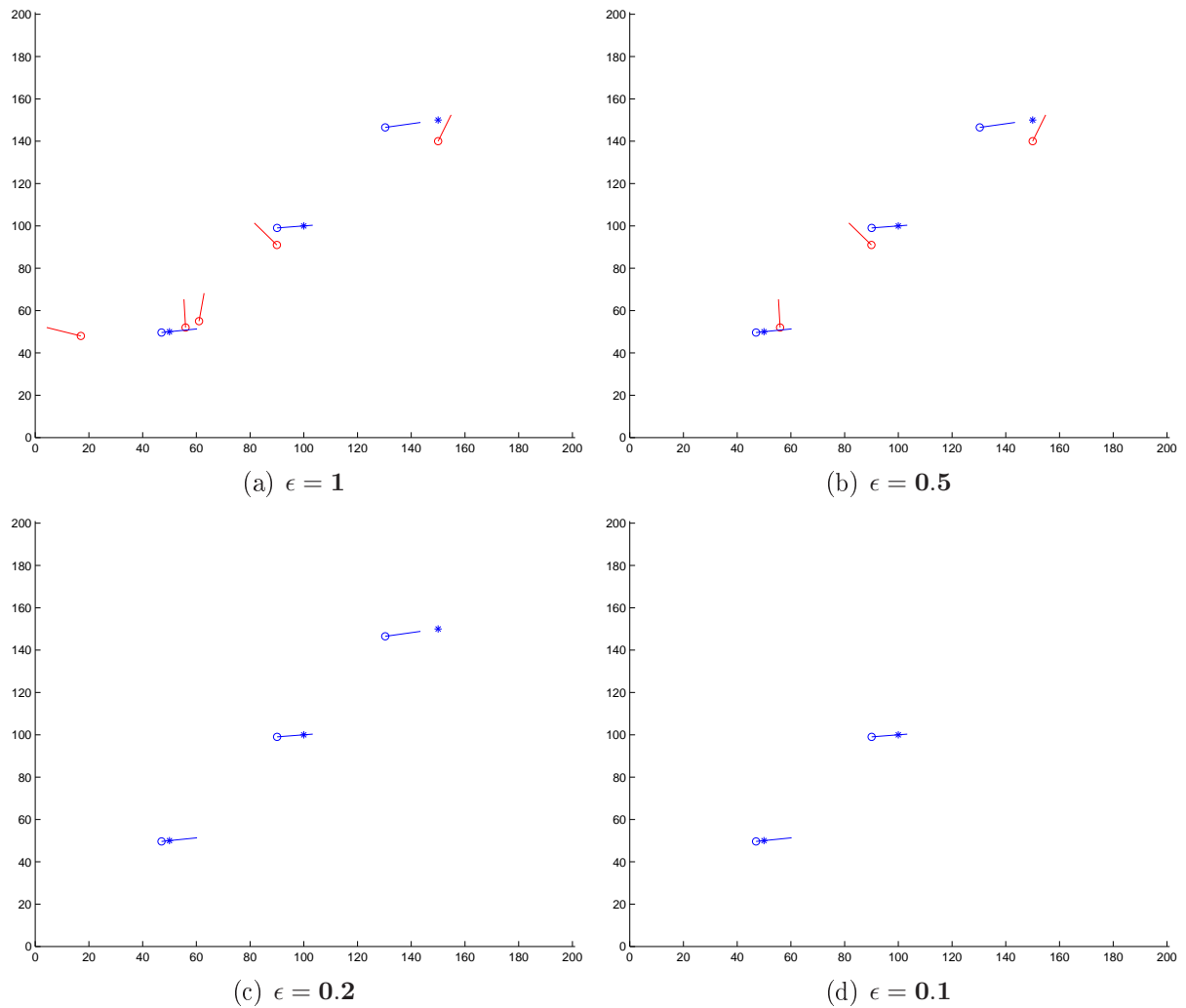


FIG. 4.19 – Résultats de l'algorithme avec $\epsilon=1, 0.5, 0.2$ et 0.1 .

Les vrais couples sont en bleu et les faux couples en rouge.

(a) $\epsilon = 1$: les trois vrais couples sont détectés, mais on a également cinq faux couples.

(b) $\epsilon = 0.5$: les trois vrais couples sont détectés, mais on a également trois faux couples.

(c) $\epsilon = 0.2$: les trois vrais couples sont détectés et il n'y a pas de faux couples détectés.

(d) $\epsilon = 0.1$: seulement deux vrais couples sur trois sont détectés et il n'y a pas de faux couples détectés.

– Nous calculons la probabilité de détecter des faux couples :

$$PFA = \frac{N_{FA}}{N_{couples} - N_D}$$

où N_{FA} est le nombre de faux couples détectés, $N_{couples}$ est le nombre de couples possibles pour l'échantillon et N_D est le nombre de vrais couples détectés.

La figure 4.20 présente la courbe de probabilité de détection en fonction de la valeur ϵ et du nombre de faux sillages.

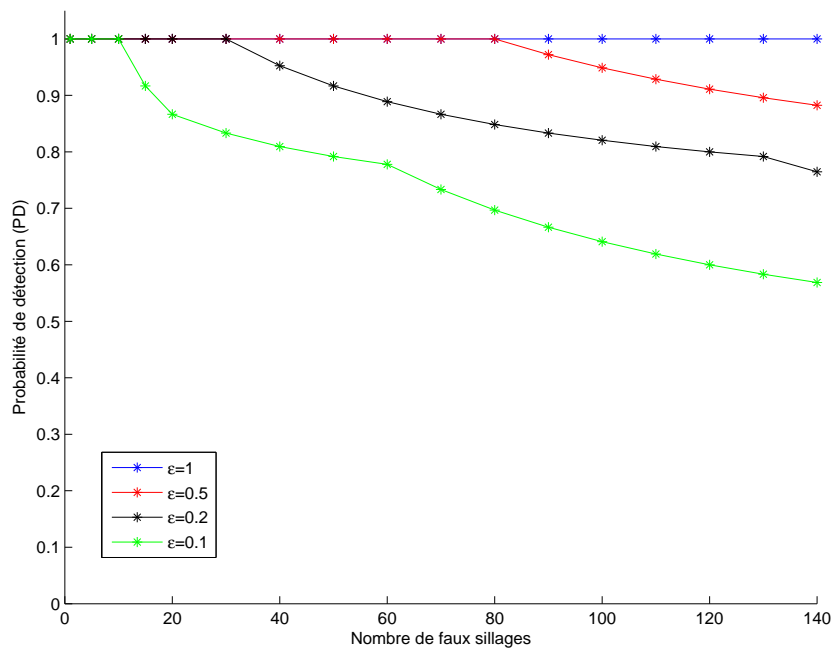


FIG. 4.20 – Courbe de probabilité de détection des vrais couples en fonction du nombre de faux sillages et pour $\epsilon=1, 0,5, 0,2$ et $0,1$.

La figure 4.21 présente la courbe de probabilité de fausse alarme en fonction de la valeur ϵ et du nombre de faux sillages.

Nous constatons qu'avec $\epsilon = 1$, les vrais couples sont toujours détectés. Par contre le nombre de fausses alarmes est important lorsque le nombre de faux sillages est faible.

Plus on va diminuer ϵ , plus le nombre de faux couples détectés va diminuer, mais en diminuant trop ϵ , le risque de ne plus détecter les vrais couples augmente.

Si nous avons peu de sillages candidats, il est préférable de prendre un ϵ petit pour ne pas avoir trop de faux couples détectés.

Par contre, quand nous avons un nombre élevé de sillages, un ϵ trop faible sera trop sélectif et tous les couples ne seront pas détectés .

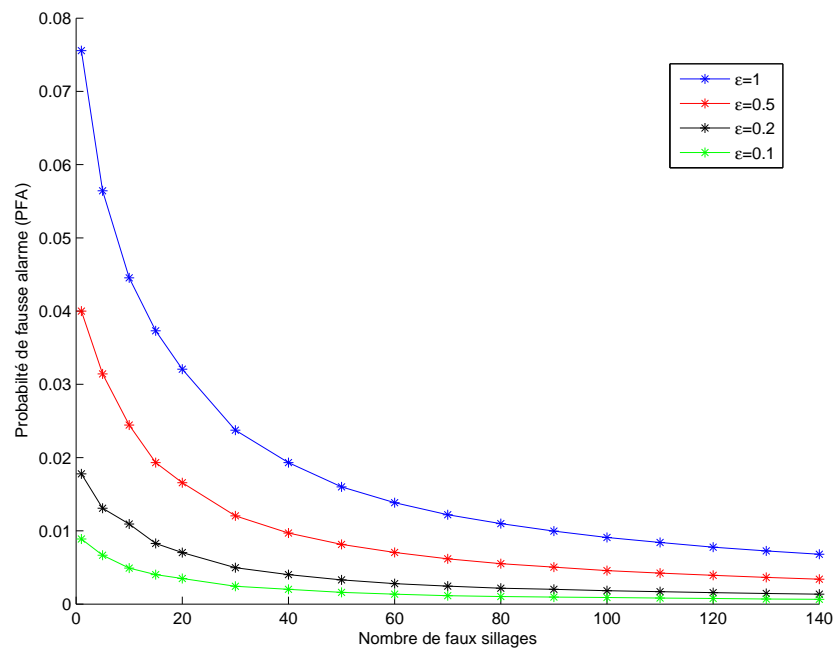


FIG. 4.21 – Courbe de probabilité de fausse alarme en fonction du nombre de faux sillages et pour $\epsilon=1, 0,5, 0,2$ et $0,1$.

Pour la suite des expérimentations sur le couplage, nous fixons $\epsilon = 0,2$ qui est un compromis entre une bonne probabilité de détection et une probabilité de fausse alarme raisonnable.

Dans l'annexe C, nous proposons une étude de l'influence du nombre de sillages et de l'angle θ sur r et l'influence du nombre de sillages et de r sur l'angle θ .

Conclusion

La méthode de couplage *a contrario* est simple et rapide. Elle permet de coupler un bateau avec son sillage. A partir d'un ensemble de sillages potentiels important, la méthode conserve uniquement ceux qui correspondent réellement au sillage du bateau et cela sans l'utilisation d'un modèle complexe.

Nous avons fixé $\epsilon=0,2$, mais étant donné l'influence du nombre de sillages candidats sur les performances en terme de détection et de fausses alarmes, il serait intéressant dans les perspectives de prévoir une valeur de ϵ qui varie en fonction du nombre de sillages candidats afin de maximiser la probabilité de détection et de minimiser la probabilité de fausse alarme.

4.4.4 Implémentation de la méthode

Dans la figure 4.8, nous avons présenté l'ensemble de la démarche pour l'approche *a contrario*. Pour chaque détection de bateau, nous avons obtenu son contour lors de l'étape de segmentation (chapitre 3). Nous prenons pour sa position le centre du contour et nous allons extraire une sous-image de taille 201×201 pixels ($N = 201$) autour de cette position.

Dans cette sous-image, nous allons appliquer la méthode de détection d'alignement que nous avons décrite en 4.4.2. A l'issue de cette étape, nous avons un ensemble de segments qui sont potentiellement des sillages. Les paramètres utilisés sont : $\epsilon = 10^7$ et $p = \frac{1}{16}$.

Pour chaque segment détecté, nous associons deux sillages candidats ; un dans chaque direction du segment ; c'est-à-dire que si nous détectons $Nb_{Segments}$ nous aurons $2 \times Nb_{Segments}$ sillages possibles pour l'étape de couplage bateau/sillage (figure 4.22).

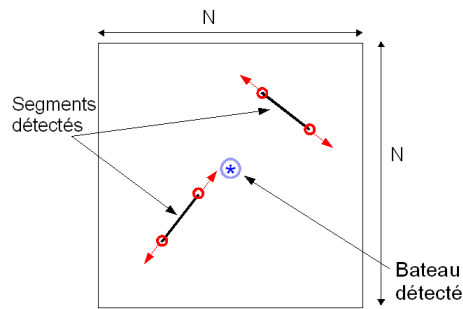


FIG. 4.22 – A partir de la position du bateau (en bleu), nous extrayons une sous-image de taille $N \times N$. Nous réalisons la détection de segments (méthode *a contrario*). Pour chaque segment détecté (en noir), deux sillages potentiels sont définis (en rouge) aux extrémités du segment.

Nous réalisons ensuite le couplage bateau/sillage exposé dans la partie 4.4.3 en utilisant le paramètre $\epsilon = 0, 2$.

Nous aurons à l'issue de cette étape, pour le bateau traité, l'ensemble des segments appartenant à son sillage.

Exemples de l'ensemble de la méthode

Dans la figure 4.23, Nous présentons deux exemples de l'ensemble de l'approche *a contrario*. Nous observons que seuls les segments appartenant au sillage du bateau sont conservés. Les nombreuses fausses alarmes de la figure 4.23(e) ont été éliminées et seul le sillage a été conservé (figure 4.23(f)).

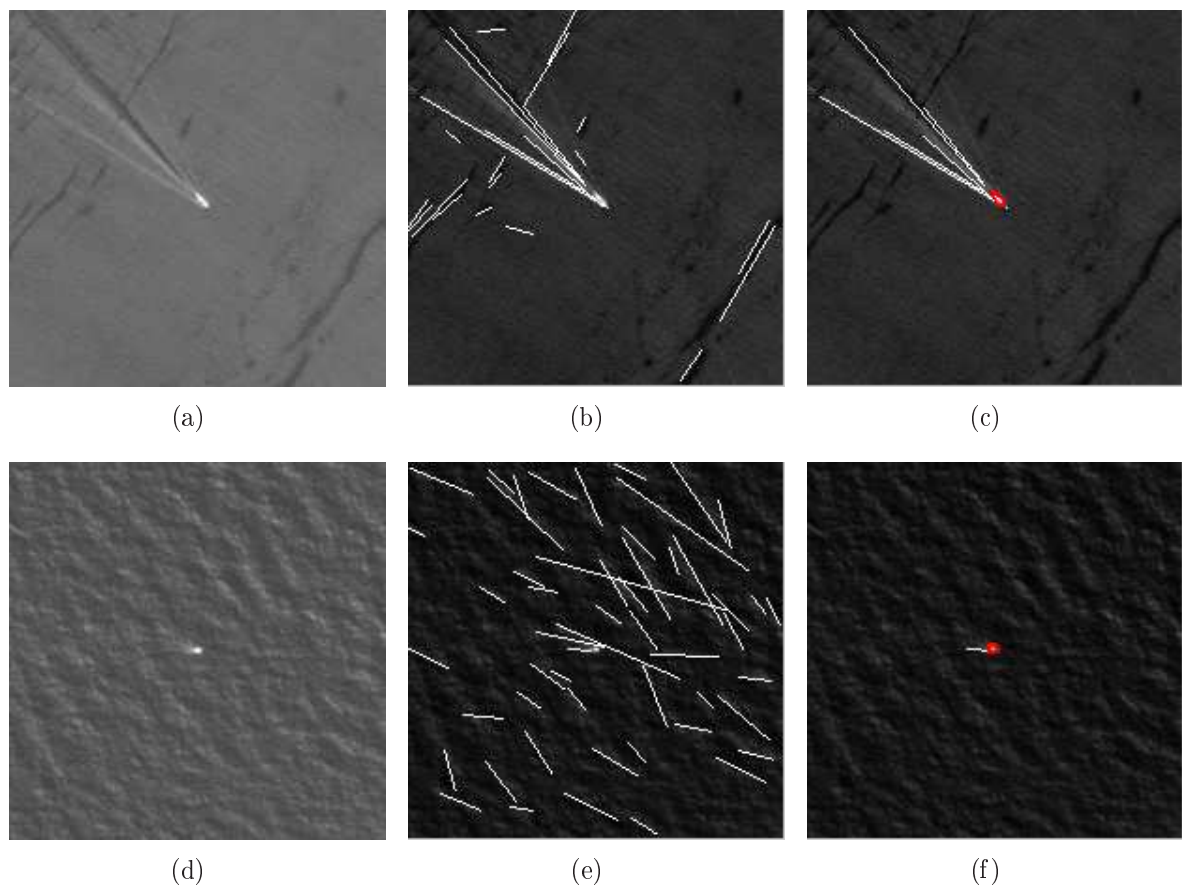


FIG. 4.23 – Exemples de l'ensemble de la méthode. (a)(d) sont les images traitées. (b)(e) sont les résultats de la détection des segments et (c)(f) sont les résultats du couplage.

4.4.5 Conclusion

L'approche *a contrario* possède plusieurs avantages. Tout d'abord, elle est non-supervisée. Un seul modèle des données est nécessaire (H_0) et en théorie, il n'y a pas de paramètres à fixer même si en pratique il faut tout de même ajuster le paramètre ϵ , mais cela reste une tâche assez aisée.

De plus, la méthode est rapide sur des images de tailles raisonnables (201×201 pixels).

En conclusion, cette approche nous donne des résultats satisfaisants pour notre problématique de détection des sillages. Nous allons maintenant expérimenter cette approche sur un échantillon de test, puis comparer les résultats obtenus avec ceux de l'**approche contours actifs**.

4.5 Expérimentation et résultats

Nous avons appliqué les deux méthodes sur le même échantillon de test et nous avons comptabilisé les cas où :

- le sillage est correctement détecté : NB_{Correct} ,
- un sillage est détecté, mais il est incorrect : $NB_{\text{Incorrect}}$,
- le sillage n'est pas détecté : $NB_{\text{NonDétecté}}$,
- un sillage est détecté alors qu'il n'y en a pas : $NB_{\text{FausseAlarme}}$,
- il n'y a pas de sillage dans l'image et il n'y a pas de sillage détecté : $NB_{\text{PasSillage}}$.

Les résultats sont synthétisés dans un tableau de la forme :

| | | Détecté | | Pas de sillage |
|-----------------------|-----------------------|---|---|--------------------------|
| | | Présence d'un sillage | | |
| R é s u l | Présence d'un sillage | Direction correcte NB_{Correct} | Direction non correcte $NB_{\text{Incorrect}}$ | $NB_{\text{NonDétecté}}$ |
| | Pas de sillage | $NB_{\text{FausseAlarme}}$ | | $NB_{\text{PasSillage}}$ |

4.5.1 Approche contours actifs

Nous présentons dans le tableau 4.1 les résultats obtenus en utilisant l'approche contours actifs.

Nous obtenons une probabilité de détection de 0,7830. Nous avons étudié les cas où nous avons une mauvaise détection du sillage. Nous avons identifié plusieurs types d'erreurs :

| | | Détecté | | |
|------------------|-----------------------|-----------------------|------------------------|----------------|
| | | Présence d'un sillage | | Pas de sillage |
| R é e l | Présence d'un sillage | Direction correcte | Direction non correcte | 16 |
| | Pas de sillage | 1 | | |

TAB. 4.1 – Résultats pour l'approche contours actifs.

- Nous avons une mauvaise orientation du sillage lorsqu'il est peu contrasté avec le fond et qu'il y a une mer agitée. Dans ce cas c'est une crête de vague qui peut être segmentée (figure 4.24(a)).
- Le sillage n'est pas détecté car il est trop loin du bateau (figure 4.24(b)).
- Le sillage n'est pas détecté car le contraste entre le fond et le sillage est faible (figure 4.24(c) 4.24(d)).

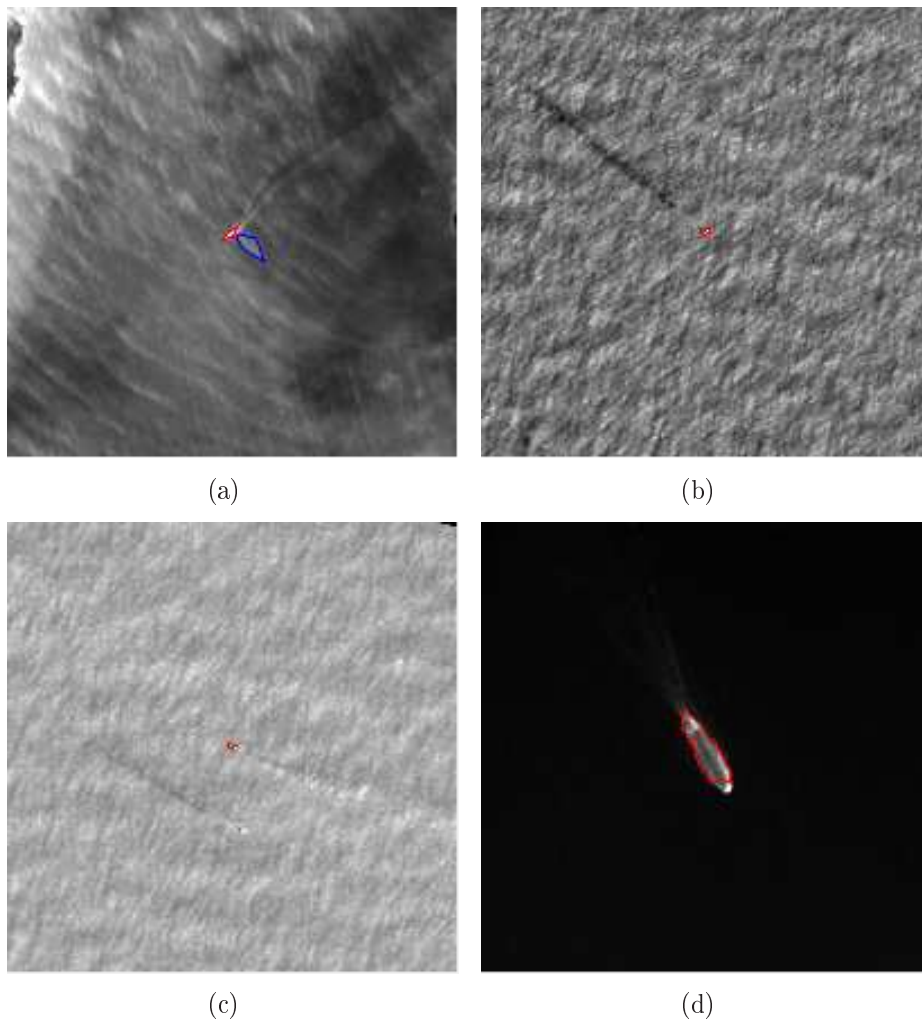


FIG. 4.24 – Exemples où l'approche contours actifs ne donne pas de bons résultats.

4.5.2 Approche *a contrario*

Nous présentons dans le tableau 4.2 les résultats obtenus en utilisant l'approche *a contrario*.

| | | Déte ct e | | |
|------------|-----------------------|-----------------------|------------------------|----------------|
| | | Pr sence d'un sillage | | Pas de sillage |
| R e sultat | Pr sence d'un sillage | Direction correcte | Direction non correcte | 14 |
| | | 74 | 8 | |
| | Pas de sillage | 2 | | 8 |

TAB. 4.2 – R sultats pour l'approche *a contrario*.

Nous obtenons une probabilit  de d tection de 0,7736.

Nous avons  tudi  les cas o  nous avons une mauvaise d tection du sillage. Nous avons observ  que lorsque les conditions m t orologiques sont mauvaises et que la mer est agit e les r sultats peuvent  tre mauvais. Dans ce cas, nous avons vu que le nombre de segments d tect s est important et que le couplage bateau/sillage n'est pas ou est mal r alis  (figure 4.25).

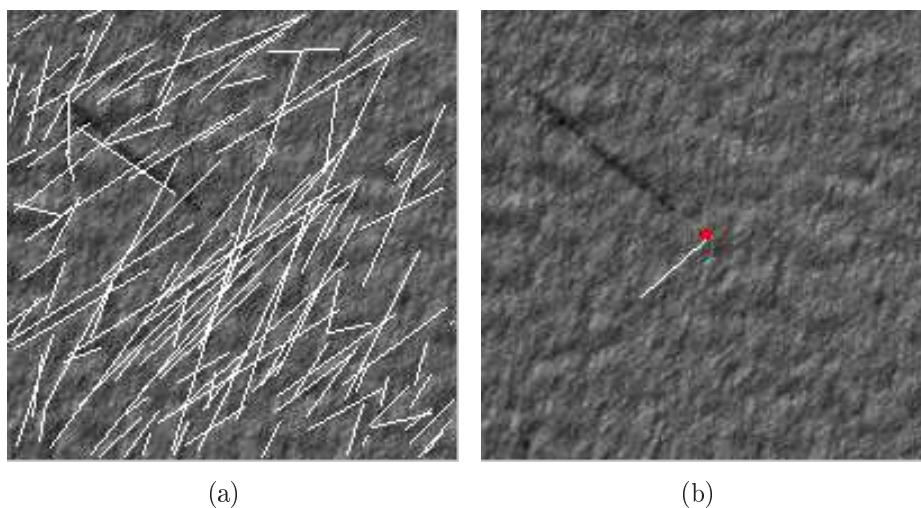


FIG. 4.25 – Exemple o  la m thode ne donne pas un bon r sultat.

4.6 Comparaison des deux approches

Nous obtenons un taux de r ussite l g rement plus  lev  pour l'approche contours actifs. Par contre, nous avons le m me nombre de sillages bien d tect s (74). C'est la d tection d'un sillage alors qu'il n'y en a pas qui fait la diff rence.

Les sillages bien détectés ne sont pas tous les mêmes. En effet, il n'y a que 58 sillages sur les 74 en commun. Cela signifie qu'en combinant les deux méthodes nous pourrions obtenir de meilleurs résultats. En effet, nous aurions un taux de réussite de 0,9340 au lieu de 0,7830.

Lorsque le sillage est constitué d'un seul élément, les deux méthodes donnent des résultats comparables comme dans la figure 4.26.

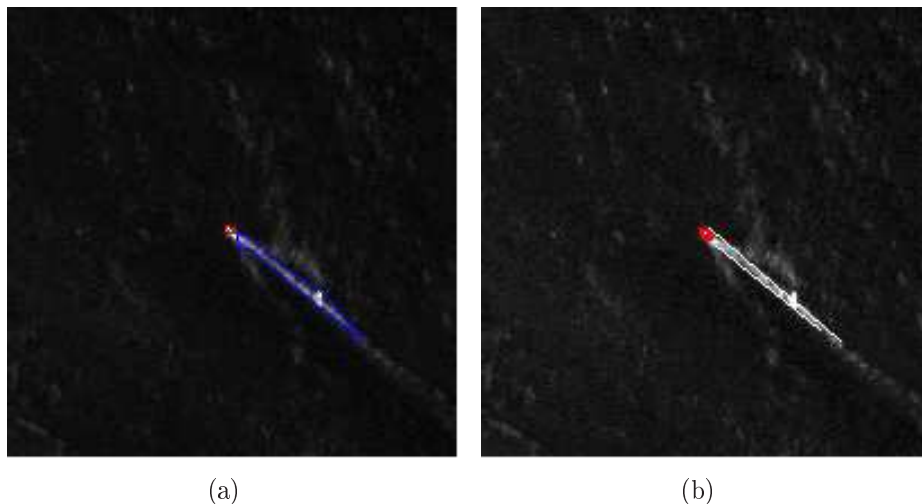


FIG. 4.26 – *Comparaison des résultats des deux approches. (a) approche contours actifs et (b) approche a contrario.*

Lorsque le sillage est constitué de plusieurs éléments (lorsque les vagues d'étrave sont visibles) alors la seconde méthode possède l'avantage de pouvoir associer à un bateau plusieurs éléments du sillage comme dans la figure 4.27. Dans ce cas l'approche *a contrario* est plus intéressante.

Lorsque le sillage est une ligne courbe alors l'approche contours actifs permet de mieux caractériser le sillage comme dans la figure 4.28 où nous pouvons voir la courbure du sillage.

Nous constatons que parfois l'approche *a contrario* donne un bon résultat alors que l'approche contours actifs détecte un mauvais sillage (figure 4.29) et vice versa. Parfois l'approche contours actifs donne un bon résultat alors que l'approche *a contrario* donne un résultat erroné (figure 4.30). Cela rend ces deux approches complémentaires.

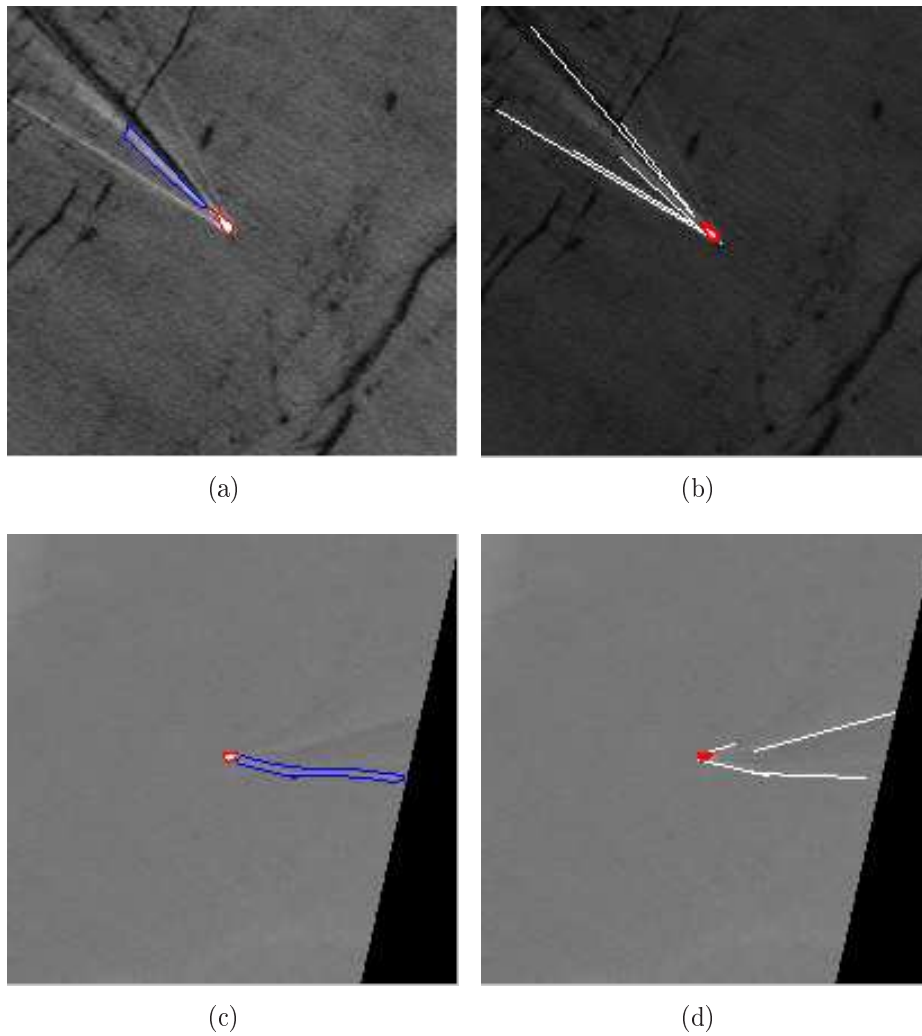


FIG. 4.27 – Comparaison des résultats des deux approches dans le cas d'un sillage composé de plusieurs éléments. (a)(c) approche contours actifs et (b)(d) approche a contrario.

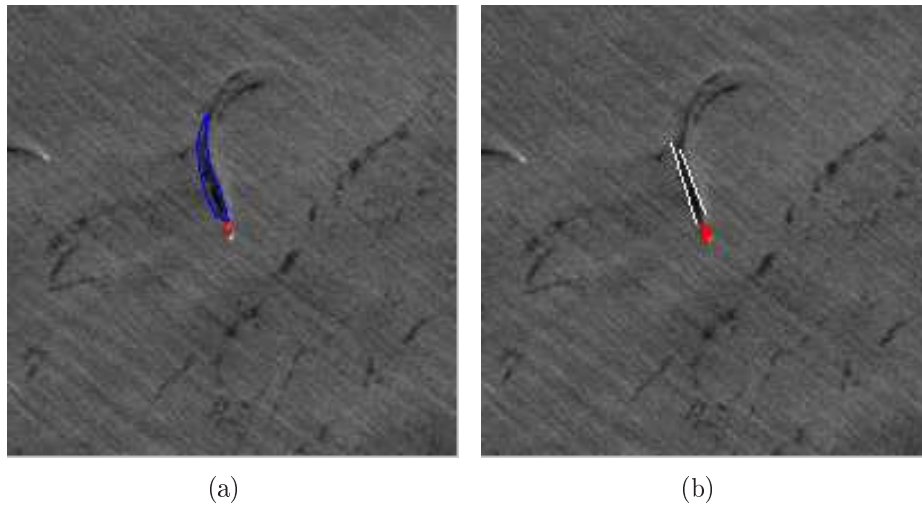


FIG. 4.28 – *Comparaison des résultats des deux approches dans le cas d'un sillage non-rectiligne. (a) approche contours actifs et (b) approche a contrario.*

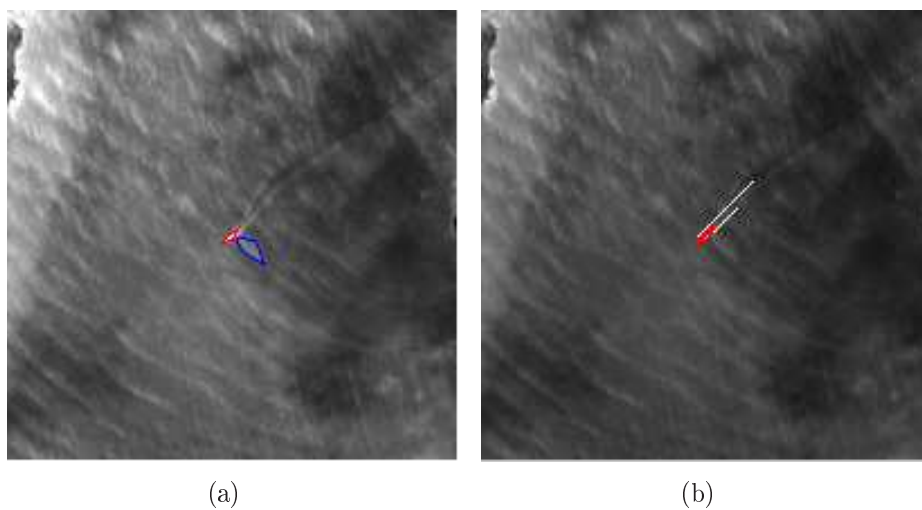


FIG. 4.29 – *Exemple où l'approche a contrario donne un résultat correct (b) alors que l'approche contours actifs a mal détecté la direction du sillage (a).*

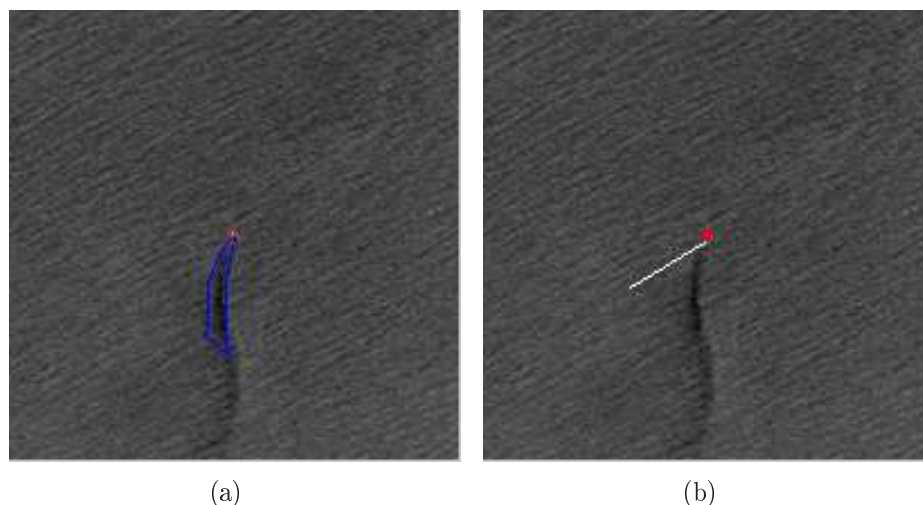


FIG. 4.30 – Exemple où l’approche contours actifs donne un résultat correct (a) alors que l’approche *a contrario* a mal détecté le sillage (b).

4.7 Conclusion

Dans ce chapitre, nous avons étudié deux approches pour la détection de sillage : une première approche utilisant les contours actifs et une seconde méthode basée sur une approche *a contrario*.

Dans l’approche contours actifs, nous utilisons la théorie de la décision bayésienne pour déterminer la présence d’un sillage et sa direction. Puis à l’aide de contours actifs nous segmentons le sillage.

Dans l’approche *a contrario* nous détectons des segments (par une approche *a contrario*) dans une région d’intérêt centrée sur le bateau. Puis, nous avons défini un couplage, basé sur une approche *a contrario*, entre le bateau et les segments.

Nous avons testé les deux méthodes sur un même jeu de test afin de pouvoir les comparer. Pour les deux approches, nous avons des taux de réussite assez proches, environ 0,78. Nous avons constaté que les cas de succès ne sont pas les mêmes et une fusion des informations des deux approches augmenterait le taux de réussite qui passerait à 0,93.

Dans les perspectives de recherche, il serait intéressant d’étudier comment combiner efficacement les deux méthodes pour améliorer les résultats.

Chapitre 5

Expérimentation et résultats de l'algorithme complet

5.1 Introduction

Encore une fois, cette thèse s'inscrit très volontairement dans un contexte industriel. De ce fait, après avoir caractérisé chaque étape au cours des chapitres précédents, il nous a semblé important de reprendre ici l'expérimentation, mais cette fois-ci dans un cadre opérationnel, afin de pouvoir conclure sur les performances de l'algorithme complet : ce que l'on peut en attendre et comment l'améliorer en se focalisant sur l'application visée plus que sur des considérations théoriques. Pour cela, nous allons présenter les résultats de l'ensemble de la méthode (pré-détection, segmentation, classification et détection des sillages) sur un ensemble d'images complètes.

Dans un premier temps, nous rappellerons les caractéristiques de la base d'images d'expérimentation (partie 5.2). Puis dans la partie 5.3, nous présenterons les résultats de la méthode complète, en détaillant étape par étape les résultats.

Dans la partie 5.4, nous exposerons les problèmes rencontrés et les solutions proposées pour l'utilisation de la méthode dans un contexte opérationnel. Nous proposerons également quelques optimisations possibles pour améliorer le temps de traitement.

Dans la partie 5.5, nous donnerons un exemple sur une image complète et enfin nous conclurons dans la partie 5.6.

5.2 La base d'expérimentation

Pour tester l'algorithme complet nous avons utilisé la base d'image 3 présentée dans la partie 1.6.3.

Nous en rappelons quelques caractéristiques :

- elle est composée de 23 images complètes,
- le nombre total de pixels utiles est de $2,24 \times 10^9$,
- les images contiennent 150 petits bateaux et 11 grands bateaux.

5.3 Les résultats

5.3.1 La pré-détection

Les paramètres de la pré-détection sont :

- $L_A=7$ pixels
- $L_w=3$ pixels

Nous présenterons dans le tableau 5.1 les résultats de la pré-détection sur la base d'expérimentation.

| | |
|---|--------|
| Nombre de pré-détections | 47658 |
| Nombre de fausses pré-détections | 47383 |
| Probabilité de détection des petits bateaux | 0,9867 |
| Probabilité de détection des grands bateaux | 1 |

TAB. 5.1 – Résultats de la pré-détection sur la base d'expérimentation.

Tous les grands bateaux ont été détectés. Par contre, deux petits bateaux ne sont pas détectés. Ces deux cibles très petites (environ 3 pixels) et peu contrastées avec le fond sont situées dans des zones nuageuses pour l'une et sur un fond non homogène pour l'autre.

La perte de ces deux cibles montre les limites du seuillage adaptatif dans des conditions difficiles : très petite cible peu contrastée avec le fond et située sur un fond non homogène.

Pour une image complète de 14000×14000 pixels (10000×10000 pixels utiles) :

- le nombre moyen de pré-détections est de 2127,
- le temps de calcul moyen est de 3.87 minutes.

Les fausses alarmes sont au nombre moyen de 2115 par image. De façon évidente, ce nombre est bien trop important pour que les pré-détections seules puissent être utiles à un opérateur humain.

5.3.2 La segmentation

Pour la segmentation nous avons fixé le nombre maximum d'itérations à 100.

Nous avons réalisé la segmentation des pré-détections de la base d'expérimentation afin d'évaluer le nombre de segmentations et le temps moyen pour une segmentation. Les résultats sont présentés dans le tableau 5.2.

Nous constatons que le nombre de segmentations est inférieur au nombre de pré-détections. C'est la conséquence de l'optimisation, décrite dans la partie 3.2.5, visant à limiter autant que possible le nombre de ces segmentations pour des pré-détections proches.

Nous constatons également que le nombre de contours obtenus est inférieur au nombre de segmentations réalisées. C'est la conséquence de la fusion des contours décrite dans la partie 3.2.5.

Le temps de traitement pour une segmentation est de 1,52 seconde. Il est à noter que le temps total de l'étape de segmentation va dépendre du nombre de pré-détections et donc du contenu de l'image (mer calme, présence de nuages, mer agitée).

| | |
|---|-------|
| Nombre de pré-détections | 47658 |
| Nombre de segmentations | 40460 |
| Nombre de contours obtenus à l'issue de la segmentation | 26825 |
| Temps de calcul moyen par segmentation (seconde) | 1,52 |

TAB. 5.2 – Nombre de pré-détections, nombre de segmentations, nombre de contours obtenus et temps de calcul moyen pour la base d'expérimentation.

5.3.3 La classification

Pour la classification, nous avons utilisé la méthode **FT-2** décrite dans le chapitre 3.

Nous présentons dans le tableau 5.3 les résultats de la classification des contours obtenus à l'issue de la segmentation.

| | Classe estimée | | |
|--------------|----------------|--------------|------------|
| | Petit bateau | Grand bateau | Pas bateau |
| Petit bateau | 147 | 0 | 1 |
| Grand bateau | 0 | 9 | 2 |
| Pas bateau | 2790 | 169 | 23707 |

TAB. 5.3 – Matrice de confusion de la méthode **FT-2** sur la base d'expérimentation

Probabilités de détection

La probabilité de détection des petits bateaux $\mathbf{PD}_{\text{petit}} = 0,9932$. Nous avons un petit bateau (figure 5.1(a)) qui est mal classé. En observant sa segmentation, nous remarquons qu'une partie de la vague d'étrave a été segmentée avec le bateau. Dans ce cas, nous avons un faible contraste entre le fond et la cible, et la présence du sillage très contrasté à proximité.

La probabilité de détection des grands bateaux $\mathbf{PD}_{\text{grand}} = 0,8182$. Nous avons deux grands navires mal classés. Si nous observons la segmentation du bateau de la figure 5.1(b), nous remarquons que des vagues sur les côtés du bateau ont été segmentées avec le bateau. De ce fait, sa forme diffère de la forme des bateaux présents dans l'échantillon d'apprentissage. Pour pallier ce problème, il faudrait un échantillon d'apprentissage plus important car pour le moment il est de seulement huit bateaux, ce qui est peu.

La segmentation du bateau de la figure 5.1(c) est incomplète. Cela est dû au nombre d'itérations qui est insuffisant pour ce navire. En effet, lorsque nous passons de 100 à 300 itérations nous obtenons le résultat de la figure 5.2 et dans ce cas la classification est correcte.

La valeur qui est suffisante pour la plupart des bateaux peut être insuffisante pour les bateaux de très grande taille. Nous pourrions envisager d'augmenter le nombre d'itérations et ainsi mieux segmenter les gros nuages qui sont parfois partiellement segmentés. En contre partie, cela va augmenter le temps de calcul. Or, le temps de traitement est un élément important de notre problématique. Il serait plus judicieux d'envisager la modification de l'initialisation du contour actif dans le cas des grands bateaux et de garder le nombre d'itérations à 100.

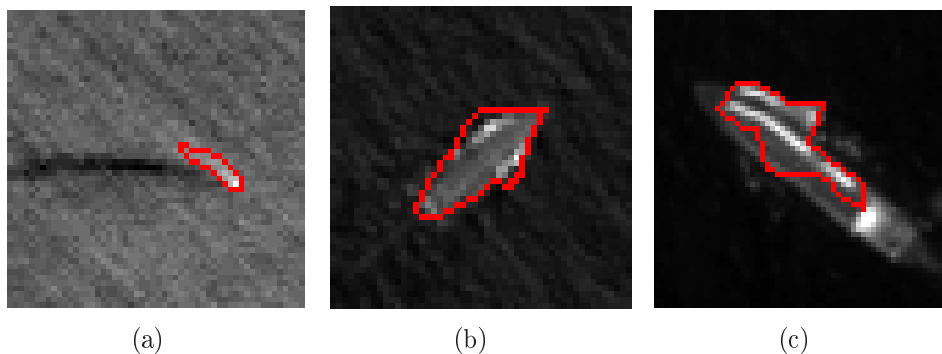


FIG. 5.1 – Illustration des trois erreurs de classification.

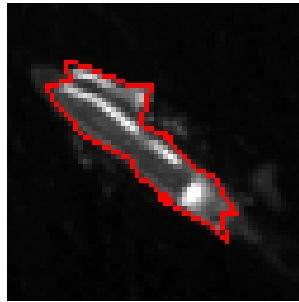


FIG. 5.2 – 300 itérations au lieu de 100.

Nombre de fausses alarmes

Le nombre total de fausses alarmes $Nb_{FA} = 2959$. Les fausses alarmes qui restent sont dues au fait que la distinction entre un très petit bateau et une crête de vague n'est pas aisée et, au final, même si un bon nombre de ces crêtes a été éliminé, il en reste néanmoins 2790. Il reste également 169 éléments qui sont classés comme des grands bateaux et qui n'en sont pas. La plupart de ces éléments sont des nuages.

5.3.4 L'algorithme complet

Dans le tableau 5.4 nous présentons les résultats finaux de l'algorithme complet sur la base d'expérimentation.

| | |
|---|--------|
| Nombre total de fausses détections | 2959 |
| Probabilité de détection des petits bateaux | 0,9735 |
| Probabilité de détection des gros bateaux | 0,8182 |

TAB. 5.4 – Résultats de l'algorithme complet sur la base d'expérimentation.

Pour une image complète de 14000×14000 pixels (10000×10000 pixels utiles) :

- le nombre moyen de pré-détections est de 2127,
- le nombre moyen de fausses détections est de 90,
- le temps de calcul moyen est d'environ 35 minutes.

Probabilités de détection

Nous obtenons une probabilité de détection des petits bateaux de **0,9735**. Les cas où les petits bateaux ne sont pas détectés sont des cas assez particuliers. Dans l'ensemble cette méthode est très efficace pour la détection de petits bateaux.

Concernant les grands bateaux nous avons une probabilité de détection plus faible

0,8182. Nous avons vu que cela est dû à une segmentation parfois médiocre des grands bateaux et certainement à un classifieur non optimum étant donné le peu d'exemples à notre disposition pour l'apprentissage.

Nombres de fausses alarmes

L'ajout de la segmentation et de la classification à l'étape de pré-détection réduit le nombre de fausses alarmes d'un facteur 20. Celui-ci est désormais de l'ordre de 90 pour une image complète.

Le nombre de fausses alarmes est alors beaucoup plus raisonnable dans le contexte d'un post-traitement par un photo-interprète. Nous pouvons, en effet, considérer que la validation manuelle des vrais bateaux sera beaucoup plus rapide que le traitement complet de l'image par l'opérateur. Toutefois, ce nombre de fausses alarmes n'est pas encore assez faible pour envisager une automatisation complète de la détection des bateaux.

Temps de calcul

Le temps de calcul est de l'ordre de 35 minutes en moyenne pour le traitement d'une image complète de 14000×14000 pixels. Ce temps a été mesuré, sans optimisation du code, pour un algorithme maquette programmé en matlab, sur un ordinateur portable standard et non spécifiquement optimisé pour en réduire le temps de calcul. Nous verrons plus tard que ce temps de calcul pourra être aisément réduit.

Le temps de calcul et le nombre de fausses alarmes sont fortement liés au contenu de l'image et en particulier à la présence de nuages et d'une mer agitée.

5.4 Problèmes rencontrés et les solutions proposées

Dans cette partie, nous allons faire la synthèse des problèmes rencontrés par la méthode, puis nous donnerons quelques solutions pour son utilisation dans un contexte opérationnel.

Les fausses alarmes dues à l'état de la mer

Dans une image complète, certaines zones peuvent présenter de nombreuses crêtes de vague dues aux conditions météorologiques. Nous avons constaté que ce phénomène génère

un grand nombre de fausses alarmes comme nous pouvons le constater dans la figure 5.3. Parmi les fausses alarmes, nous avons estimé que plus de 90% de celles-ci étaient dues à la présence de crêtes de vague.

Finalement cela augmente le temps de calcul, nous avons vu qu'il était fortement lié aux nombre de pré-détections qui est lui même lié au contenu de l'image. Le nombre de fausses alarmes est élevé car il peut y avoir confusion entre un très petit bateau et les crêtes de vague.

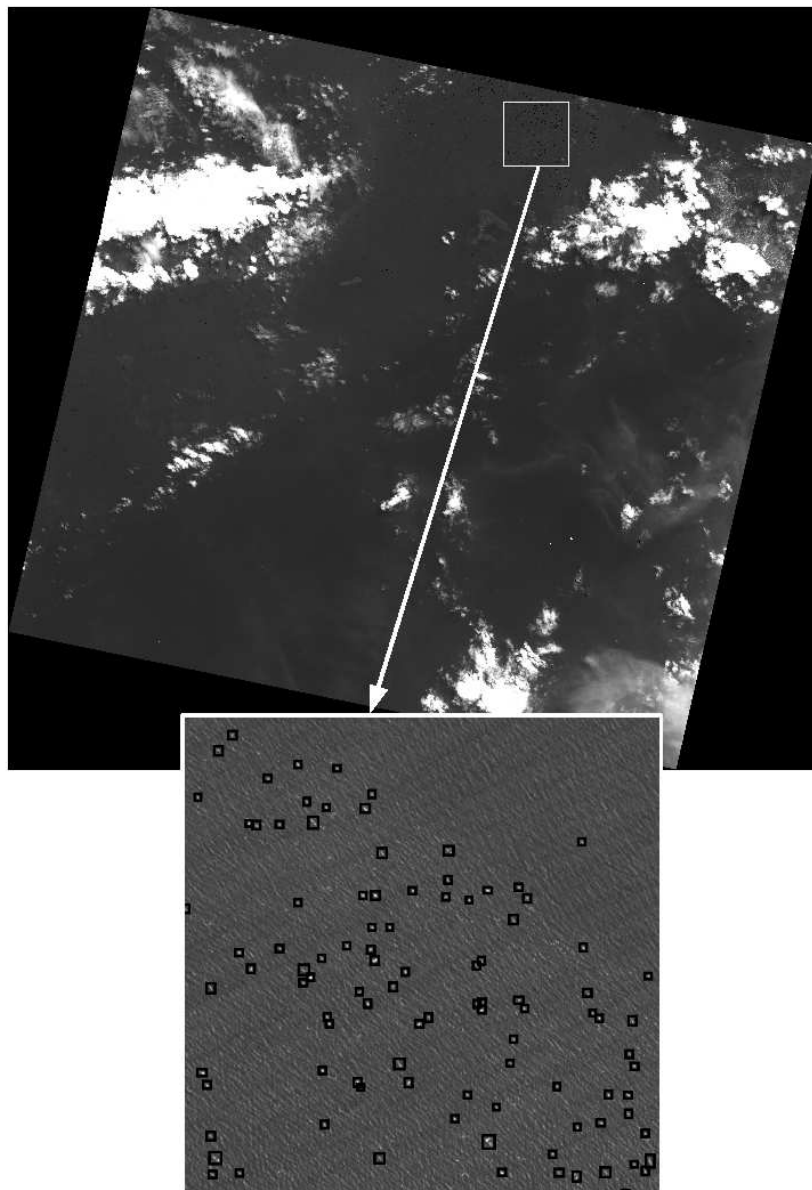


FIG. 5.3 – Certaines zones de cette image présentent de nombreuses fausses alarmes (carrés noirs) dues aux crêtes de vague.

Pour les traitements, les images sont découpées en dalles. Nous pouvons donc envisager de renoncer à traiter une dalle quand l'état de la mer génère trop de pré-détections. En effet, les nombreuses fausses alarmes rendent les résultats difficiles à exploiter. De plus, le traitement de ces fausses alarmes prend un temps de calcul important. Dans le cas de l'aide à l'opérateur humain, il faudra le signaler d'une manière visuelle afin qu'il sache rapidement que la dalle n'a pas été traitée automatiquement et qu'il doit le faire manuellement.

Nous pourrions déterminer un seuil pour le nombre de pré-détections. Au-delà de ce seuil, nous ne traiterons pas la dalle ou bien il faudrait utiliser une méthode pour caractériser l'état de la mer et en fonction de celui-ci de décider ou non du traitement de la dalle.

Temps de calcul

Le temps moyen de calcul qui est d'environ 35 minutes cache de grandes disparités. En effet, nous pouvons aller d'un temps de traitement de 3 minutes à 1 heure. La partie segmentation prend beaucoup de temps et dépend du nombre de pré-détections. La segmentation d'une pré-détection prend environ 1,5 seconde. S'il y a de nombreuses pré-détections alors le temps total de segmentation peut être important.

Donc pour réduire le temps de calcul, nous pourrions paralléliser certaines tâches. Nous avons vu dans la partie 2.2.12 que nous travaillons dalle par dalle. Nous pouvons donc facilement traiter ces dalles en parallèle. A l'intérieur d'une même dalle on peut également envisager de réaliser les segmentations en parallèle. Ce qui permettrait aussi d'augmenter le nombre d'itérations et ainsi d'améliorer la segmentation des grands bateaux et des nuages et donc améliorer les résultats de la classification.

Le temps de calcul pourrait être aisément divisé par 10 en changeant de langage. Du fait de la lenteur avec laquelle Matlab traite les boucles, très nombreuses dans la segmentation, le temps pris par la phase de segmentation est très important.

On pourrait aussi envisager de modifier la segmentation en utilisant une méthode plus rapide comme par exemple une segmentation par la méthode des k-means [46] à 2 classes et en gardant les pixels connexes à la position de la pré-détection. Cela n'a pas encore été testé et constitue une perspective de recherche.

Nous pouvons aussi envisager l'utilisation d'un processeur graphique ou GPU (Graphics Processing Unit) pour réaliser certains calculs.

5.5 Exemple

Nous présentons ici le résultat de la méthode sur une image complète SPOT 5 panchromatique de résolution 5m et de taille 140000×140000 pixels (figure 5.4).

Cette scène contient 17 petits bateaux et 5 grands bateaux.

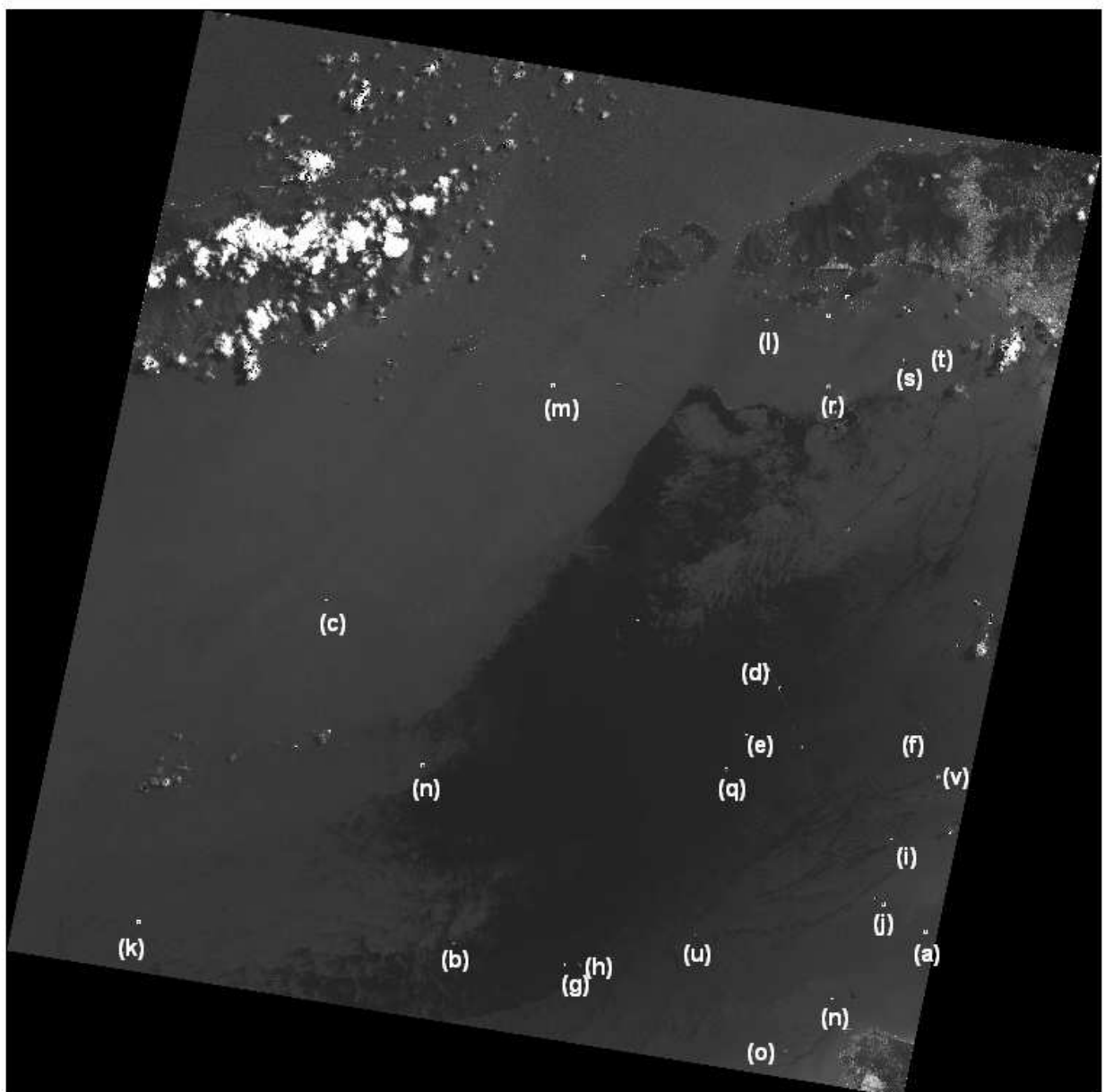


FIG. 5.4 – Image SPOT 5 panchromatique de résolution 5m et taille 140000×140000 pixels. Les différentes lettres entre parenthèses correspondent aux bateaux détectés.

Pour le traitement de cette image, nous avons 409 pré-détections, 330 segmentations

sont réalisées et au final nous obtenons la détection des 17 petits bateaux et des 5 grands bateaux plus 15 fausses alarmes. Le temps de traitement de l'image est d'environ 5 minutes. Le tableau 5.5 résume ces différentes informations.

| | |
|---|-----|
| Nombre de pré-détections | 409 |
| Nombre de segmentations | 330 |
| Nombre final de fausses alarmes | 15 |
| Probabilité de détection des petits bateaux | 1 |
| Probabilité de détection des grands bateaux | 1 |
| Temps de calcul total (minutes) | 5,3 |

TAB. 5.5 – Résumé de l'exécution de l'algorithme sur l'image de la figure 5.4.

5.5.1 Les petits bateaux détectés

Dans les figures 5.5 à 5.21, nous présentons les petits bateaux détectés avec pour chacun la détection du sillage par l'**approche contours actifs** (image du milieu) et l'**approche *a contrario*** (image de droite).

Bateaux dont le sillage est détecté par les deux approches

Pour les bateaux **(a)** (figure 5.5), **(b)** (figure 5.6), **(c)** (figure 5.7), **(d)** (figure 5.8), **(e)** (figure 5.9), **(f)** (figure 5.10), **(g)** (figure 5.11), **(h)** (figure 5.12), **(i)** (figure 5.13), **(j)** (figure 5.14) et **(k)** (figure 5.15), un sillage est détecté pour les deux approches : **contours actifs** et ***a contrario***.

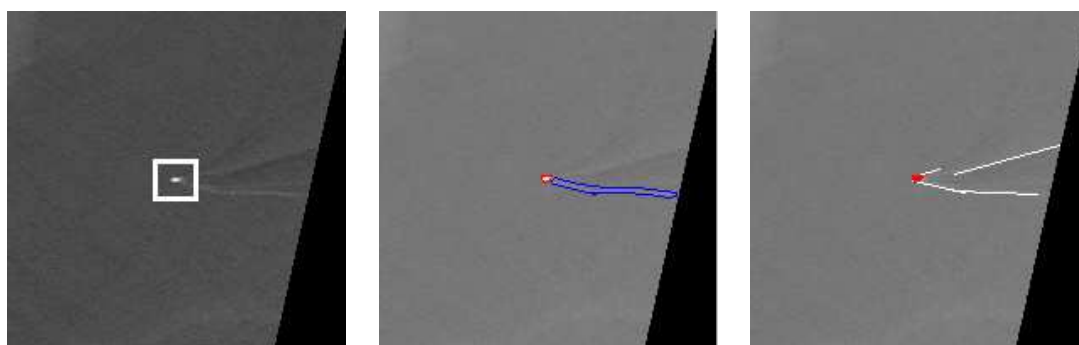
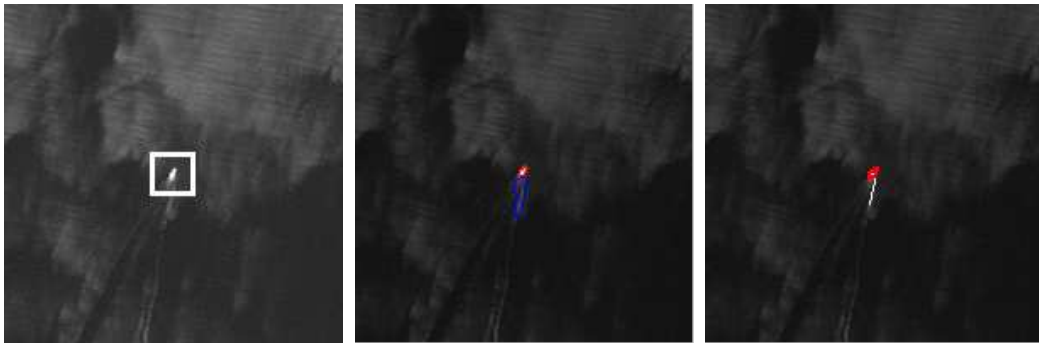
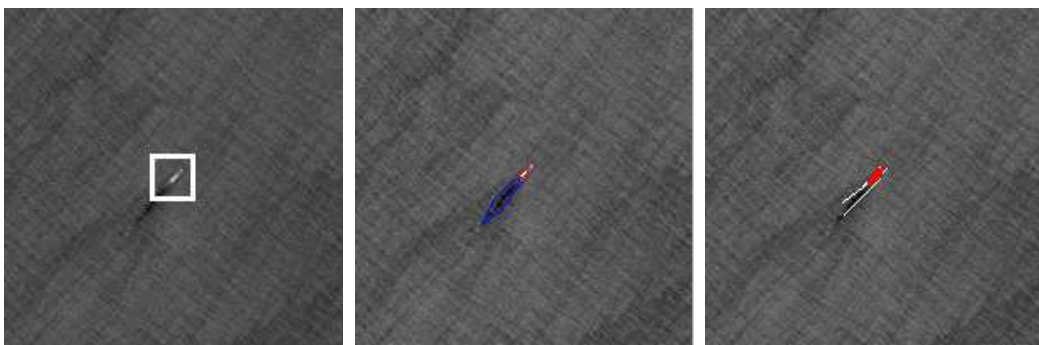
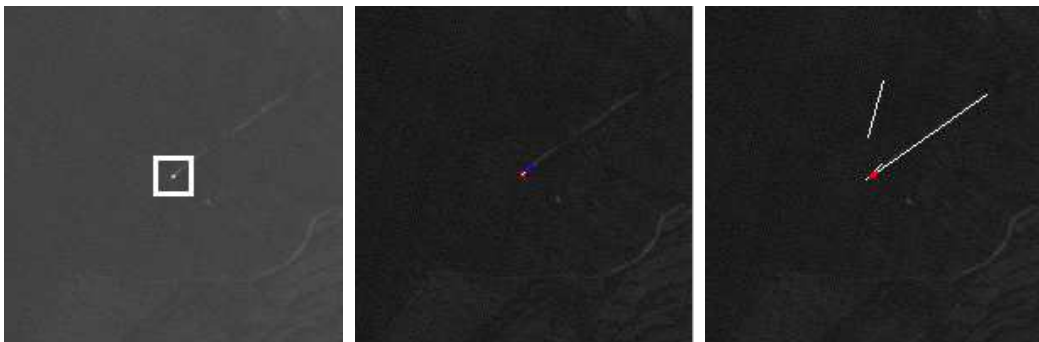
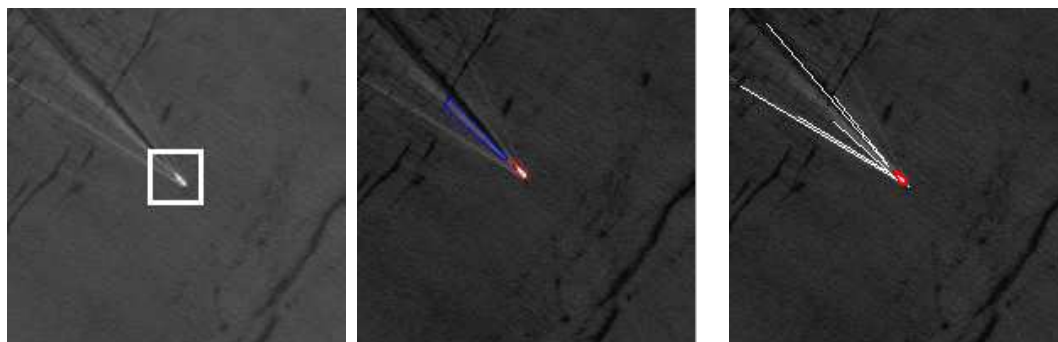
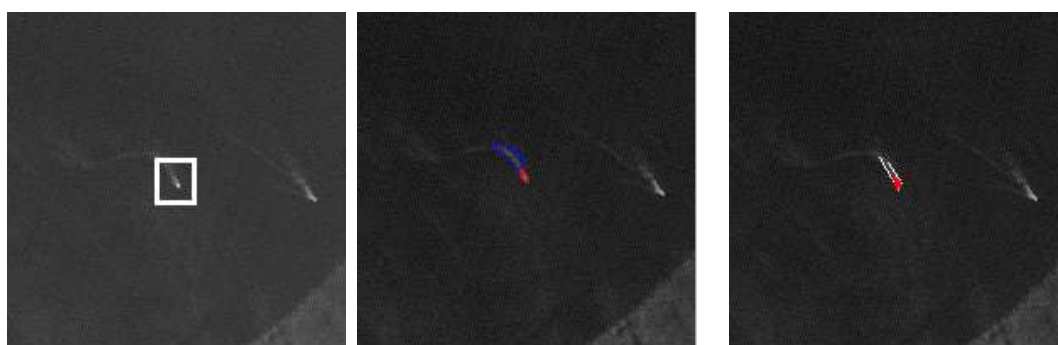
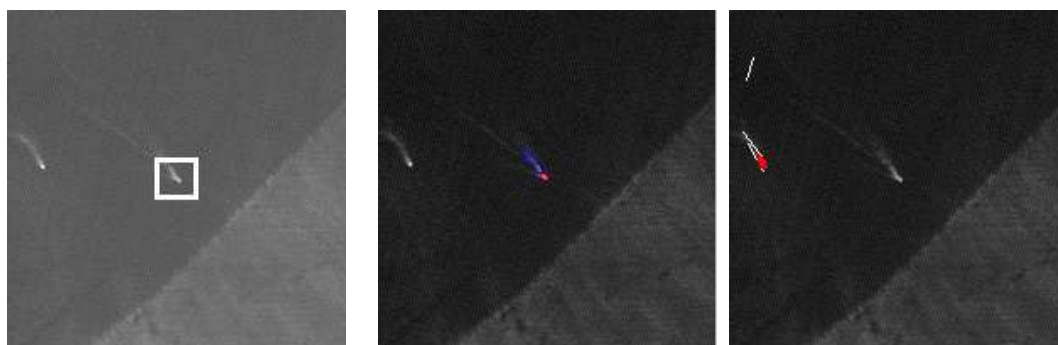
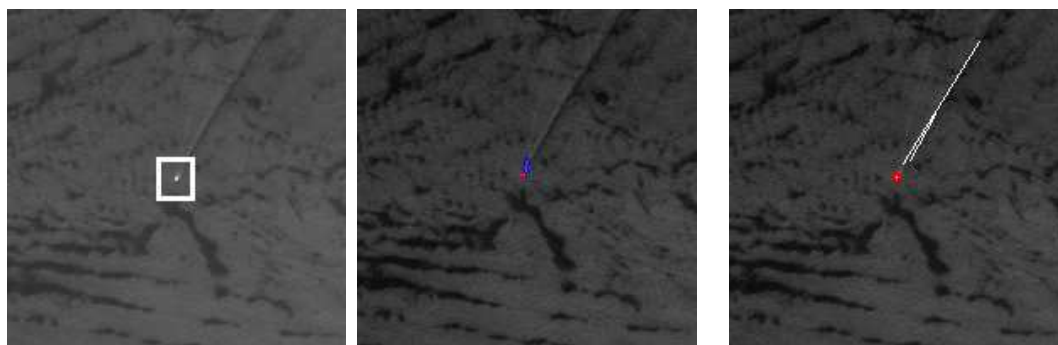
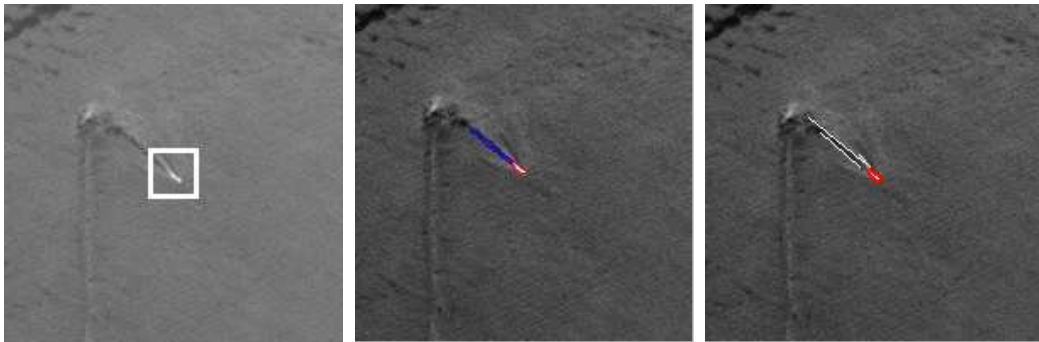
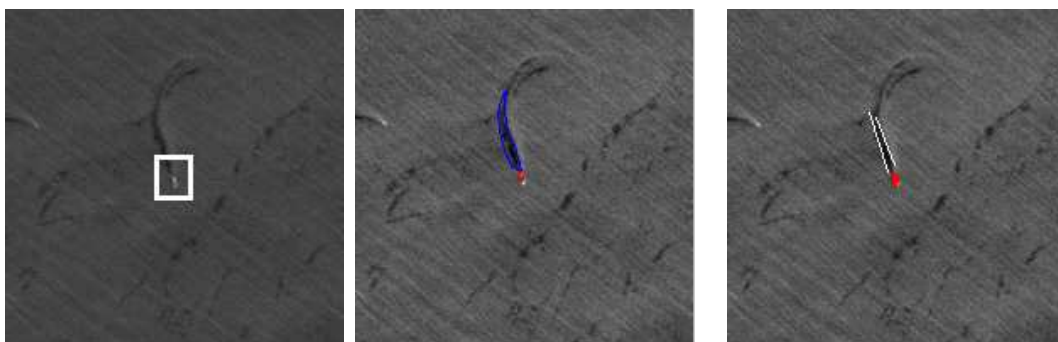


FIG. 5.5 – Bateau (a).

FIG. 5.6 - *Bateau (b)*.FIG. 5.7 - *Bateau (c)*.FIG. 5.8 - *Bateau (d)*.FIG. 5.9 - *Bateau (e)*.

FIG. 5.10 - *Bateau (f)*FIG. 5.11 - *Bateau (g)*.FIG. 5.12 - *Bateau (h)*.FIG. 5.13 - *Bateau (i)*.

FIG. 5.14 - *Bateau (j)*.FIG. 5.15 - *Bateau (k)*.

Bateaux dont le sillage est détecté par l'approche *a contrario*

Pour les bateaux **(l)** (figure 5.16), **(m)** (figure 5.17), **(n)** (figure 5.18) et **(o)** (figure 5.19), un sillage est détecté par l'approche *a contrario*, mais pas par l'approche contours actifs.

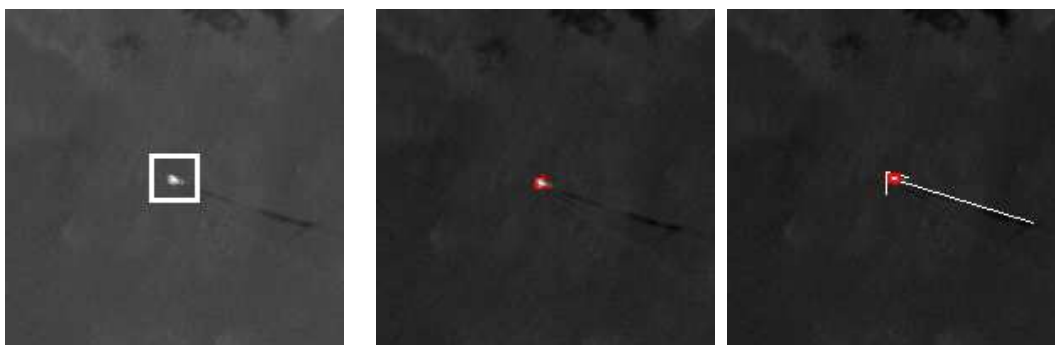


FIG. 5.16 - Bateau (l).

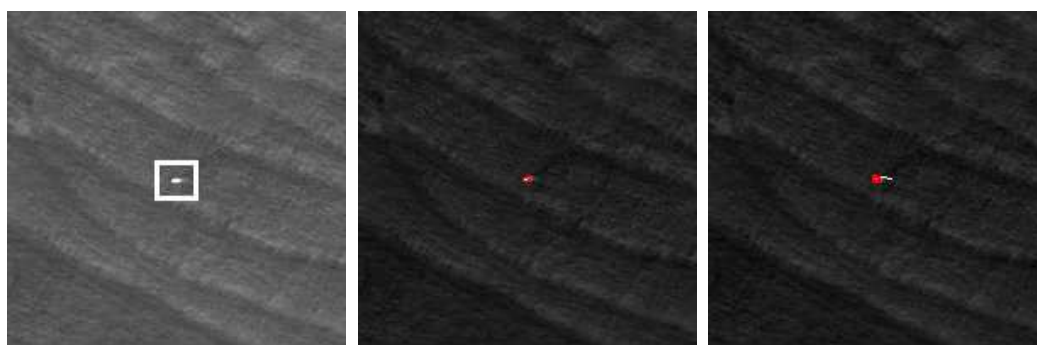


FIG. 5.17 - Bateau (m).

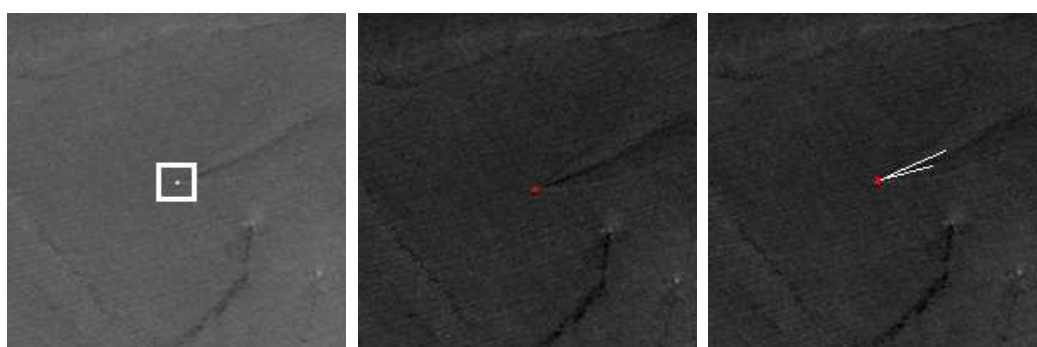
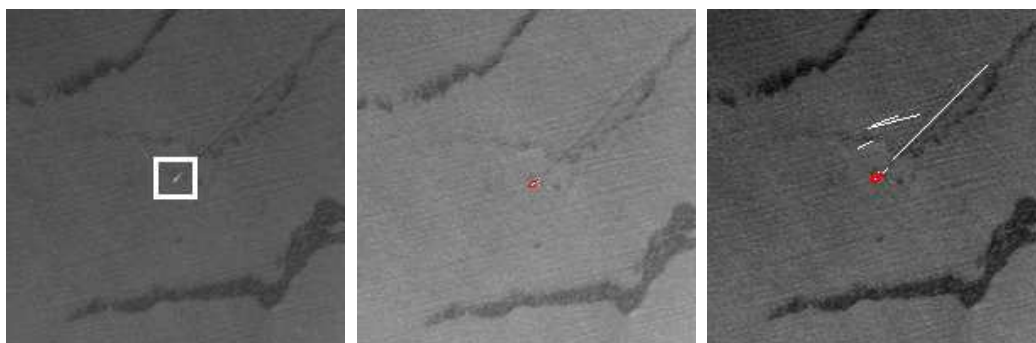


FIG. 5.18 - Bateau (n).

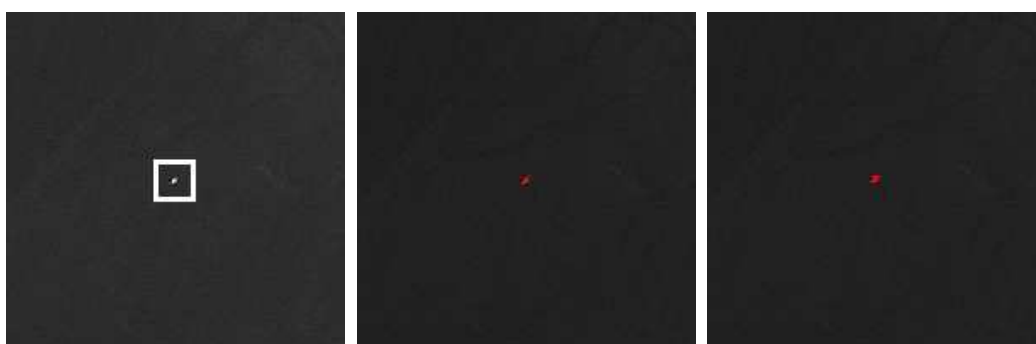
Bateau dont le sillage est détecté par l'approche contours actifs

Pour le bateau **(p)** (figure 5.20), un sillage est détecté par l'approche contours actifs, mais pas par l'approche *a contrario*.

FIG. 5.19 - *Bateau (o)*.FIG. 5.20 - *Bateau (p)*.

Bateau sans sillage

Le bateau (**q**) ne possède pas de sillage significatif et nous constatons dans la figure 5.21 que les deux méthodes de détection des sillages ne détectent pas de sillages.

FIG. 5.21 - *Bateau (q)*.

5.5.2 Les grands bateaux détectés

Les figures 5.22, 5.23, 5.24, 5.25 et 5.26 présentent les grands bateaux détectés. Les cinq bateaux sont à l'arrêt et aucun sillage n'est détecté. Nous présentons donc uniquement le résultat de la segmentation (image de droite).

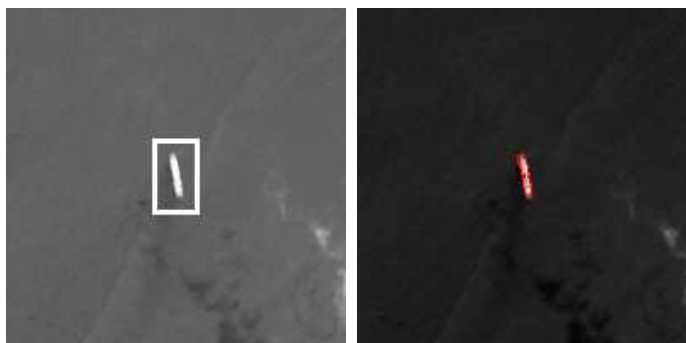


FIG. 5.22 - *Bateau (r)*.

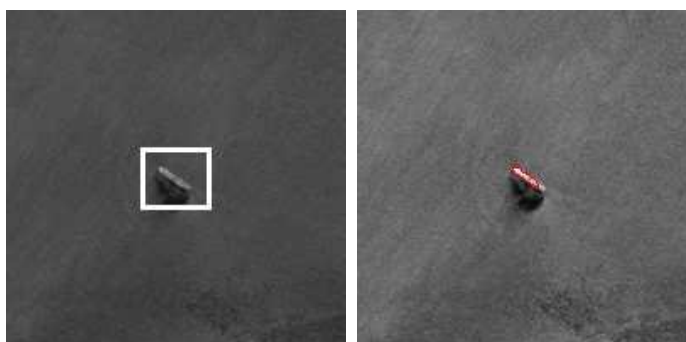


FIG. 5.23 - *Bateau (s)*.

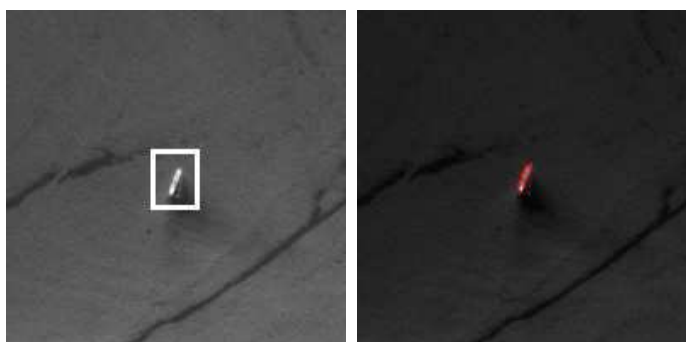


FIG. 5.24 - *Bateau (t)*.

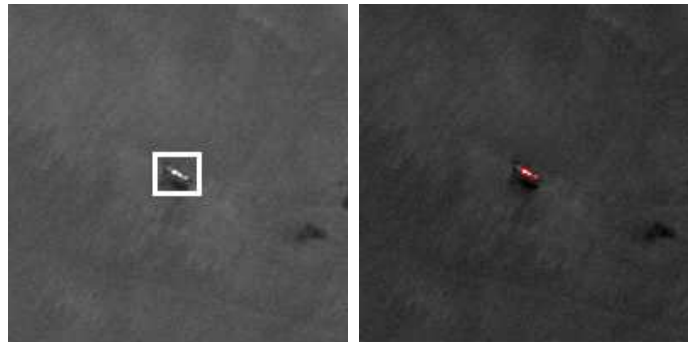


FIG. 5.25 – Bateau (u).

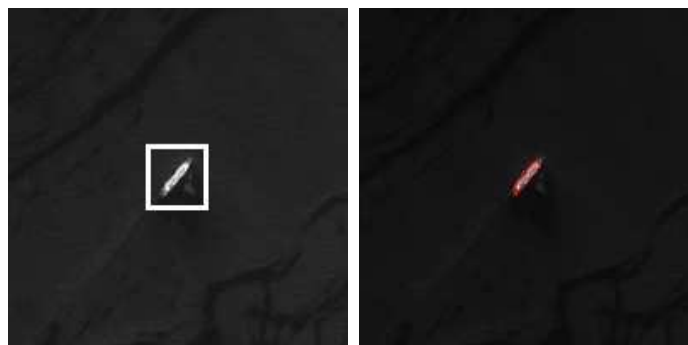


FIG. 5.26 – Bateau (v).

5.5.3 Les fausses alarmes

Il y a 15 fausses alarmes que nous pouvons répartir en trois groupes :

- Le premier groupe est présenté dans la figure 5.27. Il comporte quatre fausses alarmes liées à des zones de mer agitée.
- Le second groupe est présentée dans la figure 5.28. Il contient cinq fausses alarmes dues à un long sillage très contrasté avec le fond pour lequel la phase de pré-détection a généré plusieurs pré-détections. Il est à noter que ce type d'éléments n'était pas présent dans notre échantillon d'apprentissage.
- Le troisième groupe est présenté dans la figure 5.29. Il est composé de six fausses alarmes, que nous n'avons pas pu identifier. Il est à noter que ce type d'éléments n'était pas présent dans notre échantillon d'apprentissage.

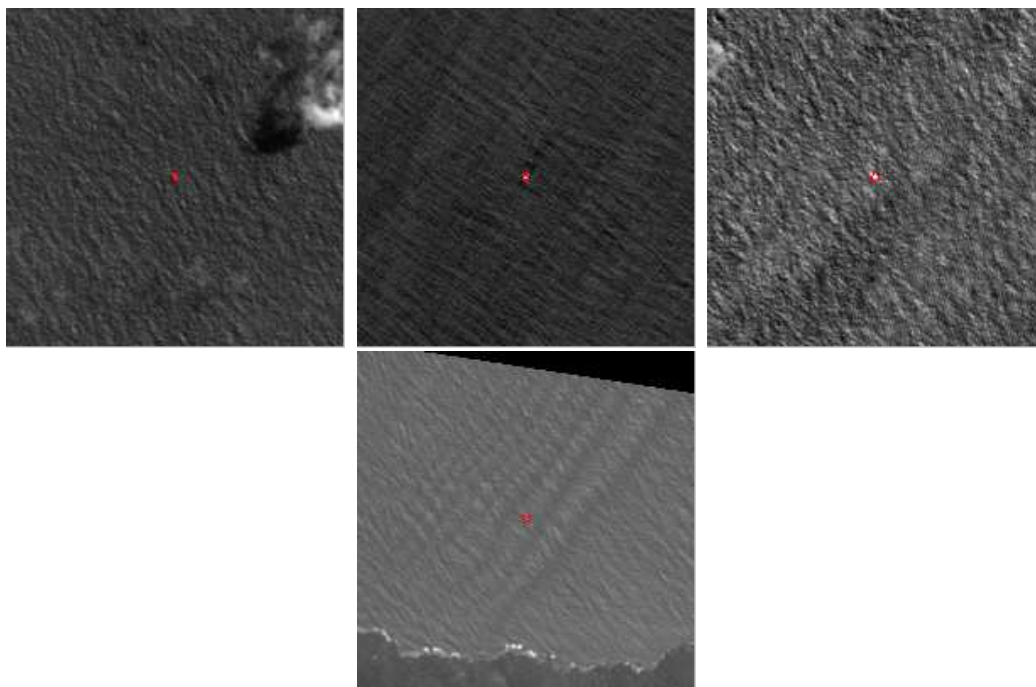


FIG. 5.27 – *Fausses alarmes dues à une mer agitée.*

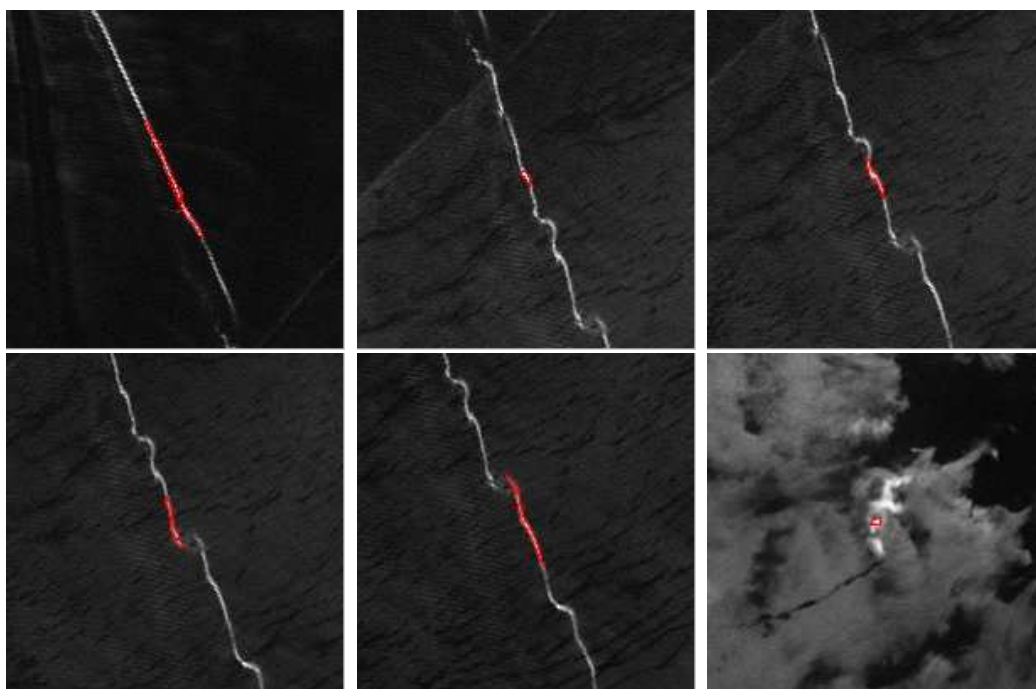


FIG. 5.28 – *Fausses alarmes dues à la présence d'un sillage qui a été pré-déTECTÉ, segmenté et mal classé.*

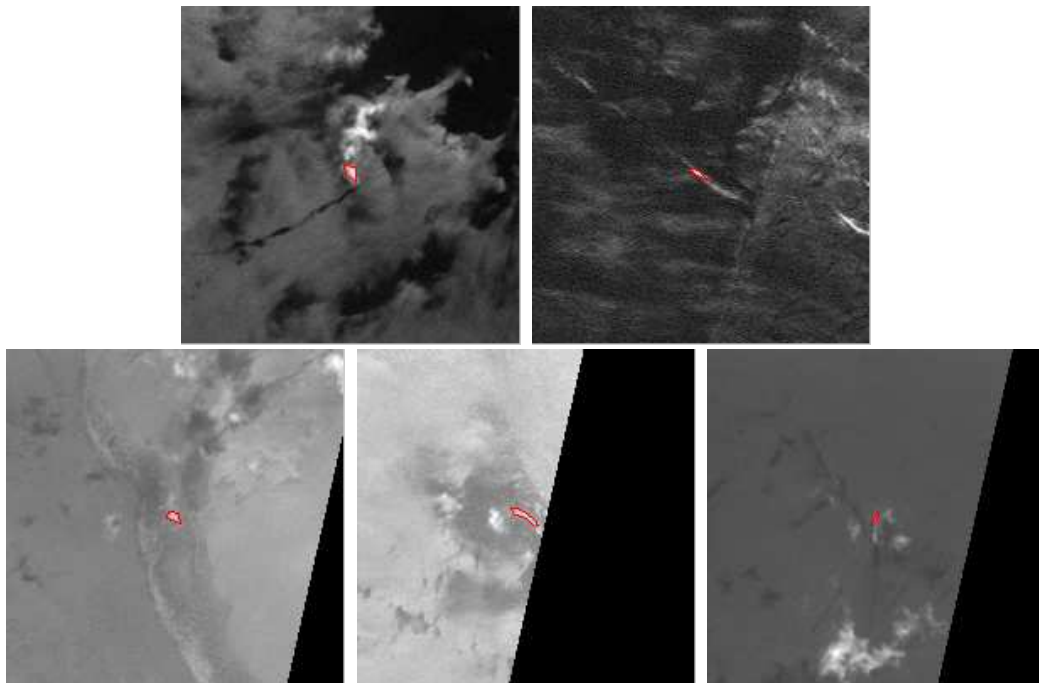


FIG. 5.29 – *Fausse alarmes que nous n'avons pas pu identifier.*

5.6 Conclusion

Dans ce chapitre, nous avons présenté et commenté les résultats de l'algorithme complet (pré-détection, segmentation, classification et détection des sillages) sur les images de la base d'image 3 (composé d'images complètes).

Les résultats sont satisfaisants en terme de probabilité de détection : 0,9735 pour les petits bateaux et 0,8182 pour les grands bateaux. Le nombre moyen de fausses détections est de 90 par image. Il reste toutefois quelques problèmes à régler concernant le traitement des zones de mer présentant de nombreuses crêtes de vagues. L'amélioration du traitement de ces zones réduirait de façon importante le nombre de fausses détections. En effet, elle représente plus de 90% des fausses alarmes. Nous avons proposé une solution pour pallier à ce problème dans un contexte opérationnel en attendant de trouver d'autres solutions pour améliorer le traitement de ces zones. Elle consiste à identifier automatiquement ces zones problématiques et de laisser l'opérateur les traiter manuellement.

Nous avons également proposé des solutions pour réduire le temps de calcul en parallélisant certains traitements.

Nous avons également vu que notre segmentation était adaptée à la segmentation des petits bateaux, mais que le nombre d'itérations choisi n'était pas assez important pour certains grands bateaux et les gros nuages. Il faudrait, sans doute, aussi revoir l'initialisation du contour actif dans ces cas. On pourrait aussi envisager d'autres méthodes de segmentation plus rapides comme, par exemple, la méthode des K-means.

Conclusions et perspectives

Conclusions

L'objectif de cette thèse était d'étudier la possibilité d'automatiser, partiellement ou totalement, la détection de bateaux à partir d'images satellitaires optiques haute résolution. Ce domaine assez récent reste encore très peu étudié et il n'existe pas encore d'outils de détection automatique sur ce type d'images. La thèse se place clairement dans une approche « Recherche et Développement ». Le but est de fournir des méthodes efficaces, rapides et robustes, tout en étant justifiées théoriquement.

Cette thèse s'insérant dans un contexte industriel bien défini (OCEANWAY), les données disponibles étaient uniquement des images SPOT 5 panchromatique de résolution 5m. L'ensemble de notre étude a donc concerné ce type d'images. De plus, des contraintes opérationnelles avaient été fixées en termes :

- de performances : un maximum de bateaux détectés, avec un minimum de fausses détections,
- de temps de traitement : moins de 45 minutes, ce qui est le temps nécessaire à l'opérateur humain pour traiter une image complète.

La première étape de notre travail a été de construire une base d'images utilisée pour tester nos algorithmes. Elle constitue aujourd'hui pour nous une base de référence pour tester et comparer, facilement et rapidement, les performances d'autres algorithmes.

Nous avons proposé une méthode de détection de bateaux composée d'une étape de pré-détection, pour définir des régions d'intérêt, suivie d'une étape de segmentation et de classification.

L'algorithme de pré-détection est basé sur la théorie bayésienne de la décision. Nous avons modifié le modèle initial de la méthode. Cela nous a permis d'obtenir de meilleurs résultats en termes de performances (probabilité de détection et nombre de fausses alarmes). Un résultat intéressant dans cette étude est que le modèle théorique ainsi modifié est plus simple que l'original. Il nécessite l'estimation de moins de paramètres, ce qui explique ses meilleures performances sur de petites cibles. La méthode est rapide (complexité en

$O(N)$), ce qui constitue un atout majeur. Elle détecte plus de 98% des petits bateaux et 100% des grands bateaux. A l'issue de ce premier traitement, il reste néanmoins un nombre important de fausses pré-détections.

Le but de l'étape de segmentation et de classification était de classer les pré-détections en trois classes : petit bateau, grand bateau et tout ce qui n'est pas un bateau. D'une part, pour réduire le nombre de fausses alarmes et d'autre part, pour caractériser les bateaux détectés.

La segmentation est basée sur l'utilisation de contours actifs. Elle constitue le traitement qui prend le plus de temps. Toutefois, ce temps pourrait être considérablement diminué en utilisant un autre langage que matlab (qui a été utilisé uniquement dans le cadre des travaux de recherches). En utilisant une plateforme de développement mieux adaptée, il serait possible d'obtenir des temps de traitement bien inférieurs à ceux de l'opérateur humain. De plus, nous avons proposé un certain nombre d'optimisations afin d'améliorer encore le temps de traitement (parallélisation de certaines tâches et utilisation de GPU).

Nous avons défini des caractéristiques de trois types : intensités, formes et textures. Puis, nous avons testé de nombreuses méthodes existantes pour finalement, sélectionner une méthode d'arbre multivarié (*functional trees*) qui donne de bons résultats.

A l'issue de cette phase, nous obtenons une probabilité de détection d'environ **97%** pour les petits bateaux et d'environ **81%** pour les grands bateaux. Les petits bateaux qui ne sont pas détectés restent des cas assez particuliers (très petits bateaux peu contrasté avec le fond). Les moins bons résultats obtenus pour les grands bateaux sont dus à un nombre d'itérations insuffisant pour la segmentation de certains bateaux, mais également à un échantillon d'apprentissage présentant assez peu d'exemples de grands navires.

Nous avons également étudié la détection des sillages qui complète la caractérisation d'un bateau en donnant une indication de cap. Nous avons proposé deux approches différentes des méthodes classiques utilisées : la première basée sur l'utilisation de contours actifs et la seconde basée sur une approche *a contrario*. Ces deux méthodes donnent des résultats satisfaisants. Une utilisation conjointe des deux méthodes améliorerait probablement encore ces résultats.

Dans l'état actuel de nos recherches, une automatisation totale de la détection des bateaux n'est pas envisageable. Toutefois, une automatisation partielle peut être envisagé dans un contexte opérationnel. Une aide à l'opérateur humain pourra être apportée en lui proposant un ensemble de pré-détections de bateaux qu'il pourra valider ou non.

Le principal problème pour une automatisation totale réside dans le nombre de fausses alarmes encore présentes à la fin du traitement. Plus de 90% de ces fausses détections sont dues à la présence de crêtes de vagues. Cet aspect constitue notre première perspective de recherche.

Perspectives

Traitement des zones de mer agitée

Nous avons constaté que certaines zones de mer présentent de nombreuses fausses détections dues à la présence de crêtes de vagues. Une étude des relations spatiales [3] entre ces différentes fausses alarmes permettrait sans doute de caractériser ces situations pour mieux les traiter. Les processus ponctuels marqués [13] semblent tout indiqués pour cela. Ainsi, nous pourrions aller vers une automatisation totale du processus de détection.

Détection des sillages

Au niveau du traitement des sillages, deux perspectives sont envisageables :

1. La première concerne l'amélioration de la détection des sillages : soit en fusionnant les informations des deux approches (contours actifs et *a contrario*), soit en cherchant un couplage efficace de ces deux méthodes. Le but étant d'obtenir une meilleure probabilité de détection des sillages.
2. La seconde perspective concerne la possibilité de détecter les sillages non rectilignes en utilisant une approche *a contrario*. Cette piste est intéressante car elle dépasse largement le cadre applicatif de cette étude.

Fusion multi-sources

Un travail de recherche devra être réalisé pour prendre en compte la possibilité d'un changement de résolution prévu dans un avenir proche avec l'arrivée sur le marché des images issues de la constellation de satellites Pléiades ou encore des satellites Formosat 2.

Il serait également intéressant de réaliser un travail visant à développer une méthodologie de fusion des données multi-sources : optiques, radars et balises VMS (*Vessel Monitoring System*).

Caractérisation des comportements des bateaux

Dans le cadre de la surveillance des pêches (cadre dans lequel s'inscrivent initialement ces travaux de recherche), une dernière perspective concerne l'étude des comportements des bateaux en activité de pêche. Il nous faudra alors utiliser la connaissance d'experts de ces comportements afin de les modéliser et de pouvoir les exploiter. Il faudra alors coupler

les informations issues de la détection des bateaux avec celles issues de la modélisation des comportements et celles plus contextuelles comme, par exemple, l'heure de prise de vue ou les horaires des marées.

Comme nous le voyons, bien qu'ayant l'impression d'avoir contribué à bien « débroussailler » le sujet, de nombreuses pistes restent à explorer.

Annexe A

Détails des calculs

A.1 Calculs pour l'algorithme GLRT-Var

dans le chapitre 2, nous avons défini :

$$L(R) = -\frac{1}{2} \sum_{s_i \in R} \left[\log(2\pi\sigma_R^2) + \frac{(s_i - m_R)^2}{\sigma_R^2} \right]$$

et

$$\log\Lambda = L(w) + L(\bar{w}) - L(A)$$

Nous obtenons :

$$\begin{aligned} \log\Lambda &= -\frac{1}{2} \sum_{s_i \in w} \left[\log(2\pi\sigma_w^2) + \frac{(s_i - m_w)^2}{\sigma_w^2} \right] \\ &\quad - \frac{1}{2} \sum_{s_i \in \bar{w}} \left[\log(2\pi\sigma_{\bar{w}}^2) + \frac{(s_i - m_{\bar{w}})^2}{\sigma_{\bar{w}}^2} \right] \\ &\quad + \frac{1}{2} \sum_{s_i \in A} \left[\log(2\pi\sigma_A^2) + \frac{(s_i - m_A)^2}{\sigma_A^2} \right] \end{aligned} \tag{A.1}$$

Après des simplifications calculatoires, nous avons :

$$\log\Lambda = -\frac{1}{2}(N_w \log(\hat{\sigma}_w^2) + N_{\bar{w}} \log(\hat{\sigma}_{\bar{w}}^2) - N_A \log(\hat{\sigma}_A^2))$$

A.2 Calculs pour l'algorithme GLRT

Nous avons pour cet algorithme, l'hypothèse supplémentaire que le fond et les cibles ont la même variance ($\sigma_A = \sigma_{\bar{w}} = \sigma_w = \sigma$).

Nous obtenons donc :

$$\begin{aligned} \log \Lambda = & -\frac{1}{2} \sum_{s_i \in w} \left[\log(2\pi\sigma^2) + \frac{(s_i - m_w)^2}{\sigma^2} \right] \\ & -\frac{1}{2} \sum_{s_i \in \bar{w}} \left[\log(2\pi\sigma^2) + \frac{(s_i - m_{\bar{w}})^2}{\sigma^2} \right] \\ & +\frac{1}{2} \sum_{s_i \in A} \left[\log(2\pi\sigma^2) + \frac{(s_i - m_A)^2}{\sigma^2} \right] \end{aligned} \quad (\text{A.2})$$

Après des simplifications calculatoires, nous avons :

$$\log \Lambda = -\frac{1}{2\sigma^2} (N_w \hat{m}_w^2 + N_{\bar{w}} \hat{m}_{\bar{w}}^2 - N_A \hat{m}_A^2) \quad (\text{A.3})$$

Nous avons donc :

$$\log \Lambda \propto N_w \hat{m}_w^2 + N_{\bar{w}} \hat{m}_{\bar{w}}^2 - N_A \hat{m}_A^2 \quad (\text{A.4})$$

Que nous simplifions en :

$$\log \Lambda = N_w \hat{m}_w^2 + N_{\bar{w}} \hat{m}_{\bar{w}}^2 - N_A \hat{m}_A^2 \quad (\text{A.5})$$

Annexe B

Choix du paramètre L_A pour GLRT-Var et GLRT-Gamma

Pour réaliser nos expérimentations, nous avons utilisé la base d'images 1 décrit dans le chapitre 1.

Nous voulons choisir une taille de fenêtre d'analyse. Pour cela, nous testons différentes tailles de fenêtre :

$$\left\{ \begin{array}{l} 7 \times 7 \text{ pixels } (L_A=7) \\ 11 \times 11 \text{ pixels } (L_A=11) \\ 15 \times 15 \text{ pixels } (L_A=15) \end{array} \right.$$

Nous construisons pour chaque algorithme (**GLRT-Var** et **GLRT-Gamma**) une courbe ROC pour chaque taille de fenêtre en faisant varier le seuil de détection. L'observation des courbes ROC nous permet de sélectionner la taille de la fenêtre la plus adaptée.

B.1 Choix de la taille de la fenêtre d'analyse pour GLRT-Var

Les différentes courbes ROC obtenues sur l'échantillon de test en utilisant l'algorithme **GLRT-Var** sont représentées dans la figure B.1.

Nous constatons que quelque soit la valeur du seuil, la fenêtre d'analyse de 15x15 pixels (courbe bleue) donne les meilleurs résultats (les valeurs PD de la fenêtre 15x15 pixels sont toujours supérieures (ou égales) aux valeurs de PD des deux autres fenêtres

quelque soit la PFA). De façon évidente, une fenêtre de 15×15 pixels ($L_A=15$) doit être utilisée, permettant, pour une PFA donnée, d'obtenir les meilleures performances possibles en détection.

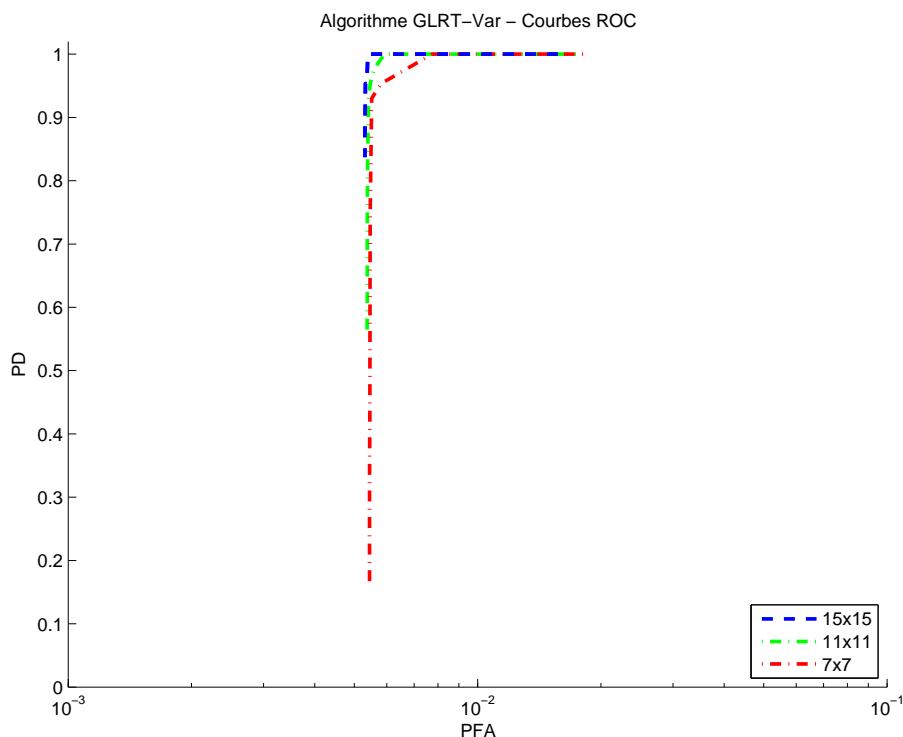


FIG. B.1 – Courbes ROC l'algorithme **GLRT-Var** et pour des tailles de fenêtre 7×7 ($L_A=7$) 11×11 ($L_A=11$) et 15×15 ($L_A=15$) pixels.

B.2 Choix de la taille de la fenêtre d'analyse pour **GLRT-Gamma**

Les différentes courbes ROC obtenues sur l'échantillon de test en utilisant l'algorithme **GLRT-Gamma** sont représentées dans la figure B.2.

La taille de fenêtre qui permet d'avoir la probabilité de détection la plus élevée avec la probabilité de fausse alarme la plus basse est, de façon évidente, une fenêtre de taille 11×11 pixels (courbe verte). Une fenêtre de 11×11 pixels ($L_A=11$) doit donc être utilisée.

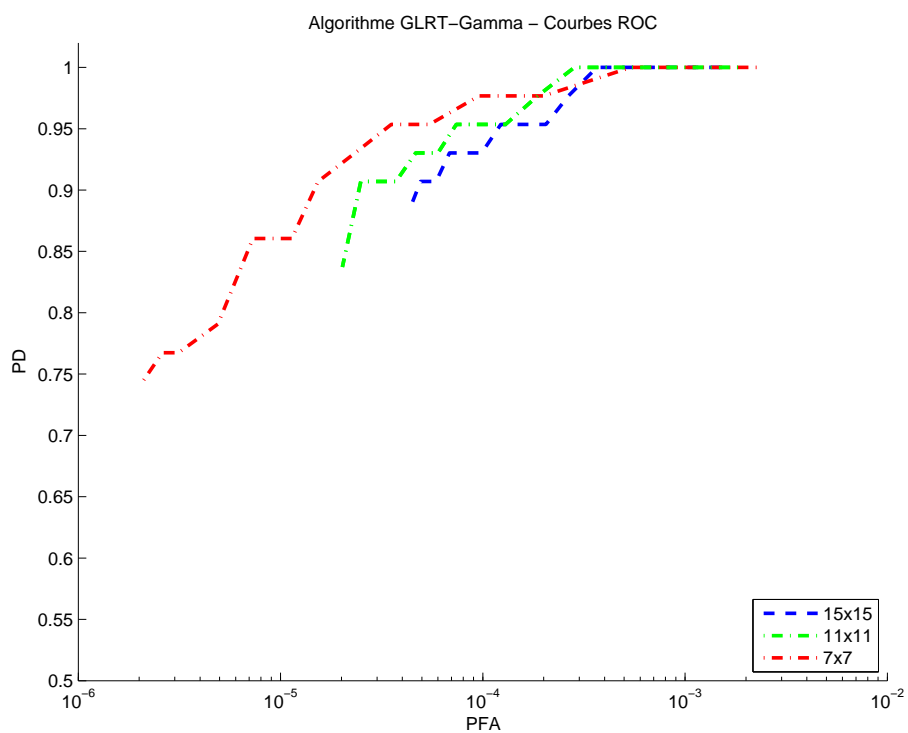


FIG. B.2 – Courbes ROC l’algorithme **GLRT-Gamma** et pour des tailles de fenêtre 7×7 ($L_A=7$) 11×11 ($L_A=11$) et 15×15 ($L_A=15$) pixels.

Annexe C

Compléments sur la détection des sillages

C.1 Influence du nombre de sillages et de θ sur r

Nous voulons maintenant déterminer la distance r « tolérable » entre un sillage et un bateau en fonction du nombre de sillages et de l'angle θ (angle entre l'orientation d'un sillage et le bateau).

Nous avons vu que le critère de décision est $\mathbf{NFA}(\mathbf{B}, \mathbf{S}) \leq \epsilon$, ce qui est équivalent à :

$$N_{\text{Couples}} \frac{\theta r^2}{N^2} \leq \epsilon \Leftrightarrow r \leq \sqrt{\frac{N^2 \cdot \epsilon}{\theta \cdot N_{\text{Couples}}}} \quad (\text{C.1})$$

où N_{Couples} est le nombre de couples bateau/sillage candidats et N est la taille de l'image.

Nous fixons une valeur pour ϵ ($\epsilon = 0, 2$) et pour θ (nous testons plusieurs valeurs pour $\theta = \{\frac{\pi}{4}, \frac{\pi}{8}, \frac{\pi}{16}, \frac{\pi}{32}\}$) puis nous faisons varier N_{couples} entre 1 et 150. Nous obtenons la courbe de la figure C.1.

Quand il y a peu de sillages, plus θ est petit, plus la distance « tolérée » entre le sillage et le bateau va être élevée. Plus le nombre de sillages est faible, plus nous avons de chances de détecter des couples dont le sillage et le bateau sont éloignés. Quand le nombre de sillages est petit, les tolérances sont très élevées ; c'est-à-dire qu'un sillage et un bateau seront couplés même si la distance est élevée ou l'angle θ grand, ce qui risque d'être gênant quand il y a peu de sillages.

Quand le nombre de sillages augmente, cette distance diminue fortement. Plus il y a

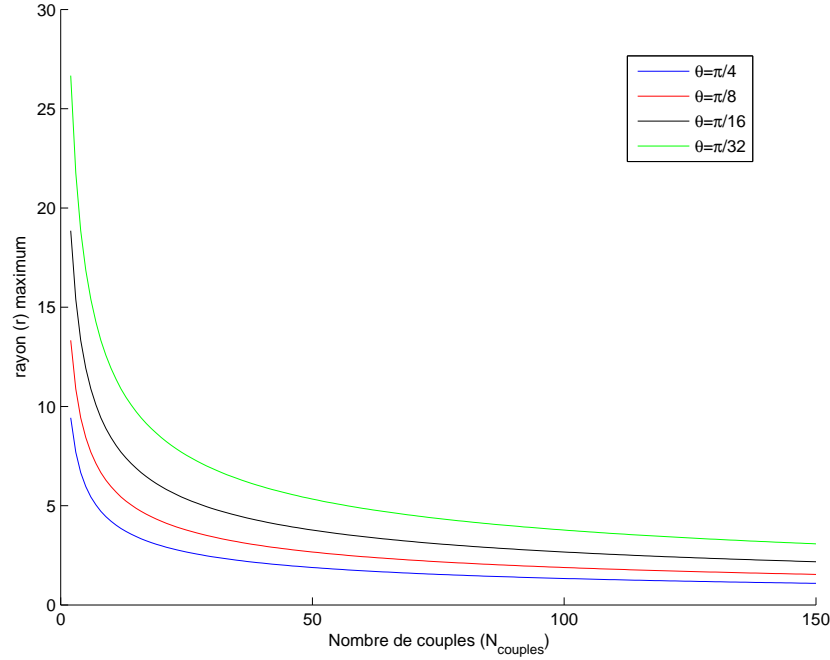


FIG. C.1 – Courbes permettant d’obtenir le rayon maximum en fonction du nombre de couples et pour $\theta = \{\frac{\pi}{4}, \frac{\pi}{8}, \frac{\pi}{16}, \frac{\pi}{32}\}$.

de sillages, plus la distance entre un sillage et un bateau devra être petite pour que le couple soit détecté.

Plus l’angle θ est petit, plus la distance entre le sillage et le bateau pourra être grande.

Plus l’angle θ sera grand, plus la distance entre le sillage et le bateau devra être petite pour que le couple soit détecté.

C.2 Influence du nombre de sillages et de r sur θ

Nous voulons maintenant déterminer l’angle θ « tolérable » en fonction du nombre de sillages candidats et de la distance entre un sillage et un bateau.

A partir de l’équation C.1 nous déduisons :

$$\theta < \frac{N^2 \cdot \epsilon}{N_{Couples} \cdot r^2} \quad (C.2)$$

où $N_{Couples}$ est le nombre de couples bateau/sillage candidats, N est la taille de l’image et r est la distance entre un sillage et le bateau.

Nous fixons ϵ ($\epsilon = 0, 2$) et r (nous testons plusieurs valeurs pour $r = 5, 10, 15, 20$), puis nous faisons varier $N_{couples}$ entre 1 et 150. Nous obtenons la courbe de la figure C.2.

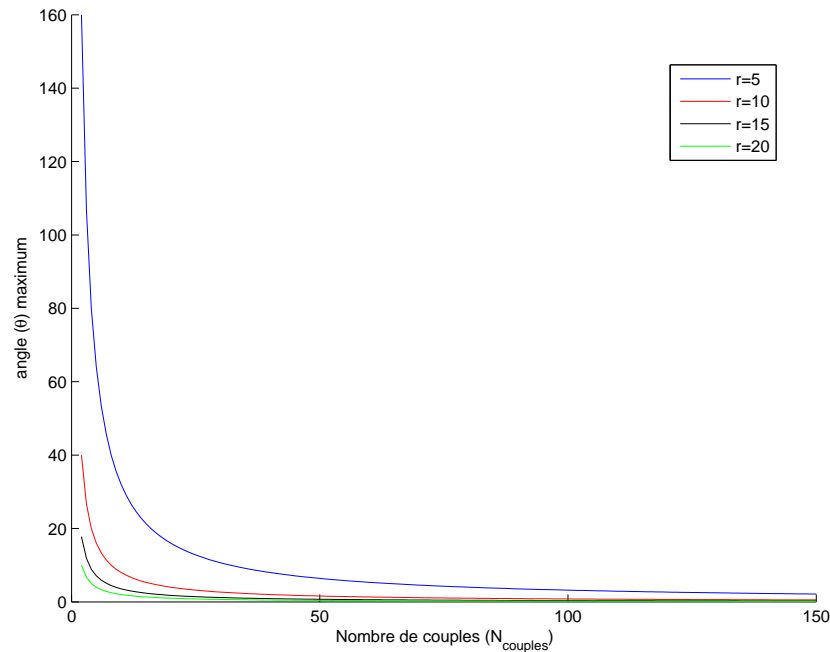


FIG. C.2 – Courbes permettant d'obtenir l'angle θ maximum en fonction du nombre de couples et pour $r = 5, 10, 15, 20$.

Quand il y a peu de sillages, l'angle θ « tolérable » est grand ; c'est-à-dire qu'un bateau situé dans n'importe quelle direction par rapport à l'orientation du sillage sera couplé avec ce sillage. Plus on va augmenter le nombre de sillages candidats, plus ce phénomène va s'atténuer.

Plus la distance r augmente plus la « tolérance » sur θ diminue. Donc plus la distance entre le sillage et le bateau sera élevée, plus l'angle du cône devra être petit pour que le couple soit détecté.

Bibliographie

- [1] J. Alves, J. Herman, and N. C. Rowe. Robust recognition of ship types from an infrared silhouette. Command and Control Research and Technology Symposium, 2004.
 - [2] A. Arnold-Bos, A. Khenchaf, and A. Martin. An evaluation of current ship wake detection algorithms en SAR images. In *Caractérisation du milieu marin*, Brest, France, October 2006.
 - [3] I. Bloch. Fuzzy spatial relationships for image processing and interpretation : A review. *Image and Vision Computing*, 23(2) :89–110, 2005.
 - [4] R. Bouckaert. Bayesian network classifiers in Weka. Technical report, 2004.
 - [5] Yi Chen and Yaqui JinDOI. Detection of ship wakes in SAR image using rotated window radon transform. *Journal of Electronics (China)*, Volume 19 -(Issue - 1) :30–36, january 2002.
 - [6] C. Chesnaud, P. Réfrégier, and V. Boulet. Statistical region snake-based segmentation adapted to different physical noise models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11) :1145–1157, 1999.
 - [7] G. Cooper and T. Dietterich. A Bayesian method for the induction of probabilistic networks from data. In *Machine Learning*, pages 309–347, 1992.
 - [8] C. Corbane, F. Marre, and M. Petit. Using SPOT-5 HRG data in panchromatic mode for operational detection of small ships in tropical area. *Sensors*, 8(5) :2959–2973, 2008.
 - [9] Ph. Courmontagne. Transformée de radon et filtrage : Application à la détection de sillages de mobiles marins. *Traitement du Signal*, Vol. 15, No. 4 :p. 297–307, 1998.
 - [10] Ph. Courmontagne. An improvement of ship wake detection based on the radon transform. *Signal Process.*, 85(8) :1634–1654, 2005.
 - [11] D.J. Crisp. The state-of-the-art in ship detection in synthetic aperture radar imagery. Technical report, Defence Science and Technology Organisation - Australia, 2004.
 - [12] Projet DECLIMS. <https://maritimeaffairs.jrc.ec.europa.eu/web/declims/home>.
-

-
- [13] X. Descombes, F. Kruggel, C. Lacoste, M. Ortner, G. Perrin, and J. Zerubia. Marked point process in image analysis : from context to geometry. In *International Conference on Spatial Point Process Modelling and its Application (SPPA)*, Castellon, Spain, 2004.
- [14] A. Desolneux, L. Moisan, and J.M. Morel. Meaningful alignments. *Int. J. Comput. Vision*, 40(1) :7–23, 2000.
- [15] A. Desolneux, L. Moisan, and J.M. Morel. Edge detection by Helmholtz principle. *JMIV*, 14(3) :271–284, May 2001.
- [16] A. Desolneux, L. Moisan, and J.M. Morel. A grouping principle and four applications. *PAMI*, 25(4) :508–513, 2003.
- [17] A. Desolneux, L. Moisan, and J.M. Morel. *Seeing, Thinking and Knowing*, chapter Gestalt Theory and Computer Vision, pages 71–101. A. Carsetti ed., Kluwer Academic Publishers, 2004.
- [18] K. Eldhuset. Principles and performance of an automated ship detection system for SAR images. *Digest - International Geoscience and Remote Sensing Symposium (IGARSS)*, 1 :358–361, 1989.
- [19] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8) :861 – 874, 2006.
- [20] Y. Freund and R. E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 1999.
- [21] J. Friedman, T. Hastie, and R. Tibshirani. Additive Logistic Regression : a Statistical View of Boosting. *The Annals of Statistics*, 38(2), 2000.
- [22] J. Gama. A linear-Bayes classifier. In *IBERAMIA-SBIA '00 : Proceedings of the International Joint Conference, 7th Ibero-American Conference on AI*, pages 269–279, London, UK, 2000.
- [23] J. Gama. Functional trees. *Mach. Learn.*, 55(3) :219–250, 2004.
- [24] A. Garzelli. Detection of ship wakes in SAR images using morphological operators. *Computers and Geosciences*, 21(10) :1201–1203, 1995.
- [25] O. Germain and P. Réfrégier. Optimal snake-based segmentation of a random luminance target on a spatially disjoint background. *Optical Letters*, 21 :1845–1847, 1996.
- [26] E. Grandchamp and P. Marthon. Driving segmentation and recognition phases using multiscale characterization. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS'03)*, pages 3441–3443, july 2003.
- [27] H. Gravelius. *Flusskunde. Göschen*, 1914.
-

-
- [28] H. Greidanus. Assessing the operationality of ship detection from space. In *European Space Agency, (Special Publication) ESA SP*, number 584, pages 31–35, 2005.
- [29] H. Greidanus. Sub-aperture behavior of SAR signatures of ships. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3579–3582, 2006.
- [30] H. Greidanus and N. Kourti. A detailed comparison between radar and optical vessel signatures. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3267–3270, 2006.
- [31] H. Greidanus and N. Kourti. Findings of the DECLIMS project - Detection and Classification of Marine Traffic from Space. In *Advances in SAR Oceanography from Envisat and ERS Missions*, volume 613 of *ESA Special Publication*, April 2006.
- [32] G. Hajduch, P. Leilde, and V. Kerbaol. Ship detection on ENVISAT ASAR data : Results, limitations and perspectives. In *Advances in SAR Oceanography from Envisat and ERS Missions*, number 613 in *ESA Special Publication*, 2006.
- [33] R.M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5) :786–804, 1979.
- [34] R.M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6) :610–621, November 1973.
- [35] R.M. Haralick, S.R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *PAMI*, 9(4) :532–550, July 1987.
- [36] M-K. Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2) :179–187, February 1962.
- [37] J.M. Kuo and K.S. Chen. Ship wake detection in synthetic aperture radar images using a combination of a wavelet correlator and radon transform. *Optical Engineering*, 41(3) :686–696, 2002. cited By (since 1996) 0.
- [38] J.M. Kuo and K.S. Chen. The application of wavelets correlator for ship wake detection in SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, 41(6) :1506–1511, June 2003.
- [39] N. Landwehr, M. Hall, and E. Frank. Logistic model trees. volume 2837, pages 241–252, 2003.
- [40] N. Landwehr, M. Hall, and E. Frank. Logistic model trees. *Machine Learning*, 59(1-2) :161–205, 2005.
- [41] Projet LIMES. <http://www.fp6-limes.eu/index.php>.
- [42] I-I. Lin and V. Khoo. Computer-based algorithm for ship detection from ERS-SAR imagery. In *Third ERS Symposium on Space at the service of our Environment*, volume 414 of *ESA Special Publication*, page 1411, 1997.
-

-
- [43] C. Liu, P.W. Vachon, and G.W. Geling. Improved ship detection with airborne polarimetric SAR data. *The Canadian Journal of Remote Sensing*, 31(1), feb 2005.
- [44] B. Lortic and M.C. Lortic. L'imagerie radar. *Villes et Citadins du Tiers-Monde.Cahiers*, pages 83–96, 1990.
- [45] A. Witkin M. Kass and D. Terzopoulos. Snakes : Active contour models. *International Journal of Computer Vision*, 1(4) :321–331, 1988.
- [46] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
- [47] S. Mallat and W.L. Hwang. Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory*, 38 :617–643, 1992.
- [48] M. Mercimek, K. Gulez, and T. V. Mumcu. Real object recognition using moment invariants. *Sadhana - Academy Proceedings in Engineering Sciences*, 30(6) :765–775, 2005.
- [49] C. J. Oliver, D. Blacknell, and R. G. White. Optimum edge detection in SAR. *Radar, Sonar and Navigation, IEE Proceedings -*, 143(1) :31–40, 1996.
- [50] V. Page, F. Goudail, and Ph. Refregier. Improved robustness to non-homogeneous background using the maximum-likelihood ratio test location algorithm. *Optics Letters*, 24 :1383–1385, 1999.
- [51] M. Petit, F. Marre, F. Marquès, F. Hynh, P. Vendeville, J. Rosé, Y. Barthélémy, F. Axes, L. Gardel, F. Baillarin, P. Calès, and P. Mérigot. IBIS : Implementation of a Boat Information System - rapport final réseau terre et espace.
- [52] J. R. Quinlan. *C4.5 : Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, 1 edition, January 1993.
- [53] M. T. Rey, J. K. Tunaley, J. T. Folinsbee, P. A. Jahans, J. A. Dixon, and M. R. vant. Application Of Radon Transform Techniques To Wake Detection In Seasat-A SAR Images. *IEEE Transactions on Geoscience and Remote Sensing*, 28 :553–560, July 1990.
- [54] J.J. Rodríguez, C. García-Osorio, J. Maudes, and J-F. Díez-Pastor. An experimental study on ensembles of functional trees. In Neamat El Gayar, Josef Kittler, and Fabio Roli, editors, *MCS*, volume 5997 of *Lecture Notes in Computer Science*, pages 64–73. Springer, 2010.
- [55] L.L. Scharf. *Statistical signal processing : detection, estimation, and time series analysis*. Addison-Wesley, 1991.
- [56] Projet SEAS-Guyane. [https ://www.seas-guyane.org/seasguyane/index.htm](https://www.seas-guyane.org/seasguyane/index.htm).
-

-
- [57] J.A. Sethian. *Level Set Methods : Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Science*. Cambridge University Press, 1996.
- [58] A.J.E. Smith, J. Chesworth, and H. Greidanus. Ship detection with envisat's alternating polarization mode. In *European Space Agency, (Special Publication) ESA SP*, number 572 in ESA Special Publication, pages 1551–1556, 2005.
- [59] M. R. Teague. Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70(8) :920–930, 1980.
- [60] M. Tello, C. Lopez-Martinez, and J. J. Mallorqui. A novel algorithm for ship detection in SAR imagery based on the wavelet transform. *IEEE Geoscience and remote sensing letters*, 2(2) :206–211, 2005.
- [61] M. Tello, C. López-Martínez, and J. J. Mallorqui. A Novel Algorithm for Automatic Ship and Oil Spill Detection Based on Time-Frequency Methods. In *Advances in SAR Oceanography from Envisat and ERS Missions*, volume 613 of *ESA Special Publication*, April 2006.
- [62] M. Tello, C. Lopez-Martinez, J.J. Mallorqui, and R. Bonastre. Automatic detection of spots and extraction of frontiers in SAR images by means of the wavelet transform : Application to ship and coastline detection. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 383–386, 2006.
- [63] M. Tello, C. López-Martínez, and J.J. Mallorqui. A novel algorithm for ship detection in SAR imagery based on the wavelet transform. *IEEE Geoscience and Remote Sensing Letters*, 2(2) :201–205, 2005.
- [64] M. Tello, J.J. Mallorqui, and C. Lopez-Martinez. Application of multiresolution and multispectral polarimetric techniques for reliable vessel monitoring and control. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, volume 1, pages 24–27, 2005.
- [65] J.K.E. Tunaley, E.H. Buller, K.H. Wu, and M.T. Rey. The simulation of the SAR image of a ship wake. *IEEE Transactions on Geoscience and Remote Sensing*, 29(1) :149–156, 1991.
- [66] R. Wang, J. Zhang, and Z. Bi. An automatic detecting method of SAR V-image ship wake. In *Proceedings of the 2005 Dragon Symposium (SP-611)*, january 2006.
- [67] R. M. Welch, S. K. Sengupta, and D. W. Chen. Cloud field classification based upon high spatial resolution textural features 1. gray level co-occurrence matrix approach. *Journal of Geophysical Research*, 93 :12663–12681, 1988.
- [68] P. J. Withagen, K. Schutte, A. M. Vossepoel, and M. G. Breuers. Automatic classification of ships from infrared (FLIR) images. In I. Kadar, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3720 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 180–187, July 1999.
-

- [69] I. H. Witten and E. Frank. *Data Mining : Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Second edition, June 2005.
 - [70] F. Zhang, B. Wu, L. Zhang, H. Huang, and Y. Tian. Illicit vessel identification in inland waters using SAR image. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3144–3147, 2006.
-

Publications

N. Proia and V. Pagé. Characterization of a Bayesian ship detection method in optical satellite images. *IEEE Geoscience and Remote Sensing Letters*, 2010.

N. Proia and V. Pagé. Méthode de détection de bateaux utilisant la théorie bayésienne dans des images satellites optiques. In *Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle (RFIA)*, CAEN, janvier 2010.

N. Proia, V. Pagé, and Jubelin G. Maritime surveillance with the use of optical satellite images. In *Observation des Côtes et des Océans : Senseurs et Systèmes, OCOSS 2010*, Brest, juin 2010.

N. Proia and V. Pagé. Caractérisation d'une méthode Bayésienne de détection de bateaux dans des images satellites optiques. In *ORASIS'09 - Congrès des jeunes chercheurs en vision par ordinateur*, Trégastel, France, juin 2009

Maritime surveillance using panchromatic optical satellite images

Abstract

The research presented in this thesis aims at determining the possibility of partial or total automatic ship detection in High Resolution optical satellite images. At first, we expose the industrial context of our study, which fixes the constraints in terms of performances and processing time. Then, we present the available satellite data for our study (SPOT 5 panchromatic images of 5m per pixel resolution). Then we detail our approach for ship detection : the first stage is a pre-detection of targets that gives us candidates. The second stage is a precise segmentation of each candidate and the third stage is a classification of candidates.

Our pre-detection method is based on the Bayesian decision theory. This pre-detection supplies a set of regions of interests (ROI). Among them, we have ships, but also false alarms due to clouds and to crests of waves. The purpose of the following stage is to eliminate the false alarms and to characterize the pre-detected ships.

Then, we detail the method used to segment candidates. It uses active contours and also uses the Bayesian decision theory. After a brief description of the characteristics extracted for the classification of ships, we present the classifier that we used.

We complete the ship characterization by presenting two methods of wave ship detection. The first approach uses active contours and the second is based on an *a contrario* approach. We compare the results of these two approaches.

To complete the study and replace it in the industrial and operational context, we present experiments of the complete algorithm on complete images to estimate the performances and the processing time of the whole method.

Keywords : maritime surveillance, ship détection, Bayes decision theory, active contours, *a contrario* approach.

Surveillance maritime par analyse d'images satellitaires optiques panchromatiques

Résumé

Les travaux présentés dans cette thèse ont pour but de déterminer la possibilité d'automatiser, partiellement ou totalement, la détection de bateaux à partir d'images satellitaires optiques haute résolution. Nous exposons tout d'abord le contexte industriel dans laquelle s'insère notre étude, qui fixe les contraintes en termes de performances et de temps de traitement. Puis, nous présentons les données satellitaires disponibles dans le cadre de notre étude (images SPOT 5 panchromatique de résolution 5m). Après avoir illustré le contenu des scènes, nous présentons la démarche adoptée pour la détection des navires : une phase de pré-détection suivie d'une étape de segmentation et d'une étape de classification.

La méthode de pré-détection utilisée s'appuie sur la théorie bayésienne de la décision. Après avoir présenté le modèle classique, nous discutons du modèle que nous avons utilisé. Cette pré-détection fournit un ensemble de régions d'intérêts. Parmi elles, nous n'avons pas uniquement des bateaux, nous avons également des fausses alarmes dues aux nuages et aux crêtes de vagues. Le but de l'étape suivante est à la fois d'éliminer les fausses alarmes et de caractériser les bateaux pré-détectés.

Ensuite, nous détaillons la méthode utilisée pour segmenter les bateaux. Elle s'appuie sur des contours actifs et utilise également la théorie bayésienne de la décision. Après avoir décrit les caractéristiques extraites pour la classification des bateaux, nous présentons le classifieur utilisé.

Nous complétons la caractérisation des navires en présentant deux méthodes de détection des sillages différentes des méthodes classiques. La première approche s'appuie sur l'utilisation de contours actifs et la seconde est basée sur une approche *a contrario*. Nous comparons les résultats des ces deux approches.

Afin de compléter l'étude et se replacer dans le contexte industriel et opérationnel de la thèse, nous présentons des expérimentations de l'ensemble de l'algorithme sur des images complètes pour évaluer les performances et le temps de traitement.

Mots clés : surveillance maritime, détection de bateaux, théorie bayésienne de la décision, contours actifs, approche *a contrario*.