



HAL
open science

Détection automatique de cibles dans des fonds complexes. Pour des images ou séquences d'images

Solenne Thivin

► **To cite this version:**

Solenne Thivin. Détection automatique de cibles dans des fonds complexes. Pour des images ou séquences d'images. Statistiques [math.ST]. Université Paris-Saclay, 2015. Français. NNT : 2015SACLS235 . tel-01296615

HAL Id: tel-01296615

<https://theses.hal.science/tel-01296615>

Submitted on 1 Apr 2016

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.

NNT : 2015SACLS235

THÈSE DE DOCTORAT
DE
L'UNIVERSITÉ PARIS-SACLAY
PRÉPARÉE À
L'UNIVERSITÉ PARIS-SUD XI

ECOLE DOCTORALE N° 142
Ecole doctorale de mathématiques d'Orsay
Spécialité de doctorat : Mathématiques appliquées

Par

Madame Solenne THIVIN

Détection automatique de cibles dans des fonds complexes
Pour des images ou séquences d'images

Thèse présentée et soutenue à Orsay le 16 décembre 2015:

Composition du Jury:

Monsieur Pascal Massart	Professeur à l'université d'Orsay	Président
Madame Julie Delon	Université Paris Descartes	Rapporteur
Madame Mathilde Mougeot	Université Paris Diderot	Examineur
Monsieur Erwan Le Pennec	Professeur à l'école Polytechnique	Directeur de thèse
Monsieur Michel Prenat	Thales Optronique	Encadrant industriel

Résumé

L'objectif principal de ces travaux de thèse a été la mise en place d'un algorithme de détection de cibles sous-résolues pour des images infra-rouges de ciel. Pour cela, nous avons d'abord cherché à modéliser les images réelles dont nous disposons. Après une étude de ces images, nous avons proposé plusieurs modèles gaussiens prenant en compte la covariance spatiale. Dans ces modèles, nous avons supposé que les images pouvaient être segmentées en zones stationnaires. Dans chaque zone, nous avons supposé une structure forte sur la matrice de covariance (comme les modèles auto-régressifs en deux dimensions par exemple).

Il a ensuite fallu choisir entre ces modèles. Pour cela, nous avons appliqué une méthode de sélection de modèles par critère de vraisemblance pénalisée introduite par Birgé et Massart. Nous avons obtenu comme résultats théoriques une inégalité oracle qui a permis de démontrer les propriétés statistiques du modèle choisi.

Une fois le modèle sélectionné, nous avons pu bâtir un test de détection. Nous nous sommes inspirés de la théorie de Neyman-Pearson et du test du rapport de vraisemblance généralisé. Notre contrainte principale a été le respect du taux de fausses alarmes par image. Pour le garantir, nous avons appris le comportement du test sur les images réelles pour en déduire le seuil à appliquer.

Nous avons ensuite remarqué que le comportement de ce test variait fortement selon la texture de l'image : image de ciel bleu uniforme, image de nuage très texturé, etc. Après avoir caractérisé les différentes textures rencontrées avec les coefficients de scattering de Stéphane Mallat, nous avons décidé de classer ces textures. Le seuil appliqué lors de la détection a alors été adapté à la texture locale du fond.

Nous avons finalement mesuré les performances de cet algorithme sur des images réelles et nous les avons comparées à d'autres méthodes de détection.

Mots-clés : Détection, Covariance spatiale, Sélection de modèles, Apprentissage, Classification non supervisée.

Remerciements

Mes premiers remerciements vont à mes encadrants : Michel Prenat et Erwan Le Pennec. Michel a été le premier à me parler de ce beau projet. Je le remercie pour sa gentillesse, sa disponibilité et surtout sa curiosité scientifique qui m'ont poussée à consolider mes explications. Merci également à lui pour son soutien dans le monde, nouveau pour moi, de l'entreprise. Je tiens à remercier Erwan pour son optimisme, ses réponses et ses explications qui ont été comme une bouée de secours à chacune de mes questions. Sa connaissance m'a paru sans limite pendant ces trois années de thèse.

Je remercie Julie Delon et Jérémie Bigot d'avoir accepté de rapporter ma thèse. Je les remercie pour leur lecture minutieuse et leurs remarques pertinentes.

Je remercie aussi Mathilde Mougeot et Pascal Massart d'avoir accepté d'être dans mon jury de thèse. Leur enthousiasme à parler et à enseigner les mathématiques me rendent très fière de pouvoir les compter dans mon jury.

Lors de ma thèse CIFRE, j'ai passé la moitié de mon temps au sein de Thales Optronique et l'autre moitié au sein du laboratoire de mathématiques d'Orsay. J'ai donc eu la chance de rencontrer deux fois plus de personnes avec qui partager mes travaux ou simplement des moments de détente. Côté TOSA, je tiens à remercier Laurence Gasnot pour son sourire et surtout son aide dans les démarches administratives. Je souhaite également adresser un grand merci à l'équipe STI et en particulier à Guillaume, Edouard, Grégoire et Emilie dont les explications, les réponses ou simplement l'écoute m'ont aidée tant de fois à prendre du recul sur mes travaux. Des liens plus que professionnels ne se sont tissés avec beaucoup d'entre eux. Côté Orsay, je tiens à remercier les enseignants/chercheurs que j'ai croisés durant mes études à Orsay. Un énorme merci à Valérie Blandin-Lavigne pour tous les soucis administratifs qu'elles ont su régler. Merci aussi à Nathalie Carrière qui est devenue une vraie amie. Un merci particulier aux doctorants que j'ai croisés plus ou moins souvent durant ces trois années : Vincent, Pierre-Antoine, Elodie, Thomas, Un merci tout spécial et plein de tendresse aux *GIRLS* : Emilie, Valérie, Céline et Mélina pour les pauses, les larmes, les fou-rires... Quelles belles années passées avec vous ! Je vous souhaite le meilleur !!!

Merci également à ceux plus éloignés du monde des mathématiques qui m'ont permis de décrocher un peu de mes travaux : Margaux, Alice, Yohann, Pierre, Sébastien (mieux connus sous le pseudonyme des *SYMPA*) ou à Maud-Louise, Julie, Elsa et à toutes les autres nageuses du Stade Français.

Je profite de ces quelques lignes pour remercier ma famille pour qui cette thèse restera sûrement un peu floue mais dont la présence et le soutien sont, pour toujours, une source de motivation et de joie de vivre. Merci donc à Christiane, Claude, Noël, Denis, Sylvie, Claude, Isabelle, Nadine, Serge, Sylvie, Marc, Cyril, Rémi, Priscilla, Loïc, Xavier, Amélie, Flavie, Laura, Benoît et Léna. De très beaux souvenirs me reviennent tandis que j'écris vos prénoms.

Un grand merci également à mon autre famille (*belle-famille* depuis peu) pour l'intérêt qu'ils ont porté à mes travaux et à mon parcours durant ces trois années. Merci à Béatrice, Serge, Christophe, Odette, Isabelle, Marie, Clément, Christel, Louis et Augustin.

Merci à mes parents, Annie et Guy. Merci à ma maman pour avoir supporté les longues conversations de mathématiques durant les repas d'hiver et pour son soutien inébranlable. Merci à mon

papa pour nos échanges mathématiques et ses tentatives de lecture à chacun de mes écrits. Leur confiance en ma réussite aussi bien professionnelle que personnelle ont fait de moi la personne que je suis aujourd'hui. Merci pour tout ça et tout le reste!

Merci également à mon frère Clément et à Eléonore. Je leur souhaite une belle entrée dans la vie active, dans l'industrie ou dans la recherche!

Enfin, merci à *Pierre* pour sa présence et son soutien irremplaçables.

A vous tous, je dédie cette thèse.

Table des matières

Résumé	3
Remerciements	5
Notations	13
Introduction	21
1 Théorie de la détection	29
1.1 Tests et stratégies	30
1.1.1 Rappels sur les tests	30
1.1.2 Différentes stratégies	32
1.1.3 Dans un cadre de détection	33
1.2 Test du rapport de vraisemblance pour la détection	35
1.2.1 Expression du test du rapport de vraisemblance	35
1.2.2 Test du rapport de vraisemblance généralisé	37
1.3 Mesures des performances d'un test de détection	39
1.3.1 Probabilité de fausses alarmes	39
1.3.2 Probabilité de détection	40
1.3.3 Courbes COR	41
1.3.4 Estimation des grandeurs	42
1.3.5 Echantillon d'images réelles	46
2 Modèles pour les textures	51
2.1 Premières modélisations	52
2.1.1 Estimation du vecteur moyenne	52
2.1.2 Hypothèses sur la matrice de covariance	55
2.2 Modèles indirects	56
2.2.1 Modèles auto-régressifs simultanés	56
2.2.2 Modèles auto-régressifs conditionnels	59
2.2.3 Intérêts des modèles SAR et CAR	60
2.2.4 Estimation des paramètres par maximum de vraisemblance	63
2.3 Modèles directs	64
2.3.1 Exemples de modèles directs	64
2.3.2 Estimation du maximum de vraisemblance sous contraintes	65
2.4 Modèles stationnaires par morceaux	67
2.4.1 Modèles SAR non stationnaires	68
2.4.2 Définition des modèles gaussiens finis stationnaires par morceaux	68
2.4.3 Estimation des paramètres par maximum de vraisemblance	71
3 Sélection de modèles	75
3.1 Cadre général	76
3.1.1 Modèles oracles	76
3.1.2 Sélection pratique	77
3.1.3 Comparaison des estimateurs	77

3.2	Reformulation des hypothèses	79
3.2.1	Autre expression de l'hypothèse ($H_{\mathcal{M}(\mathcal{P},\mathcal{S})}$)	79
3.2.2	Autre expression des hypothèses (H_m)	80
3.2.3	Inégalité oracle pour les modèles gaussiens stationnaires par morceaux	81
3.3	Applications aux images réelles	81
3.3.1	Collections particulières de partitions	81
3.3.2	Ensemble \mathcal{S} particulier	84
3.3.3	Parcours des ensembles $\mathcal{M}(\mathcal{P}_1, \mathcal{S}_1)$ et $\mathcal{M}(\mathcal{P}_2, \mathcal{S}_1)$	84
3.3.4	Réglage de la pénalité	87
3.3.5	Exemples sur images simulées	87
3.3.6	Exemples sur images réelles	90
3.4	La pseudo log-vraisemblance	91
3.4.1	Définition	91
3.4.2	Exemples sur images simulées puis réelles	92
4	Algorithme de détection	97
4.1	Filtrage proposé	98
4.1.1	Etape 1 : Retrait de la moyenne	98
4.1.2	Etape 2 : Estimation de la matrice de covariance	100
4.1.3	Etape 3 : Retrait de la structure de covariance	101
4.1.4	Définition de la statistique de test	102
4.2	Apprentissage des seuils	104
4.2.1	Etape hors-ligne d'estimation de quantiles	104
4.2.2	Performances de l'étape hors-ligne d'estimation	106
4.3	Prise en compte de la texture locale	108
4.3.1	Descripteurs de Portilla/Simoncelli	108
4.3.2	Coefficients de scattering de Mallat	111
4.3.3	Mélanges de lois gaussiennes	112
4.3.4	Apprentissage par classe	113
4.4	Performances de l'algorithme général	117
4.4.1	Algorithme antérieur	117
4.4.2	Courbes COR pour un rapport signal sur bruit fixé	118
4.4.3	Courbes COR à probabilité de fausses alarmes fixée	121
4.4.4	Images réelles	126
	Conclusion	131
	Perspectives	133
A	Idée de preuve pour l'algorithme d'estimation sous contraintes	137
A.1	Calcul de la différentielle	137
A.2	Recherche du maximum de vraisemblance	137
B	Démonstration du théorème 2	139
B.1	Hypothèse ($H_{\mathcal{M}(\mathcal{P},\mathcal{S})}$)	139
B.2	Démonstration de la proposition 2	139
B.2.1	$[t_\theta^-, t_\theta^+]$ crochet de taille ε	139
B.2.2	$[t_\theta^-, t_\theta^+]$ contient f_θ	140
B.2.3	Démonstration Lemme 1	143
C	Autres démonstrations du chapitre 3	146
C.1	Démonstration de la proposition 4	146
C.2	Expression de la pseudo-vraisemblance pour un modèle SAR aux 4 ppv	147

D Courbes COR pour un rapport signal sur bruit fixé	149
D.1 Classe 1	149
D.2 Classe 2	150
D.3 Classe 3	151
D.4 Classe 5	152
D.5 Classe 7	153
D.6 Classe 8	154
Bibliographie	157
Index	163

Notations

I	Image réelle
M	Taille de l'image I
$\mathbf{I} = (I_1, \dots, I_n)$	Echantillon d'images réelles

Test de détection

$H_0^{i,j}$	Hypothèse nulle
$H_1^{i,j}$	Hypothèse alternative
T	Test de détection de $H_0^{i,j}$ contre $H_1^{i,j}$
T_{TLP}	Test proposé à l'issue de nos travaux
$\Delta(I, i, j), \Delta(\mathbf{I}, i, j)$	Statistique de test de T pour l'image I ou l'échantillon \mathbf{I}
s	Seuil du test
$\alpha(T)$	Taux d'erreur de première espèce du test T
$\beta(T)$	Taux d'erreur de seconde espèce du test T
α^*	Probabilité de fausses alarmes souhaitée (niveau)
q_{α^*}	Quantile de la variable $\Delta(\mathbf{X}, i, j)$ pour le niveau α^*
$q_{\alpha^*}^l$	Quantile de la variable $\Delta(\mathbf{X}, i, j)$ pour la classe l et pour le niveau α^*
$\text{pfa}(T, s)$	Probabilité de fausses alarmes du test T pour un seuil s
c	Intensité de la cible
$\text{pd}(T, s, c)$	Probabilité de détection du test T pour une intensité c et un seuil s

Modélisation des textures

$\mathbf{B} \sim N(\mathbf{m}_0, \Sigma)$	Champ spatial modélisant le fond de ciel
\mathbf{X}	Champ spatial modélisant les images I
\mathbf{Y}	Champ spatial \mathbf{X} centré
$f_{\mathbf{X}}(\cdot H_k)$	Densité du champ \mathbf{X} sous l'hypothèse H_k
$f_{\mathbf{X}}(\mathbf{I} H_k)$	Vraisemblance de l'échantillon \mathbf{I} sous l'hypothèse H_k
R	Zone de la grille $\llbracket 1, M \rrbracket^2$
$\#(R)$	Nombre de pixels de la zone R
\mathbf{X}_R	Restriction du champ \mathbf{X} à la zone R
S	Modèle stationnaire sur \mathbb{Z}^2
$\dim(S)$	Dimension du modèle S
\mathcal{S}	Collection de modèles stationnaires sur \mathbb{Z}^2
$S(R)$	Restriction du modèle S à la zone R
$\mathbf{R} = (R_1, \dots, R_P)$	Partition de la grille $\llbracket 1, M \rrbracket^2$
$ \mathbf{R} $	Nombre de zones dans la partition \mathbf{R}
\mathcal{P}	Collection de partitions de la grille $\llbracket 1, M \rrbracket^2$
$m = (R_1, \dots, R_P, S_1, \dots, S_P)$	Modèle stationnaire par morceaux sur la grille $\llbracket 1, M \rrbracket^2$
$\dim(m)$	Dimension du modèle stationnaire par morceaux m
θ_m	Paramètres du modèle m
$\hat{\theta}_m$	Estimateur du maximum de vraisemblance des paramètres du modèle m
$\mathcal{M}(\mathcal{P}, \mathcal{S})$	Collection de modèles stationnaires par morceaux définis à partir des partitions de \mathcal{P} et des modèles stationnaires de \mathcal{S}

Sélection de modèles

$KL(f, g)$	Divergence de Kullback-Liebler entre les densités f et g
$h^2(f, g)$	Distance de Hellinger entre les densités f et g
\mathcal{M}	Collection de modèles
$(H_{\mathcal{M}})$	Hypothèses portant sur le modèle \mathcal{M}
m	Modèle de la collection \mathcal{M}
(ρ_m, σ_m)	Grandeurs liées au modèle m
$(H_m), (G_m)$	Hypothèse portant sur le modèle m
$H_{\square}(m, \epsilon)$	Entropie à crochet de taille ϵ du modèle m
m^*	Modèle oracle
\mathbf{x}	Réalisation du champ spatial \mathbf{X}
$lv(\mathbf{x})$	Log-vraisemblance de l'observation \mathbf{x}
$plv(\mathbf{x})$	Pseudo log-vraisemblance de l'observation \mathbf{x}
\hat{s}_m	Estimateur du maximum de vraisemblance de la densité du modèle m
$kl_n(x, \hat{s}_m)$	Distance de Kullback-Lieber empirique pour l'observation x
$\gamma_n(x, \hat{s}_m)$	Risque empirique pour l'observation x
crit	Critère de sélection de modèles
pen	Fonction pénalité
λ	Coefficient de proportionnalité entre la pénalité et la dimension des modèles m
\hat{m}	Modèle sélectionné pour le critère crit pénalisé

Filtrage proposé

$\widehat{\mathbf{m}\mathbf{o}}$	Estimation de $\mathbf{m}\mathbf{o}$ par filtrage laplacien
N	Taille des patches découpés depuis I
N_p	Nombre de patches dans l'image I
P_1, \dots, P_{N_p}	Patches découpés depuis I
Σ_k	Matrice de covariance du patch P_k
(a_k, b_k, σ_k)	Paramètres de la matrice Σ_k
Z_k	Décomposition de Cholesky de la matrice Σ_k

Apprentissage

N_A	Taille de l'échantillon d'apprentissage
P_1, \dots, P_{N_A}	Patches de l'échantillon d'apprentissage
\hat{q}_{α^*}	Quantile empirique de l'échantillon d'apprentissage pour le niveau α^*
$\hat{\alpha}_i$	Taux de l'échantillon de test pour une pfa souhaitée α_i^*

Classification

\mathbf{v}_i	Vecteur des descripteurs calculés sur le patch P_i de l'échantillon d'apprentissage
\mathbf{V}	Variable aléatoire modélisant le comportement de \mathbf{v}_i
K	Nombre de classes du modèle de mélange gaussien
C_1, \dots, C_K	Classes du modèle de mélange gaussien
$\phi(\cdot \mathbf{m}\mathbf{o}, \Sigma)$	Densité du modèle gaussien de moyenne $\mathbf{m}\mathbf{o}$ et de matrice de covariance Σ
$\theta = (\pi_k, m_k, \Sigma_k)_{1 \leq k \leq K}$	Paramètres du mélange de lois gaussiennes
$\hat{\theta}_{EMV}^k$	Estimateur par l'algorithme EM sous l'hypothèse d'un modèle à k classes
\hat{K}_{ICL}	Estimateur du nombre de classes par le critère ICL

Mesures de performances

r	Rapport signal sur bruit (RSB)
$\widehat{\text{pfa}}_n(I, T, s)$	Taux de fausses alarmes de l'échantillon \mathbf{I}
$\widehat{\text{pd}}_n(I, T, s, r)$	Taux de détection de l'échantillon \mathbf{I} pour un rapport signal sur bruit r

Acronymes

AIC	Akaike Information Criterium
ACP	Analyse en composantes principales
ARMA	Auto-Regressif Moving Average
CAR	Conditional Auto-Regressif
CART	Classification And Regression Tree
CEM	Classification Expectation-Maximization
COR	Caractéristiques Opérationnelles de Réception
EM	Expectation-Maximisation
ICL	Integrated Completed Likelihood
KL	Kullback-Liebler
MIXMOD	MIXture MODelling
NP	Neyman-Pearson
PLV	Pseudo Log-Vraisemblance
PVC	Pseudo-Vraisemblance Conditionnelle
RSB	Rapport Signal sur Bruit
SAR	Simultaneous Auto-Regressif
TFAC	Taux de Fausses Alarmes Constant
TRV	Test du Rapport de Vraisemblance
TRVG	Test du Rapport de Vraisemblance Généralisé

Table des figures

1	Intérêt d'un apprentissage par texture	24
2	Schéma final de l'algorithme	25
1.1	Transformation de l'observation matricielle en observation vectorielle	34
1.2	Loi de la statistique de test sous chacune des hypothèses	40
1.3	Exemple de courbe COR à intensité fixée	41
1.4	Comparaison des courbes COR de deux tests	42
1.5	Exemple de courbe COR à pfa fixée à 10^{-4}	42
1.6	Tracé du taux de fausses alarmes en fonction de la probabilité de fausses alarmes	44
1.7	Exemples de signatures de cibles	44
1.8	Exemples de courbes COR empiriques pour différents rapports signal sur bruit	45
1.9	Courbe COR pour une probabilité de fausses alarmes fixée	46
1.10	Exemples d'images réelles de TOSA	47
2.1	Exemple de modélisation de mo par une fonction linéaire des coordonnées : (a) image réelle, (b) vecteur mo estimé, (c) histogramme des résidus et (d) Corrélacion empirique des résidus	53
2.2	Exemple de modélisation de mo par une fonction linéaire des coordonnées : (a) image réelle, (b) vecteur mo estimé, (c) histogramme des résidus et (d) Corrélacion empirique des résidus	54
2.3	Exemple de modélisation de mo par une fonction linéaire des coordonnées : (a) image réelle, (b) vecteur mo estimé, (c) histogramme des résidus et (d) Corrélacion empirique des résidus	55
2.4	Choix de voisinage pour des champs de taille finie	57
2.5	Simulations de champs SAR pour différents bruits	58
2.6	Exemples de réalisations de modèles SAR isotropes : à gauche avec $a = 0$ et à droite avec $a = 0.249$	58
2.7	Exemples de réalisations de modèles SAR stationnaire directionnels	59
2.8	Voisinages des 4 et des 12 plus proches voisins	60
2.9	Comparaison des estimateurs du maximum de vraisemblance et des moindres carrés ordinaires pour $a = 0.1$ et $a = 0.249$	64
2.10	Estimation des paramètres σ^2 , γ et α aux différentes itérations.	67
2.11	Modèle SAR fini stationnaire par morceaux et sa matrice de covariance	69
2.12	Réalisations de modèles SAR finis et stationnaires par morceaux	70
3.1	Deux exemples de partitions récursives dyadiques	82
3.2	Arbres dyadiques et codes associés pour les partitions de la figure 3.1	82
3.3	Deux exemples de partitions récursives split	83
3.4	Deux exemples de partitions de la collection \mathcal{P}_1	85
3.5	Exemple de fusion possible	86
3.6	Modèles SAR par morceaux simulés avec 1 zone corrélée et une zone non corrélée : (a) réalisation d'un tel modèle, (b) estimation obtenue après sélection de modèle	88
3.7	Modèle SAR par morceaux avec 2 zones fortement corrélées : (a) réalisation d'un tel modèle, (b) estimation obtenue après sélection de modèle	88
3.8	Segmentation obtenue avec l'utilisation des partitions récursives dyadiques	89

3.9	Exemples de sélection de partition pour des images réelles.	90
3.10	Modèle estimé par le critère de pseudo log-vraisemblance pénalisée et par le critère de log-vraisemblance pour l'image de la figure 3.8a	93
3.11	Sélection de modèles par pseudo log-vraisemblance pour la colonne de gauche et par log-vraisemblance sur la colonne de droite.	94
4.1	Schéma de l'algorithme de détection	98
4.2	Exemple de retrait du vecteur moyenne pour une image de ciel bleu sans nuages.	99
4.3	Exemple de retrait du vecteur moyenne pour une image de ciel bleu avec nuages.	100
4.4	Présentation des étapes de filtrage proposé dans le manuscrit	102
4.5	Exemple de retrait de la moyenne et de la structure de covariance sur une image de ciel sans nuage.	102
4.6	Exemple de retrait de la moyenne et de la structure de covariance sur une image de ciel avec nuages.	103
4.7	Schéma de l'algorithme avec la phase hors-ligne.	104
4.8	Exemples d'images réelles de l'échantillon « d'apprentissage »	105
4.9	Seuils estimés à partir de l'échantillon d'apprentissage.	106
4.10	Exemples d'images réelles de l'échantillon de test	106
4.11	Taux de fausses alarmes empiriques de l'échantillon de test	107
4.12	Exemple d'une image réelle et la répartition des fausses alarmes	107
4.13	Exemple d'une réalisation $N(0,Id)$ et de la répartition des fausses alarmes associée	107
4.14	Graphiques de la répartition des fausses alarmes pour les deux approches envisagées	108
4.15	Schéma de l'algorithme avec les deux phases hors-ligne	109
4.16	Diagramme de la « Pyramide orientable » proposée par Simoncelli et Freeman	109
4.17	Exemples d'images réelles et d'images synthétisées avec les mêmes valeurs pour les descripteurs de Simoncelli/Portilla	110
4.18	Schéma de la décomposition proposée par Mallat	111
4.19	Tableau récapitulatif des effectifs dans chacune des classes de l'échantillon d'apprentissage	113
4.20	Classification obtenue pour les patchs issus d'une image de la base d'apprentissage	114
4.21	Classification obtenue pour les patchs issus d'une image de la base d'apprentissage	114
4.22	Seuils estimés dans chacune des classes à partir de l'échantillon d'apprentissage	115
4.23	Tableau récapitulatif des effectifs dans chacune des classes de l'échantillon de test	115
4.24	Taux de fausses alarmes empiriques par classe pour l'échantillon de test	116
4.25	Taux de fausses alarmes empiriques avec l'étape de classification pour l'échantillon de test	116
4.26	Présentation des étapes de l'algorithme antérieur	118
4.27	Courbes COR pour des valeurs de RSB fixées à 4,8 et 10.	120
4.28	Courbes COR pour un RSB de 5 pour les classes 4,5,6,7, et 8.	122
4.29	Courbes COR pour les cinq probabilités de fausses alarmes étudiées	123
4.30	Courbes COR pour la classe 4 pour les 5 probabilités de fausses alarmes étudiées.	124
4.31	Courbes COR pour la classe 6 pour les 5 probabilités de fausses alarmes étudiées.	125
4.32	Schéma de la chaîne de détection intégrant un choix pour l'algorithme utilisé	126
4.33	Exemple d'image réelle et zoom sur la cible	126
4.34	Bruit après le filtre puis après le seuillage pour le patch contenant la cible	127
D.1	Courbes COR pour la classe 1 pour les 5 probabilités de fausses alarmes étudiées.	149
D.2	Courbes COR pour la classe 2 pour les 5 probabilités de fausses alarmes étudiées.	150
D.3	Courbes COR pour la classe 3 pour les 5 probabilités de fausses alarmes étudiées.	151
D.4	Courbes COR pour la classe 5 pour les 5 probabilités de fausses alarmes étudiées.	152
D.5	Courbes COR pour la classe 7 pour les 5 probabilités de fausses alarmes étudiées.	153
D.6	Courbes COR pour la classe 8 pour les 5 probabilités de fausses alarmes étudiées.	154

Introduction

Motivations

Le problème général posé lors de cette thèse est le suivant : nous souhaitons détecter dans des images réelles des cibles de petite taille. Actuellement, Thales Optronique (TOSA) dispose d'un algorithme de détection ayant de bonnes performances de détection sur fond de ciel uniforme. Nous souhaitons proposer un algorithme qui permette d'améliorer ces performances lorsque le fond est plus complexe, en présence de nuages par exemple.

Les images réelles sur lesquelles nous avons travaillé sont des images de ciel en infra-rouges acquises par TOSA à l'aide de systèmes passifs comme des capteurs infrarouges. Elles sont de différentes natures : ciel bleu uniforme, ciel bleu et nuages, nuages épais, nuages directionnels, etc. Dans le manuscrit, nous noterons \mathbf{B} la variable aléatoire modélisant le fond.

Nous souhaitons détecter les cibles au plus tôt, c'est-à-dire quand elles sont encore loin du capteur et que leur taille ne dépasse pas celle d'un pixel.

La contrainte principale lors de ces travaux a été d'effectuer cette détection à taux de fausses alarmes constant (TFAC), c'est-à-dire de pouvoir contrôler le nombre de fois où le test se trompe.

Pour des images de ciel bleu uniforme, les travaux de thèse d'Emilie Vasquez [Vas11] ont donné des résultats très satisfaisants et nous nous en sommes d'ailleurs inspirés pour proposer un nouvel algorithme. Le modèle qu'elle propose fait l'hypothèse que les pixels sont des réalisations gaussiennes indépendantes. Cette hypothèse d'indépendance est tout à fait acceptable dans le cas d'images de ciel bleu uniforme. En revanche, pour des images de nuages, la covariance spatiale entre les pixels est assez forte et on ne peut donc pas la supposer nulle. L'objectif principal a donc été de prendre en compte cette covariance spatiale pour proposer un nouveau modèle statistique pour le contenu de ces images. Une fois le modèle fixé, nous avons pu mettre en place un test de détection basé sur ce modèle et inspiré du test du rapport de vraisemblance. Dans ce test, une statistique calculée sur une image I est comparée à un seuil s . C'est le comportement de cette statistique en l'absence de cibles qui va mener au réglage du seuil s . La compréhension de ce comportement et le réglage du seuil ont été des axes principaux lors de nos travaux.

Dans la suite du manuscrit, chaque image I est vue comme une réalisation d'un champ spatial \mathbf{X} sur une grille S . En l'absence de cibles, la loi de \mathbf{X} est la même que celle de \mathbf{B} , le modèle pour le fond.

Modèles

La connaissance de la densité de la loi de probabilité f du champ \mathbf{B} en l'absence de cibles nous permettrait de mettre en place un test de détection. Cette loi étant cependant inconnue, nous allons devoir l'estimer. Pour cela, nous allons construire une collection de modèles pour le champ \mathbf{B} . Nous utiliserons ensuite un outil de sélection de modèles pour choisir parmi ces modèles. Nous sommes partis d'un modèle très général et répandu en statistiques spatiales : le modèle gaussien. Des travaux de détection en milieu non gaussien ont été menés par Pascal [Pas06]. Dans ce modèle, la loi du champ \mathbf{B} est entièrement déterminée par son vecteur moyenne \mathbf{m}_0 et sa matrice de covariance Σ . La collection de modèles sera donc une collection de couples (\mathbf{m}_0, Σ) . Il nous arrivera aussi de caractériser un modèle par le couple $(\mathbf{m}_0, \Sigma^{-1})$ où Σ^{-1} , l'inverse de la matrice de covariance, sera appelé *matrice de précision*. Après avoir remarqué que la moyenne pouvait être supprimée par un

La loi f du champ \mathbf{X} étant inconnue, nous ne pouvons pas identifier le modèle oracle. Nous avons cependant accès à I qui est une réalisation de la loi f . Nous allons donc utiliser le contenu de l'image I pour sélectionner un modèle \hat{m} dont le comportement théorique sera proche de celui du modèle oracle m^* . Pour cela, nous remplaçons le risque non calculable par un critère de log-vraisemblance pénalisée calculable directement à partir de l'image I . Le modèle \hat{m} sélectionné est alors défini par

$$\hat{m} \in \operatorname{argmin}_{m \in \mathcal{M}} [-\log(\hat{f}_m(I)) + \operatorname{pen}(m)] \quad (3)$$

où pen est une fonction positive ne dépendant que du modèle m . L'estimateur de f retenu est donné par $\hat{f} = \hat{f}_{\hat{m}}$.

En appliquant un théorème de Massart [Mas03], nous avons pu démontrer une *inégalité oracle* pour cette procédure de sélection de modèles. Elle compare le comportement de l'estimateur oracle \hat{f}_{m^*} à celui de l'estimateur constructible \hat{f} .

Malgré ses bons aspects théoriques, la procédure proposée (en particulier la recherche de maximum dans l'équation (3)) est trop coûteuse en temps de calcul pour les applications visées par TOSA. Nous avons donc limité la recherche du maximum à deux collections de partitions bien choisies [LPC13] : la collection des partitions dyadiques récursives et la collection des partitions récursives split. Grâce à la forme particulière de ces collections, présentées dans le chapitre 3, nous avons pu appliquer un algorithme de recherche de maximum, basé sur l'algorithme CART [BFSO84], plus rapide et plus efficace qu'un parcours de toutes les partitions de la collection. Le temps de calcul nécessaire est devenu correct pour une étude théorique mais reste encore trop élevé pour les applications en temps réel souhaitées par TOSA. Nous avons alors inversé le problème : au lieu de sélectionner pour chaque image un nouveau modèle, nous avons mené une étude empirique sur une large base d'images pour sélectionner un modèle que nous appliquerons ensuite à toutes les images. En d'autres termes, au lieu de faire la sélection de modèle dans le cœur de l'algorithme, nous avons fait la sélection de modèles en amont de l'algorithme.

Pour les applications pratiques, nous avons sélectionné la partition en rectangles de taille 16×16 (appelés patches dans la suite du manuscrit) et nous avons choisi comme structure parcimonieuse une structure aux 4 plus proches voisins. Les matrices de covariance Σ ou matrices de précision Σ^{-1} seront donc parcimonieuses avec au plus 5 coefficients non nuls par ligne.

Détection

L'objectif de la détection est double : nous voulons à la fois dire s'il y a ou non une cible dans l'image I mais aussi pouvoir situer cette cible. Nous effectuons donc un test par pixel (i, j) . Même s'il est localisé, ce test nécessitera quand même la valeur de l'image I pour l'ensemble (ou une partie) des pixels.

Tout au long de nos travaux, les cibles sont supposées sous-résolues et additives, c'est-à-dire que leur taille ne dépasse pas un pixel et que dans un pixel contenant une cible nous observons la valeur du fond (ciel) auquel s'ajoute la valeur de la cible (que l'on appellera intensité dans la suite).

Pour tester le pixel (i, j) , les hypothèses mises en jeu sont les suivantes :

$$\begin{aligned} H_0^{i,j} \text{ (hypothèse nulle) :} & \ll \text{Absence de cibles dans le pixel } (i, j) \gg \\ H_1^{i,j} \text{ (hypothèse alternative) :} & \ll \text{Présence d'une cible d'intensité } c > 0 \text{ inconnue dans le pixel } (i, j) \gg \end{aligned}$$

A l'aide d'une statistique de test calculée sur l'image I , nous allons faire un choix entre ces deux hypothèses. Nous appelons test le choix d'une statistique de test $\Delta(i, j)$ ainsi que la fonction qui à $\Delta(I, i, j)$ associe une décision ($H_0^{i,j}$ ou $H_1^{i,j}$). Un test peut être faux de 2 façons : soit il décide $H_1^{i,j}$ alors qu'il n'y a pas de cibles dans le pixel (i, j) (erreur de première espèce), soit il décide $H_0^{i,j}$ alors qu'il y a une cible dans le pixel (i, j) (erreur de seconde espèce). Il n'est pas possible de minimiser ces deux erreurs simultanément. Selon l'importance qu'on leur associe, la stratégie de test, et donc le choix de la statistique $\Delta(I, i, j)$, ne seront pas les mêmes.

Pour TOSA, le risque de rejeter l'hypothèse nulle alors qu'elle est vraie est beaucoup plus coûteux que celui de la conserver à tort. La stratégie adoptée consiste donc à imposer l'erreur de première espèce. Plusieurs tests le permettent. Toutefois, nous savons, grâce à la théorie Neyman-Pearson [Lé73], que le test dit du rapport de vraisemblance est le test qui a le plus faible taux d'erreur de seconde espèce parmi tous les tests ayant le même taux d'erreur de première espèce. Nous nous en sommes donc inspirés.

Notre modèle \hat{m} est donné par une partition $R = (R_1, \dots, R_P)$ de la grille S . Les pixels appartenant à des zones différentes sont supposés indépendants. La présence d'une cible dans le pixel (i, j) ne modifie donc pas le comportement des pixels dans les autres zones que celle contenant le pixel (i, j) . Elles ne sont donc pas discriminantes pour les hypothèses $H_0^{i,j}$ et $H_1^{i,j}$. Pour tenir compte de cette remarque, nous allons modifier un peu notre test. Les hypothèses ne portent plus que sur la zone de la partition de l'image I contenant le pixel (i, j) .

Si nous notons $R_{i,j}$ cette zone et $f_{i,j,0}$ (resp. $f_{i,j,1}$) la densité du champ \mathbf{X} restreint à la zone $R_{i,j}$ sous l'hypothèse $H_0^{i,j}$ (resp. $H_1^{i,j}$), le test du rapport de vraisemblance est alors de la forme

$$T^\alpha(i, j) = \mathbb{1}_{\frac{\hat{f}_{i,j,1}(I_{|R_{i,j}})}{\hat{f}_{i,j,0}(I_{|R_{i,j}})} \geq s} \quad \text{où } s \text{ est défini par } P\left(\frac{\hat{f}_{i,j,1}(I_{|R_{i,j}})}{\hat{f}_{i,j,0}(I_{|R_{i,j}})} \geq s | H_0\right) = \alpha.$$

La définition précédente du seuil s est une définition théorique qui garantit la probabilité de fausses alarmes. Cependant, nous ne sommes pas capables d'exprimer explicitement la loi de la variable aléatoire $\frac{\hat{f}_{i,j,1}(I_{|R_{i,j}})}{\hat{f}_{i,j,0}(I_{|R_{i,j}})}$ et donc incapables de connaître la valeur théorique de s . La valeur de ce seuil est cependant indispensable pour la mise en place de l'algorithme au complet.

Apprentissage des seuils

Il a donc fallu estimer la valeur de ce seuil s . Sur un échantillon d'apprentissage composé d'images de textures différentes, nous avons estimé ce seuil s comme quantile empirique de la statistique de test $\Delta(I, i, j)$. Cependant, ce choix de seuil ne donne pas les résultats souhaités. En effet, les fausses alarmes sont concentrées dans les zones de nuages et certaines cibles présentes dans les zones de ciel bleu ne sont pas détectées. La figure 1 permet de comprendre pourquoi. Elle représente les valeurs maximum de la statistique de test selon la texture dans le cas d'absence de cibles.

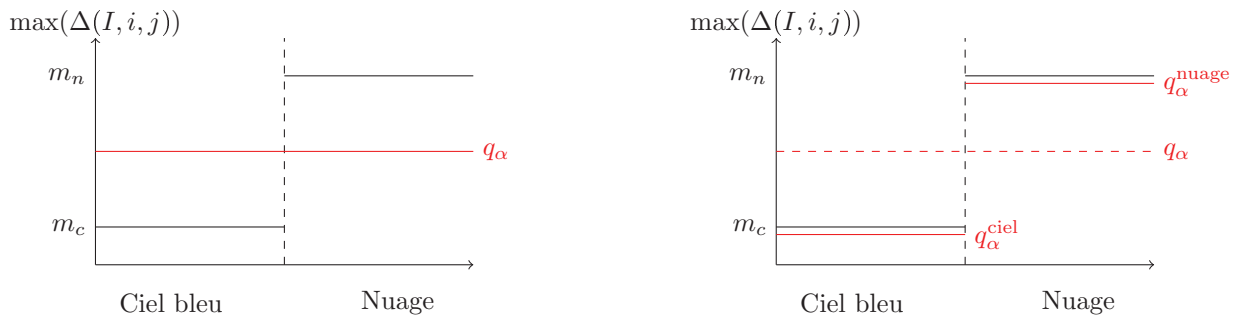


FIGURE 1 – Intérêt d'un apprentissage par texture

Supposons que l'on se fixe un niveau α et que l'on estime un seuil, noté q_α , sur tout l'échantillon. La probabilité de fausses alarmes sera garantie sur tout l'échantillon. En pratique, les fausses alarmes seront localisées dans les zones de nuages. De plus, ce seuil sera bien au dessus de la valeur maximale de la statistique de test dans les zones de ciel bleu et ainsi les cibles pour lesquelles la valeur de la statistique de test sera entre m_c et q_α ne seront pas détectées. En revanche, si nous estimons un seuil par texture, q_α^{zone} , la probabilité de fausses alarmes sera maîtrisée dans chacune des textures

et le taux de détection dans les zones de ciel sera augmenté.

Ainsi, nous avons souhaité estimer un seuil s par texture. Nous avons d'abord caractérisé le contenu d'un patch pour en déduire sa texture. Pour cela, nous avons calculé sur les patches d'un échantillon d'apprentissage des descripteurs proposés dans la littérature pour des applications de classification. Nous avons ensuite fait appel à des outils de classification non supervisée par modèles de mélanges de lois gaussiennes pour pouvoir décomposer notre échantillon d'apprentissage en différents sous-échantillons, chacun correspondant à une texture. Puis, nous avons estimé les seuils comme quantiles empiriques des différentes classes pour différentes probabilités de fausses alarmes souhaitées. A la fin de cette étape d'apprentissage, nous disposons d'une matrice de seuils : chaque ligne correspond à une classe et chaque colonne à une probabilité de fausses alarmes souhaitée.

Il est important de remarquer que cette étape d'apprentissage des seuils par classe est une étape qui se fait en amont de l'algorithme final et une seule fois. La matrice de seuils alors obtenue sera une donnée fixe de notre algorithme.

Principe de l'algorithme

Le schéma de la figure 2 permet de comprendre l'algorithme dans son intégralité.

Plusieurs données de fonctionnement sont nécessaires à l'algorithme : la définition des textures ainsi que l'outil de classification associé ; la matrice de seuils obtenue lors d'une phase d'apprentissage.

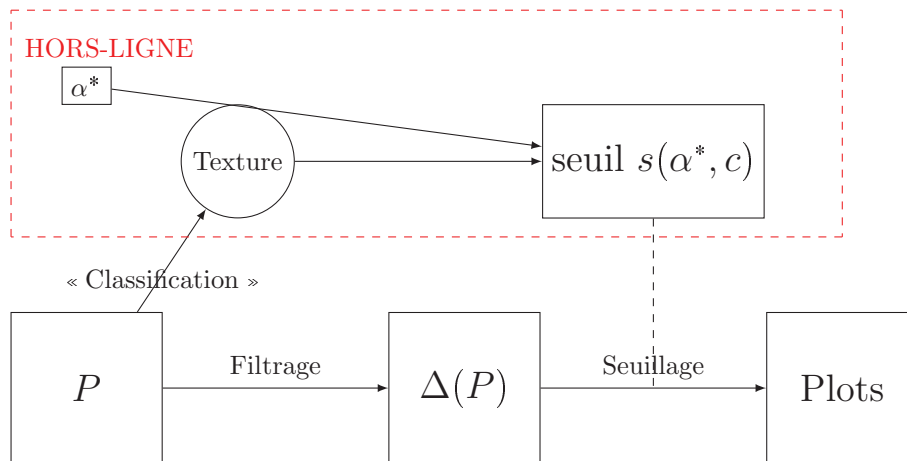


FIGURE 2 – Schéma final de l'algorithme

A l'entrée de l'algorithme se trouve un patch P de taille 16×16 ainsi que la probabilité de fausses alarmes souhaitée α^* . Nous calculons pour chacun des pixels (i, j) du patch P la statistique de test correspondant au couple d'hypothèses $H_0^{i,j}$, $H_1^{i,j}$, ce qui définit un nouveau patch $\Delta(P)$. En parallèle, le patch P est classé dans une des textures (classe k) grâce à l'outil de classification. A l'aide de la matrice de seuils, nous associons à la probabilité de fausses alarmes α^* et à la classe k un seuil $s(\alpha^*, k)$. Les pixels (i, j) pour lesquels la valeur $\Delta(I, i, j)$ dépasse le seuil $s(\alpha^*, k)$ sont considérés comme des pixels contenant une cible. Nous parlons alors de *plots*.

La partie entre pointillés du schéma de la figure 2 correspond à l'étape hors ligne de notre algorithme. La définition des classes ainsi que l'estimation des seuils sont des opérations effectuées une seule fois et en amont de la détection au sens strict. Ces opérations pourront bien sûr être relancées lorsque la base d'apprentissage augmentera ou si l'utilisateur souhaite effectuer la détection dans un milieu particulier. Il sera alors pertinent de construire une base d'apprentissage avec des images correspondant à ce milieu.

Contributions

Nos contributions lors de cette thèse sont aussi bien d'aspect théoriques que pratiques.

Nous avons construit une collection de modèles gaussiens stationnaires par morceaux dans le but de segmenter les images réelles en zones stationnaires. Dans ces zones, nous avons proposé une modélisation basée sur une matrice de covariance structurée et parcimonieuse. D'un point de vue pratique, nous avons également implémenté une procédure d'estimation des paramètres par maximum de vraisemblance pour ces modèles à partir des travaux de Burg et coauteurs [BLW82].

Afin de choisir entre ces modèles le plus adapté à nos données, nous avons traduit les hypothèses d'un théorème de sélection de modèles [Mas03] dans le cadre de nos modèles. Nous avons alors pu en déduire une inégalité oracle pour la procédure de sélection de modèles proposée. En parallèle, et assurés de la pertinence de notre approche par le résultat théorique obtenu, nous avons programmé un algorithme de segmentation d'images en zones rectangulaires à partir d'un critère de vraisemblance pénalisée et de l'algorithme CART [BFSO84].

Pour répondre aux attentes plus industrielle de ces travaux, nous avons mis en commun tous ces outils pour développer un algorithme de détection. Nous avons défini la statistique de test puis nous avons dû choisir à quel seuil elle serait comparée. Pour choisir ce seuil, nous avons défini une méthode d'apprentissage associant la classification de textures. Pour un échantillon d'images réelles de TOSA, nous avons classé des patchs de taille 16×16 en K classes grâce à des outils de classification non supervisée. Nous avons ensuite appris des quantiles empiriques pour chacune de ces classes.

Enfin, sur un échantillon de test, nous avons appliqué cette chaîne de détection afin d'en mesurer les performances. Ces dernières ont été comparées à l'état de l'art chez TOSA.

Organisation du manuscrit

Ce manuscrit est composé de quatre chapitres. Les chapitres 1 et 4 sont des chapitres d'applications. Les chapitre 2 et 3 contiennent les résultats théoriques sur lesquels nous nous sommes appuyés pour construire un nouvel algorithme de détection.

Dans le **chapitre 1**, nous rappelons les notions liées à la théorie des tests statistiques. Nous développons ensuite la théorie de la détection. Puis, nous détaillons le test du rapport de vraisemblance qui nous a servi d'heuristique pour la construction d'un nouveau test. Enfin, nous présentons les outils pour mesurer les performances d'un algorithme de détection de cibles sous-résolues.

Le **chapitre 2** est consacré à la présentation des modèles stationnaires et non stationnaires que nous proposons pour les fonds. Nous étudions deux modèles indirects en particulier puis quelques modèles directs. Enfin, nous introduisons la définition de modèle stationnaire par morceaux. Nous verrons comment résoudre les problèmes liés à cette définition.

Dans le **chapitre 3**, nous présentons le cadre théorique de la sélection de modèles par un critère de log-vraisemblance pénalisée. Nous appliquerons ensuite ces outils aux modèles définis dans le chapitre 2 pour modéliser au mieux les images réelles dont nous disposons. Enfin, nous verrons un critère alternatif plus rapide à calculer mais pour lequel nous n'avons pas de résultat théorique.

Le **chapitre 4** est dédié à la présentation de l'algorithme de détection que nous avons développé. Nous détaillons tout d'abord les différentes étapes de la construction de la statistique de test. Nous présentons ensuite les outils d'apprentissage et de classification utilisés lors de l'étape hors-ligne de notre démarche. Nous terminons ce chapitre par la présentation des performances de détection de l'algorithme proposé ainsi qu'une comparaison à un autre algorithme.

INTRODUCTION

Nous concluons ce manuscrit en dégageant les améliorations apportées par cette méthode ainsi que les perspectives qu'elle offre.

Chapitre 1

Théorie de la détection

Sommaire

1.1	Tests et stratégies	30
1.1.1	Rappels sur les tests	30
1.1.2	Différentes stratégies	32
1.1.3	Dans un cadre de détection	33
1.2	Test du rapport de vraisemblance pour la détection	35
1.2.1	Expression du test du rapport de vraisemblance	35
1.2.2	Test du rapport de vraisemblance généralisé	37
1.3	Mesures des performances d'un test de détection	39
1.3.1	Probabilité de fausses alarmes	39
1.3.2	Probabilité de détection	40
1.3.3	Courbes COR	41
1.3.4	Estimation des grandeurs	42
1.3.5	Echantillon d'images réelles	46

Introduction

Thales Optronique (TOSA) est une société du groupe Thales qui produit, entre autres, des caméras infra-rouges. Ces caméras peuvent être utilisées au sol ou embarquées à bord d'avions ou de bateaux. Elles permettent de récupérer une mesure du rayonnement infrarouge des scènes observées.

TOSA développe également des algorithmes pour « voir » de possibles menaces présentes dans ces scènes. Ces algorithmes fonctionnent en plusieurs étapes :

- **Détection** : l'utilisateur (ou l'algorithme) repère l'objet quand il est encore loin de la caméra. Cette étape peut aussi être divisée en plusieurs étapes.
 - *détection de plots* : Image par image, l'utilisateur localise un objet (un plot).
 - *détection de cibles* : En prenant en compte l'aspect temporel, plusieurs plots sont identifiés comme correspondant à la même cible. A l'inverse, lors de cette étape, certains plots sont identifiés comme des fausses détections et l'utilisateur ne s'y intéresse plus.
- **Reconnaissance** : l'utilisateur est maintenant capable de définir la nature de l'objet. Est-ce un oiseau, un hélicoptère, un avion, ... ?
- **Identification** : dans un contexte militaire par exemple, l'utilisateur est capable de dire si l'objet est ami ou ennemi.

Ces algorithmes sont basés sur des détections d'anomalie. Les images contenant des cibles sont considérées comme des « erreurs de comportement » du fond. Cette approche est beaucoup utilisée en détection [MDC10]. Elle est également utilisée chez TOSA et nous l'appliquerons nous aussi.

Les travaux présentés dans ce manuscrit s'inscrivent dans l'étape de détection de plots, lorsque l'objet est encore loin de la caméra. Chaque détection va entraîner une étape de reconnaissance puis d'identification. Ces étapes sont coûteuses. Il est donc indispensable que le détecteur ne déclenche pas trop de fausses alarmes, c'est-à-dire qu'il détecte un objet alors qu'il n'y en a pas.

Après avoir positionné la théorie de la détection dans un cadre plus général de test statistique ([Lé73], [Mic06]), nous verrons quelle est la stratégie qui permet de maîtriser au mieux ce nombre de fausses alarmes. Enfin, nous définirons les outils permettant de mesurer les performances d'un test de détection.

1.1 Tests et stratégies

Pour une image I , nous souhaitons répondre à la question suivante

Q : Y a-t-il une cible dans l'image I ?

Le problème de la détection est une application de la notion plus générale de tests statistiques. Dans la suite, nous allons rappeler quelques définitions liées à la notion de tests avant de voir plus en détail l'application à la détection.

1.1.1 Rappels sur les tests

Nous disposons d'une observation I pour laquelle nous souhaitons répondre à une question comme **Q** par exemple. Pour cette observation I , un test permet de faire un choix entre une hypothèse dite nulle, notée H_0 , et une hypothèse dite alternative, notée H_1 . Dans le cas de la question **Q**, les deux hypothèses mises en jeu sont les suivantes

H_0 (hypothèse nulle) : « Absence de cible dans l'image I »
 H_1 (hypothèse alternative) : « Présence d'une cible dans l'image I »

Nous appelons test de l'hypothèse H_0 contre l'hypothèse H_1 toute fonction T qui à une observation I associe une décision. La fonction T est donc à valeurs dans l'ensemble $\{H_0, H_1\}$. Pour simplifier les écritures, nous supposons que la fonction T est à valeur dans l'ensemble $\{0, 1\}$ en associant naturellement la valeur 0 à l'hypothèse H_0 et la valeur 1 à l'hypothèse H_1 . Finalement, la fonction T est une indicatrice $T = 1_\Delta$ et pour définir un test T il faut définir l'événement Δ .

L'ensemble de toutes les observations possibles est alors séparé en deux zones par le test : la zone d'acceptation pour les observations I telles que $T(I) = 0$ et la zone de rejet si $T(I) = 1$.

Les deux hypothèses H_0 et H_1 ne jouent pas des rôles symétriques. Le test de H_0 contre H_1 n'est pas le même que le test de H_1 contre H_0 . D'ailleurs, si un test T décide H_0 , on dit qu'il accepte l'hypothèse H_0 . S'il décide H_1 , on ne dit pas qu'il accepte l'hypothèse H_1 mais qu'il rejette l'hypothèse H_0 . L'hypothèse nulle doit être formulée dans le but d'être rejetée.

Il peut arriver que nous ne disposions pas d'a priori pour formuler correctement l'hypothèse alternative. Baraud et coauteurs [BHL03] ont alors proposé de remplacer l'hypothèse alternative H_1 par une collection d'hypothèses $\{H_{1,m}, m \in \mathcal{M}\}$. L'hypothèse H_0 est rejetée si un des tests de H_0 contre $H_{1,m}$ la rejette.

A partir de là, il existe deux méthodes statistiques assez éloignées pour mettre en place le test de l'hypothèse H_0 contre l'hypothèse H_1 .

Dans une approche modèle statistique, nous pouvons supposer que l'observation I est la réalisation d'une variable aléatoire \mathbf{X} . La connaissance ou l'estimation de la loi du champ \mathbf{X} sous chacune des deux hypothèses permettent de choisir entre les deux hypothèses H_0 ou H_1 .

Dans une approche apprentissage statistique, le test est envisagé comme une classification. L'image I appartient soit à la région d'acceptation soit à la région de rejet de l'hypothèse H_0 . Cette méthode est très bien présentée et schématisée dans l'article de Theiler [The14].

Dans nos travaux, nous avons utilisé la première approche. C'est celle employée chez TOSA. Nous verrons dans la dernière section de ce chapitre que cette méthode permet, en plus, de mesurer les performances théoriques des tests proposés.

D'un point de vue statistique, nous supposons donc que l'observation I est la réalisation d'une variable aléatoire que nous noterons \mathbf{X} dans la suite du manuscrit. Cette variable aléatoire n'a pas la même loi sous chacune des deux hypothèses. C'est justement cette différence de comportement que le test va essayer de discerner. Nous notons $f_{\mathbf{X}}(\cdot|H_0)$ la densité de la variable \mathbf{X} sous l'hypothèse H_0 et $f_{\mathbf{X}}(\cdot|H_1)$ sa densité sous l'hypothèse H_1 . La quantité $f_{\mathbf{X}}(I|H_k)$ représente alors la vraisemblance de l'observation I sous l'hypothèse H_k . La quantité $T(\mathbf{X})$ est elle aussi une variable aléatoire, à valeurs dans $\{0, 1\}$.

Lorsque nous effectuons un test de l'hypothèse H_0 contre l'hypothèse H_1 , nous ne savons pas sous quelle hypothèse nous nous trouvons. Quatre cas sont alors possibles.

- Nous sommes dans la situation H_0 et le test décide H_0 .
- Nous sommes dans la situation H_1 et le test décide H_1 .
- Nous sommes dans la situation H_0 et le test décide H_1 . Dans ce cas, nous parlons d'**erreurs de première espèce** et on note α , le taux d'erreurs de première espèce

$$\alpha = \mathbb{P}(T(\mathbf{X}) = 1|H_0)$$

Ce taux correspond à la probabilité de rejeter à tort l'hypothèse H_0 .

- Nous sommes dans la situation H_1 et le test décide H_0 . Dans ce cas, nous parlons d'**erreurs de seconde espèce** et on note β , le taux d'erreurs de seconde espèce

$$\beta = \mathbb{P}(T(\mathbf{X}) = 0|H_1).$$

Ce taux correspond à la probabilité d'accepter à tort l'hypothèse H_0 .

Les deux premiers sont des cas de bon fonctionnement du test T . Dans les deux derniers cas, le test T se trompe. Idéalement, nous souhaiterions définir un test T^* qui minimise simultanément ces deux taux d'erreurs. Cependant, il n'existe pas de lien direct entre les deux taux d'erreurs. Nous ne pouvons les minimiser simultanément. Nous allons donc devoir faire un compromis entre ces deux sources d'erreurs.

Nous pouvons alors définir différentes stratégies de tests selon l'importance associée à chacune de ces erreurs. Dans la suite, nous allons présenter trois stratégies classiques : celle de Bayes, celle du maximum a posteriori et celle de Neyman-Pearson [Lé73].

1.1.2 Différentes stratégies

Les deux premières stratégies présentées vont nécessiter un a priori sur les hypothèses H_0 et H_1 . Plus précisément, si nous notons $q = \mathbb{P}(H_0)$ et $p = \mathbb{P}(H_1) = 1 - q$, nous avons besoin de connaître les valeurs de p et de q pour mettre en place la stratégie de Bayes et celle du maximum a posteriori.

Dans le cas de la question **Q**, q représente la probabilité qu'il n'y ait pas de cible dans l'image I et p la probabilité qu'il y ait une cible dans l'image I .

Stratégie de Bayes

Dans cette stratégie, nous affectons un coût à chacune des décisions prises par le test. Nous notons $\Pi_{k,l}$ pour $k = 0, 1$ et $l = 0, 1$ le coût associé à la situation où $T(I)$ décide H_l et où l'hypothèse H_k est vraie. Il y a alors un coût r_0 associé à la décision H_0 et un coût r_1 associé à la décision H_1 .

$$\begin{aligned} r_0 &= \Pi_{00}\mathbb{P}(T(I) = 0|H_0) + \Pi_{01}\mathbb{P}(T(I) = 1|H_0) = \Pi_{00}(1 - \alpha) + \Pi_{01}\alpha \\ r_1 &= \Pi_{10}\mathbb{P}(T(I) = 0|H_1) + \Pi_{11}\mathbb{P}(T(I) = 1|H_1) = \Pi_{10}\beta + \Pi_{11}(1 - \beta) \end{aligned}$$

La stratégie de Bayes consiste à minimiser le coût moyen $R = qr_0 + pr_1$. Pour une observation I , le test de Bayes T_{Bayes} peut s'exprimer de la façon suivante [Lévine]

$$\begin{aligned} \text{Si } \frac{pf_{\mathbf{X}}(I|H_1)}{qf_{\mathbf{X}}(I|H_0)} &\leq \frac{\Pi_{01} - \Pi_{00}}{\Pi_{10} - \Pi_{11}} &\rightarrow &\text{ on décide } H_0. \\ \text{Si } \frac{pf_{\mathbf{X}}(I|H_1)}{qf_{\mathbf{X}}(I|H_0)} &\geq \frac{\Pi_{01} - \Pi_{00}}{\Pi_{10} - \Pi_{11}} &\rightarrow &\text{ on décide } H_1. \end{aligned}$$

Nous pouvons l'écrire sous forme d'indicatrice

$$T_{\text{Bayes}} = \mathbb{1}_{\frac{f_{\mathbf{X}}(I|H_1)}{f_{\mathbf{X}}(I|H_0)} \geq \frac{q(\Pi_{01} - \Pi_{00})}{p(\Pi_{10} - \Pi_{11})}}$$

Finalement, le test de Bayes consiste à calculer la quantité $\frac{f_{\mathbf{X}}(I|H_1)}{f_{\mathbf{X}}(I|H_0)}$, que nous appelons *statistique de test* puis à la comparer à un seuil. Nous verrons que toutes les stratégies présentées sont basées sur ce même principe.

Stratégie du Maximum A Posteriori

Dans cette stratégie, nous nous intéressons plutôt à la loi a posteriori de chacune des hypothèses. Pour une observation I et une hypothèse H_k , la vraisemblance a posteriori est définie par

$$f_{\mathbf{X}}(H_k|I) = \frac{\mathbb{P}(H_k)f_{\mathbf{X}}(I|H_k)}{qf_{\mathbf{X}}(I|H_0) + pf_{\mathbf{X}}(I|H_1)}.$$

Le test du maximum a posteriori T_{MAP} décide l'hypothèse H_k qui a la vraisemblance a posteriori $f_{\mathbf{X}}(H_k|I)$ la plus grande. Il est défini de la façon suivante

$$\begin{aligned} \text{Si } \frac{pf_{\mathbf{X}}(I|H_1)}{qf_{\mathbf{X}}(I|H_0)} &\leq 1 &\rightarrow &\text{ on décide } H_0. \\ \text{Si } \frac{pf_{\mathbf{X}}(I|H_1)}{qf_{\mathbf{X}}(I|H_0)} &\geq 1 &\rightarrow &\text{ on décide } H_1. \end{aligned}$$

Sous forme d'indicatrice, nous obtenons

$$T_{\text{MAP}} = \mathbb{1}_{\frac{f_{\mathbf{X}}(I|H_1)}{f_{\mathbf{X}}(I|H_0)} \geq \frac{q}{p}}$$

Nous pouvons déjà faire une remarque importante. Dans les deux stratégies présentées, la statistique de test à calculer dans les différents tests est la même : $\frac{f_{\mathbf{X}}(I|H_1)}{f_{\mathbf{X}}(I|H_0)}$. Le seuil auquel elle est comparée n'est pas défini de la même façon dans chacune des stratégies. Toutefois, si nous imposons un coût nul pour les deux situations de bon fonctionnement du test, $\Pi_{00} = \Pi_{11} = 0$, et

si nous supposons que les coûts associés aux deux types d'erreurs sont les mêmes, $\Pi_{01} = \Pi_{10}$, alors les deux stratégies sont identiques.

Ces deux stratégies ont le désavantage de nécessiter un a priori sur les réalisations des hypothèses H_0 et H_1 . Au contraire, la stratégie de Neyman-Pearson, présentée ci-dessous, n'utilise aucun a priori sur les probabilités de réalisation des hypothèses H_0 ou H_1 .

Stratégie de Neyman-Pearson

Pour un test T , nous ne pouvons pas minimiser simultanément le taux d'erreurs de première espèce, $\alpha(T)$, et le taux d'erreurs de seconde espèce, $\beta(T)$. Plus précisément, nous verrons dans le paragraphe 1.3 qu'une volonté de diminuer le taux d'erreur de première espèce va entraîner une augmentation du taux d'erreurs de seconde espèce. La stratégie de Neyman-Pearson date des années 1930 et propose de fixer la valeur du taux de première espèce et de minimiser le taux d'erreur de seconde espèce. Contrairement aux précédentes stratégies qui nécessitent des a priori sur les probabilités de réalisations des hypothèses H_0 et H_1 , l'a priori est ici sur le taux d'erreurs de première espèce.

Ainsi, pour un réel donné $\alpha^* \in [0, 1]$, le test de Neyman-Pearson de niveau α^* , noté $T_{NP}(\alpha^*)$, est défini comme le test ayant une erreur de seconde espèce $\beta(T)$ la plus faible possible parmi tous les tests ayant une erreur de première espèce $\alpha(T)$ égale à α^* . Nous pouvons donc le définir de la façon suivante

$$T_{NP}(\alpha^*) = \underset{T, \alpha(T)=\alpha^*}{\operatorname{argmin}} \beta(T).$$

La puissance d'un test T est définie par la quantité $1 - \beta(T)$. Son niveau correspond à son taux d'erreurs de première espèce $\alpha(T)$. Avec ce nouveau vocabulaire, nous pouvons dire que le test de Neyman-Pearson de niveau α^* est défini comme le test le plus puissant parmi tous les tests de niveau α^* .

Nous pouvons montrer que le test construit de la façon suivante

$$T(\alpha^*) = \mathbf{1}_{\frac{f_{\mathbf{X}}(I|H_1)}{f_{\mathbf{X}}(I|H_0)} \geq s} \quad \text{où} \quad \mathbb{P}\left(\frac{f_{\mathbf{X}}(I|H_1)}{f_{\mathbf{X}}(I|H_0)} > s | H_0\right) = \alpha^*$$

est le test le plus puissant parmi tous les tests de niveau α^* . Il s'agit donc du test de Neyman-Pearson de niveau α^* .

$$\boxed{T_{NP}(\alpha^*) = \mathbf{1}_{\frac{f_{\mathbf{X}}(I|H_1)}{f_{\mathbf{X}}(I|H_0)} \geq s} \quad \text{où} \quad \mathbb{P}\left(\frac{f_{\mathbf{X}}(I|H_1)}{f_{\mathbf{X}}(I|H_0)} > s | H_0\right) = \alpha^*}$$

Comme pour les stratégies précédentes, la statistique de test $\frac{f_{\mathbf{X}}(I|H_1)}{f_{\mathbf{X}}(I|H_0)}$ est comparée à un seuil. Toutefois, dans cette stratégie, le seuil dépend de la loi de la statistique de test sous l'hypothèse H_0 . Plus précisément, le seuil s correspond au quantile de niveau $1 - \alpha^*$ de la loi de la variable $\frac{f_{\mathbf{X}}(\mathbf{X}|H_1)}{f_{\mathbf{X}}(\mathbf{X}|H_0)}$ sous l'hypothèse H_0 .

De ces trois stratégies, aucune n'est meilleure que les autres. Tout dépend du cadre dans lequel nous travaillons. Nous allons voir que la stratégie de Neyman-Pearson est la plus appropriée à nos travaux de détection.

1.1.3 Dans un cadre de détection

Dans nos travaux, l'objectif principal est de détecter la présence d'une cible dans l'image I . Nous souhaitons également localiser cette cible dans l'image.

Nous travaillons avec des images I de taille $M \times M$ que l'on schématise par des matrices de taille $M \times M$. Chaque case de ces matrices est constituée d'un *pixel*. Dans la suite du manuscrit, nous n'allons plus considérer I comme un élément de $\mathbb{M}_M(\mathbb{R})$ mais comme un élément de \mathbb{R}^{M^2} . La matrice de pixels est réarrangée pour former un vecteur de taille M^2 comme expliqué sur la figure

1.1. La valeur présente dans le pixel (s, t) se retrouve à la position $(s-1)M+t$ du vecteur colonne. La notation I désignera aussi bien la matrice que le vecteur colonne équivalent. Enfin, puisque nous souhaitons conserver l'aspect spatial, pour les notations, nous utiliserons la notation $I_{s,t}$ pour faire référence à $I_{(s-1)M+t}$.

$$I = \begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & I_{s,t} & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} \in \mathcal{M}_M(\mathbb{R}) \quad \Rightarrow \quad I = \begin{pmatrix} \cdot \\ \cdot \\ I_{(s-1)M+t} \\ \cdot \\ \cdot \end{pmatrix} \in \mathbb{R}^{M^2}$$

FIGURE 1.1 – Transformation de l'observation matricielle en observation vectorielle

La localisation de la cible se traduit par un test portant sur un pixel précis. Pour chaque pixel (i, j) de l'image I , nous allons tenter de répondre à la question suivante.

Q' : Y a-t-il une cible dans le pixel (i, j) ?

Les deux hypothèses mises en jeu par le test vont alors être plus locales que celles présentées avant. Elles deviennent

$$\begin{aligned} H_0^{i,j} &: \text{« Absence de cible dans le pixel } (i, j) \text{ »} \\ H_1^{i,j} &: \text{« Présence d'une cible dans le pixel } (i, j) \text{ »} \end{aligned}$$

Avec ce test, nous ne décidons de la présence ou de l'absence d'une cible que dans le pixel (i, j) . Bien que la réponse du test ne concerne que le pixel (i, j) , la statistique de test ne sera pas seulement calculée à partir de la variable $X_{i,j}$ mais sur l'ensemble (ou un sous-ensemble) des pixels de l'image I . Pour effectuer la détection dans toute l'image I , nous lancerons ce test pour chacun de ses pixels.

Dans la suite, nous fixons un pixel (i, j) et nous noterons H_1 ou H_0 au lieu de $H_1^{i,j}$ et $H_0^{i,j}$ pour ne pas surcharger l'écriture.

Vocabulaire

Pour un test de détection T , nous parlons de fausses alarmes plutôt que d'erreurs de première espèce et de non détections plutôt que d'erreurs de seconde espèce. Dans la suite, nous appellerons probabilité de fausses alarmes le taux d'erreurs de première espèce du test T

$$\text{pfa}(T) = \alpha = \mathbb{P}(T(\mathbf{X}) = 1 | H_0)$$

et probabilité de détection la puissance du test T

$$\text{pd}(T) = 1 - \beta = 1 - \mathbb{P}(T(\mathbf{X}) = 1 | H_0) = \mathbb{P}(T(\mathbf{X}) = 1 | H_1)$$

Ainsi, le test de Neyman-Pearson de niveau α^* est celui qui a le meilleur taux de détection parmi tous les tests ayant un taux de fausses alarmes fixé à la valeur α^* .

Stratégie retenue

Les caméras produites par TOSA peuvent être utilisées dans un cadre de veille. Le système de surveillance est autonome et ne dispose pas toujours d'a priori sur la présence ou non d'une cible. Lorsqu'il en dispose, pour ne pas risquer d'accroître les erreurs, nous préférons ne pas utiliser ces informations comme des a priori. Pour ces raisons opérationnelles, les stratégies de Bayes et du maximum de vraisemblance ne sont pas les plus adaptées.

D'autre part, comme présenté au début de ce chapitre, chaque détection va entraîner une réaction d'un utilisateur humain ou d'un système d'armes pour obtenir plus d'informations sur la cible détectée, comme sa nature par exemple. Le nombre de fausses alarmes ne doit donc pas être trop élevé. Idéalement, ce nombre devrait être nul mais cela entraînerait alors un nombre de

détection également nul. Nous devons donc trouver un compromis acceptable pour l'utilisateur entre la probabilité de fausses alarmes et la probabilité de détection. En pratique, l'utilisateur autorise une probabilité de fausses alarmes limite, par exemple cinq fausses alarmes par image à la sortie de l'étape de détection de plots. Ce nombre sera fortement diminué grâce à l'étape de détection de cibles. Nous souhaitons alors avoir la meilleure probabilité de détection pour cette probabilité de fausses alarmes fixée par l'utilisateur.

Avec ces deux arguments, la stratégie de Neyman-Pearson nous paraît donc la stratégie la plus appropriée pour nos travaux parmi les trois présentées.

Hypothèses sur le fond et les cibles

Dans un cadre de détection, nous souhaitons, pour l'image I , bâtir un test ayant pour hypothèse nulle

$$H_0 : \ll \text{Absence de cible dans le pixel } (i, j) \gg$$

Le fond de ciel vu à travers une fenêtre de taille $M \times M$ est modélisé **en l'absence de cibles** par un champ gaussien sur \mathbb{R}^{M^2} , noté \mathbf{B} , de moyenne \mathbf{mo} et de matrice de covariance Σ .

$$\mathbf{B} \sim N(\mathbf{mo}, \Sigma)$$

Sous l'hypothèse H_0 , l'image I ne contient que le fond de ciel

$$\text{Sous } H_0, \forall (k, l) \in \llbracket 1, M \rrbracket^2, I(k, l) = \mathbf{B}(k, l).$$

Puisque nous souhaitons détecter les objets lorsqu'ils sont encore loin de la caméra (à grande portée), nous avons supposé les cibles ponctuelles. En imagerie, nous les qualifions de subpixeliques : leur taille ne dépasse pas un pixel. D'autre part, nous supposons que la contribution de la cible vient s'ajouter à celle du fond. Les cibles ont donc aussi été supposées additives.

Nous appelons *intensité* de la cible, que nous notons c , la quantité qui vient s'ajouter à la valeur du fond dans le pixel (i, j) contenant une cible. L'hypothèse alternative est alors la suivante

$$H_{1,c} : \ll \text{Présence d'une cible d'intensité } c > 0 \text{ dans le pixel } (i, j) \gg$$

Sous cette hypothèses, l'image est la somme du fond de ciel et de la cible.

$$\text{Sous } H_{1,c}, \forall (k, l) \in \llbracket 1, M \rrbracket^2, I(k, l) = \mathbf{B}(k, l) + c\delta_{i,j}.$$

où $\delta_{i,j}$ est un vecteur de taille M^2 avec un 1 à la position correspondant au pixel (i, j) et des 0 partout ailleurs.

1.2 Test du rapport de vraisemblance pour la détection

Dans cette partie, nous allons présenter plus en détail la théorie de Neyman-Pearson ainsi que la construction du test associé.

Nous disposons d'un échantillon $\mathbf{I} = (I_1, \dots, I_n)$ de n images réelles que nous supposons être des réalisations *indépendantes* d'un champ gaussien noté \mathbf{X} . Pour un niveau α^* , le test de Neyman-Pearson est défini de la façon suivante

$$T_{NP}(\alpha^*) = \mathbb{1}_{\frac{f_{\mathbf{X}}(\mathbf{I}|H_1)}{f_{\mathbf{X}}(\mathbf{I}|H_0)} \geq s} \text{ où } \mathbb{P}\left(\frac{f_{\mathbf{X}}(\mathbf{I}|H_1)}{f_{\mathbf{X}}(\mathbf{I}|H_0)} > s | H_0\right) = \alpha^*.$$

1.2.1 Expression du test du rapport de vraisemblance

La statistique de test utilisée dans le test de Neyman-Pearson est le rapport des vraisemblances de l'échantillon $\mathbf{I} = (I_1, \dots, I_n)$ sous chacune des hypothèses H_0 et H_1 . Pour cette raison, il est aussi appelé **Test du Rapport de Vraisemblance**. Il a été utilisé par Vasquez [Vas11] et Genin [Gen13] lors de leurs travaux sur la détection. Scharf et Firendlander [SF94] l'ont également étudié. Nous

le noterons TRV dans la suite du manuscrit. Nous notons également $\Delta(\mathbf{I}, i, j)$ la statistique de test associée

$$\Delta(\mathbf{I}, i, j) = \frac{f_{\mathbf{X}}(I_1, \dots, I_n | H_1)}{f_{\mathbf{X}}(I_1, \dots, I_n | H_0)}$$

Puisque les observations (I_1, \dots, I_n) sont des réalisations indépendantes de la même variable aléatoire \mathbf{X} , la vraisemblance de l'échantillon sous chacune des hypothèses est le produit de la vraisemblance de chacune des réalisations (I_1, \dots, I_n) .

$$\Delta(\mathbf{I}, i, j) = \frac{f_{\mathbf{X}}(I_1, \dots, I_n | H_1)}{f_{\mathbf{X}}(I_1, \dots, I_n | H_0)} = \prod_{k=1}^n \frac{f_{\mathbf{X}}(I_k | H_1)}{f_{\mathbf{X}}(I_k | H_0)}.$$

Pour calculer cette statistique, nous avons alors besoin de connaître la loi du champ \mathbf{X} sous chacune des deux hypothèses.

Hypothèse H_0

Sous l'hypothèse H_0 , il n'y a pas de cible et les observations ne contiennent que du fond de ciel. La loi du champ \mathbf{X} est donc la même que celle du champ \mathbf{B}

$$H_0 : \mathbf{X} \sim N(\mathbf{m}_0, \Sigma)$$

Sa densité est $f_{\mathbf{X}}(\mathbf{X} | H_0) = \frac{1}{(2\pi)^{\frac{M^2}{2}} \det(\Sigma)^{\frac{1}{2}}} \exp\left(-\frac{1}{2} {}^t(\mathbf{X} - \mathbf{m}_0) \Sigma^{-1} (\mathbf{X} - \mathbf{m}_0)\right)$.

La vraisemblance de l'échantillon est alors

$$f_{\mathbf{X}}(I_1, \dots, I_n | H_0) = \frac{1}{(2\pi)^{\frac{nM^2}{2}} \det(\Sigma)^{\frac{n}{2}}} \exp\left(-\frac{1}{2} \sum_{k=1}^n {}^t(I_k - \mathbf{m}_0) \Sigma^{-1} (I_k - \mathbf{m}_0)\right).$$

Hypothèse H_1

Sous l'hypothèse H_1 , il faut prendre en compte la présence de la cible au pixel (i, j) . Nous avons supposé les cibles subpixelles et additives. L'ajout d'une cible d'intensité c dans le pixel (i, j) ne va donc modifier la valeur de l'image qu'au niveau du pixel (i, j) . Ainsi, pour une cible d'intensité c présente dans le pixel (i, j) , le modèle suivi par le champ \mathbf{X} devient alors

$$H_{1,c} : \mathbf{X} \sim N(\mathbf{m}_0 + c\delta_{i,j}, \Sigma)$$

où $\delta_{i,j}$ est un vecteur de taille M^2 avec un 1 à la position correspondant au pixel (i, j) et des 0 partout ailleurs.

La densité de l'échantillon est alors

$$f_{\mathbf{X}}(I_1, \dots, I_n | H_1) = \frac{1}{(2\pi)^{\frac{nM^2}{2}} \det(\Sigma)^{\frac{n}{2}}} \exp\left(-\frac{1}{2} \sum_{k=1}^n {}^t(I_k - \mathbf{m}_0 - c\delta_{i,j}) \Sigma^{-1} (I_k - \mathbf{m}_0 - c\delta_{i,j})\right).$$

Nous pouvons maintenant construire la statistique de test du rapport de vraisemblance pour une intensité c .

$$\Delta(\mathbf{I}, i, j) = \frac{f_{\mathbf{X}}(I_1, \dots, I_n | H_1)}{f_{\mathbf{X}}(I_1, \dots, I_n | H_0)} = \exp\left(c {}^t \delta_{i,j} \Sigma^{-1} \left(\sum_{k=1}^n (I_k - \mathbf{m}_0)\right) - \frac{nc^2}{2} {}^t \delta_{i,j} \Sigma^{-1} \delta_{i,j}\right)$$

Détermination du seuil de détection

Cette quantité doit ensuite être comparée à un seuil. Si elle dépasse le seuil, on rejette l'hypothèse H_0 . Sinon, on l'accepte. Pour un niveau α^* , ce seuil est défini comme le quantile de niveau $1 - \alpha^*$ de la loi de la variable $\Delta(\mathbf{I}, i, j)$ sous l'hypothèse H_0 . Nous avons besoin de connaître la loi de cette statistique de test sous l'hypothèse H_0 .

Puisque la fonction $x \mapsto \exp(cx - \frac{nc^2}{2} {}^t \delta_{i,j} \Sigma^{-1} \delta_{i,j})$ est strictement croissante, nous allons seulement nous intéresser à la loi de la variable ${}^t \delta_{i,j} \Sigma^{-1} \left(\sum_{k=1}^n (I_k - \mathbf{m}_0)\right)$. Il est important de noter que cela n'est vrai que parce que nous avons supposé l'intensité de la cible c positive.

Sous l'hypothèse H_0 ,

$$\begin{aligned} \forall k \in \llbracket 1, n \rrbracket, I_k &\sim N(\mathbf{m}\mathbf{o}, \Sigma) \\ \sum_{k=1}^n (I_k - \mathbf{m}\mathbf{o}) &\sim N(0, n\Sigma) \\ {}^t\delta_{i,j}\Sigma^{-1} \left(\sum_{k=1}^n (I_k - \mathbf{m}\mathbf{o}) \right) &\sim N(0, n {}^t\delta_{i,j}\Sigma^{-1}\delta_{i,j}) \\ \frac{{}^t\delta_{i,j}\Sigma^{-1} \left(\sum_{k=1}^n (I_k - \mathbf{m}\mathbf{o}) \right)}{\sqrt{n {}^t\delta_{i,j}\Sigma^{-1}\delta_{i,j}}} &\sim N(0, 1) \end{aligned}$$

Finalement, pour un niveau $\alpha^* \in [0, 1]$, si q_{α^*} est le quantile de niveau $1 - \alpha^*$ de la loi normale centrée réduite $N(0, 1)$, le test est défini de la façon suivante

$$T_{NP}(\alpha^*) = 1_{\tilde{\Delta}(\mathbf{I}, i, j) > s} \quad \text{où} \quad \tilde{\Delta}(\mathbf{I}, i, j) = \frac{{}^t\delta_{i,j}\Sigma^{-1} \left(\sum_{k=1}^n (I_k - \mathbf{m}\mathbf{o}) \right)}{\sqrt{n {}^t\delta_{i,j}\Sigma^{-1}\delta_{i,j}}} \text{ et } s = q_{\alpha^*}.$$

Une remarque importante à faire est que ce test ne dépend finalement pas de l'intensité c de la cible que l'on veut détecter lorsque celle-ci est connue. Une autre remarque importante à faire est que ceci est dû à l'hypothèse gaussienne émise pour le champ \mathbf{X} . Pour une autre loi (un mélange de lois gaussiennes par exemple), nous n'aboutirions pas aux mêmes conclusions.

1.2.2 Test du rapport de vraisemblance généralisé

Dans la partie précédente, nous avons vu comment construire le test du rapport de vraisemblance lorsque tous les paramètres du modèle sont connus. Dans cette partie, nous allons voir comment se modifie ce test lorsque certains ou tous les paramètres sont inconnus [Kel86]. Nous parlons alors du **T**est du **R**apport de **V**raisemblance **G**énéralisé. Il est noté TRVG.

Intensité de la cible inconnue

Lorsque l'intensité de la cible est inconnue, nous devons l'estimer avant de pouvoir construire le test. Nous l'estimons par maximum de vraisemblance, ce qui nous conduit à utiliser le test du rapport de vraisemblance généralisé défini par

$$\frac{\max_{c>0} f_1(I_1, \dots, I_n, c)}{f_0(I_1, \dots, I_n)} = \frac{f_1(I_1, \dots, I_n, \hat{c}_{EMV})}{f_0(I_1, \dots, I_n)} \underset{H_1}{\overset{H_0}{\leq}} s.$$

où \hat{c}_{EMV} est l'estimateur du maximum de vraisemblance de l'intensité de la cible. Nous avons une formule exacte pour cet estimateur.

Puisque la fonction $x \mapsto \frac{1}{(2\pi)^{\frac{nM^2}{2}} \det(\Sigma)^{\frac{n}{2}}} \exp(-x)$ est strictement décroissante, le maximum de la vraisemblance est aussi le minimisateur de la fonction $c \mapsto \sum_{k=1}^n {}^t(I_k - \mathbf{m}\mathbf{o} - c\delta_{i,j})\Sigma^{-1}(I_k - \mathbf{m}\mathbf{o} - c\delta_{i,j})$. En cherchant un annulateur de la dérivée de cette fonction, nous obtenons une formule exacte pour cet estimateur

$$\hat{c}_{EMV} = \frac{{}^t\delta_{i,j}\Sigma^{-1} \left(\sum_{k=1}^n (I_k - \mathbf{m}\mathbf{o}) \right)}{n {}^t\delta_{i,j}\Sigma^{-1}\delta_{i,j}}$$

La vraisemblance devient alors

$$f_1(I, \hat{c}_{EMV}) = \frac{1}{(2\pi)^{\frac{nM^2}{2}} \det(\Sigma)^{\frac{n}{2}}} \exp \left(-\frac{1}{2} \sum_{k=1}^n {}^t(I_k - \mathbf{m}\mathbf{o})\Sigma^{-1}(I_k - \mathbf{m}\mathbf{o}) + \frac{\left[{}^t\delta_{i,j}\Sigma^{-1} \left(\sum_{k=1}^n (I_k - \mathbf{m}\mathbf{o}) \right) \right]^2}{2n {}^t\delta_{i,j}\Sigma^{-1}\delta_{i,j}} \right).$$

Finalement, la statistique de test est la suivante

$$\Delta(\mathbf{I}, i, j) = \frac{f_1(I_1, \dots, I_n, \hat{c}_{EMV})}{f_0(I_1, \dots, I_n)} = \exp\left(\frac{\left[{}^t\delta_{i,j}\Sigma^{-1}\left[\sum_{k=1}^n (I_k - \mathbf{m}\mathbf{o})\right]\right]^2}{2n{}^t\delta_{i,j}\Sigma^{-1}\delta_{i,j}}\right).$$

Cette quantité doit être comparée à un seuil qui dépend de la loi de cette variable sous l'hypothèse H_0 . Toujours par croissance de la fonction $x \mapsto \exp(\frac{x}{2})$, nous ne nous intéressons qu'à la variable

$$\frac{\left[{}^t\delta_{i,j}\Sigma^{-1}\left(\sum_{k=1}^n (I_k - \mathbf{m}\mathbf{o})\right)\right]^2}{{}^t\delta_{i,j}\Sigma^{-1}\delta_{i,j}}.$$

Sous l'hypothèse H_0 ,

$$\begin{aligned} {}^t\delta_{i,j}\Sigma^{-1}\left(\sum_{k=1}^n (I_k - \mathbf{m}\mathbf{o})\right) &\sim N(0, n{}^t\delta_{i,j}\Sigma^{-1}\delta_{i,j}) \\ \frac{{}^t\delta_{i,j}\Sigma^{-1}\left(\sum_{k=1}^n (I_k - \mathbf{m}\mathbf{o})\right)}{\sqrt{n{}^t\delta_{i,j}\Sigma^{-1}\delta_{i,j}}} &\sim N(0, 1) \\ \frac{\left[{}^t\delta_{i,j}\Sigma^{-1}\left(\sum_{k=1}^n (I_k - \mathbf{m}\mathbf{o})\right)\right]^2}{n{}^t\delta_{i,j}\Sigma^{-1}\delta_{i,j}} &\sim \chi^2(1) \end{aligned}$$

Finalement, pour un niveau $\alpha^* \in [0, 1]$, si qc_{α^*} est le quantile de niveau $1 - \alpha^*$ de la loi du Chi-deux à un degré de liberté $\chi^2(1)$, le test est défini de la façon suivante

$$T_{NP}(\alpha^*) = \mathbb{1}_{\hat{\Delta}(\mathbf{I}, i, j) > s} \quad \text{où} \quad \hat{\Delta}(\mathbf{I}, i, j) = \frac{\left[{}^t\delta_{i,j}\Sigma^{-1}\left(\sum_{k=1}^n (I_k - \mathbf{m}\mathbf{o})\right)\right]^2}{n{}^t\delta_{i,j}\Sigma^{-1}\delta_{i,j}} \quad \text{et } s = qc_{\alpha^*}.$$

Tous les paramètres inconnus

Cette fois-ci, tous les paramètres sont inconnus. Lorsque nous avons accès à plusieurs images sous les hypothèses $H_0^{i,j}$ ou $H_1^{i,j}$, nous pouvons les estimer par maximum de vraisemblance [MRG85]. Le test est alors le suivant

$$\frac{\max_{\mathbf{m}\mathbf{o}, \Sigma, c > 0} f_1(I_1, \dots, I_n, \mathbf{m}\mathbf{o}, \Sigma, c)}{\max_{\mathbf{m}\mathbf{o}, \Sigma} f_0(I_1, \dots, I_n, \mathbf{m}\mathbf{o}, \Sigma)} = \frac{f_1(I_1, \dots, I_n, \widehat{\mathbf{m}\mathbf{o}}_1, \widehat{\Sigma}_1, \hat{c}_1)}{f_0(I_1, \dots, I_n, \widehat{\mathbf{m}\mathbf{o}}_0, \widehat{\Sigma}_0)} \underset{H_1}{\underset{H_0}{\leq}} s.$$

où $(\widehat{\mathbf{m}\mathbf{o}}_1, \widehat{\Sigma}_1, \hat{c}_1)$ sont les estimateurs par maximum de vraisemblance des paramètres sous l'hypothèse H_1 et $(\widehat{\mathbf{m}\mathbf{o}}_0, \widehat{\Sigma}_0)$ ceux sous l'hypothèse H_0 . En utilisant les dérivées des vraisemblances sous chacune des hypothèses, nous obtenons les deux systèmes suivants

$$H_0 : \begin{cases} \widehat{\mathbf{m}\mathbf{o}}_0 = \frac{1}{n} \sum_{k=1}^n I_k \\ \widehat{\Sigma}_0 = \frac{1}{n} \sum_{k=1}^n {}^t(I_k - \widehat{\mathbf{m}\mathbf{o}}_0)(I_k - \widehat{\mathbf{m}\mathbf{o}}_0) \end{cases}$$

$$H_1 : \begin{cases} \hat{c}_1 = \frac{{}^t\delta_{i,j}\widehat{\Sigma}_1^{-1}\left(\sum_{k=1}^n (I_k - \widehat{\mathbf{m}\mathbf{o}}_1)\right)}{n{}^t\delta_{i,j}\widehat{\Sigma}_1^{-1}\delta_{i,j}} \\ \widehat{\mathbf{m}\mathbf{o}}_1 = \frac{1}{n} \sum_{k=1}^n (I_k - \hat{c}_1\delta_{i,j}) \\ \widehat{\Sigma}_1 = \frac{1}{n} \sum_{k=1}^n {}^t(I_k - \widehat{\mathbf{m}\mathbf{o}}_1 - \hat{c}_1\delta_{i,j})(I_k - \widehat{\mathbf{m}\mathbf{o}}_1 - \hat{c}_1\delta_{i,j}) \end{cases}$$

Il faut ensuite connaître la loi du rapport $\frac{f_1(I_1, \dots, I_n, \widehat{\mathbf{m}}\mathbf{o}_1, \widehat{\Sigma}_1, \widehat{c}_1)}{f_0(I_1, \dots, I_n, \widehat{\mathbf{m}}\mathbf{o}_0, \widehat{\Sigma}_0)}$ sous l'hypothèse H_0 pour pouvoir choisir le seuil s .

Plusieurs remarques sont alors à faire pour replacer la théorie de Neyman-Pearson dans notre cadre de travail. Tout d'abord, nous ne travaillons qu'avec une seule image I , c'est-à-dire que $n = 1$. Cela a pour conséquence, par exemple, que $\widehat{\mathbf{m}}\mathbf{o}_0 = I$ et que $\widehat{\Sigma}_0 = 0$. D'autre part, la loi de cette statistique $\frac{f_1(I_1, \dots, I_n, \widehat{\mathbf{m}}\mathbf{o}_1, \widehat{\Sigma}_1, \widehat{c}_1)}{f_0(I_1, \dots, I_n, \widehat{\mathbf{m}}\mathbf{o}_0, \widehat{\Sigma}_0)}$ n'est pas connue sous l'hypothèse H_0 et nous ne pouvons donc pas choisir le seuil s comme un de ces quantiles.

Dans la suite, nous n'appliquerons donc pas directement le test du rapport de vraisemblance généralisé. En revanche, la théorie de Neyman-Pearson va nous servir d'heuristique pour construire un test de détection. Nous allons définir une statistique de test $\Delta(I, i, j)$ plus simple que celle proposée par le test de vraisemblance généralisé. Nous ne connaissons pas non plus sa loi sous l'hypothèse H_0 mais nous l'apprendrons. Nous présenterons dans le chapitre 4 la statistique du test $\Delta(I, i, j)$ que nous avons utilisée ainsi que les méthodes d'apprentissage mises en œuvre.

1.3 Mesures des performances d'un test de détection

Supposons maintenant que nous ayons construit un test de détection T . Pour une observation I , nous notons $\Delta(I, i, j)$ la statistique de test et s le seuil auquel elle sera comparée. Si $\Delta(I, i, j) < s$, on accepte l'hypothèse H_0 . Sinon, on la rejette.

$$\Delta(I, i, j) \underset{H_1}{\overset{H_0}{\leq}} s.$$

Nous allons présenter plusieurs grandeurs utilisées dans un cadre de détection pour évaluer les performances de notre détecteur T . Ces grandeurs théoriques nécessitent pour leur calcul de connaître par exemple des lois statistiques, qui sont justement inconnues dans nos applications. Nous verrons, par la suite, comment les estimer. Ces grandeurs empiriques, mesurées sur notre détecteur T , permettront dans le chapitre 4 de le positionner vis-à-vis d'autres détecteurs déjà présents chez TOSA.

1.3.1 Probabilité de fausses alarmes

Nous avons vu que, pour répondre aux demandes de l'utilisateur, nous souhaitons construire un détecteur qui respecte la probabilité de fausses alarmes fixée par l'utilisateur. Le première grandeur caractéristique de notre détecteur est donc sa probabilité de fausses alarmes.

La *probabilité de fausses alarmes*, notée *pfa* dans la suite, est définie comme la probabilité que le test décide de rejeter l'hypothèse H_0 alors que l'hypothèse H_0 est vraie. Si \mathbf{X} est une variable aléatoire dont l'observation I est une réalisation,

$$\text{pfa}(T, s) = \mathbb{P}(\Delta(\mathbf{X}) > s | H_0) = \int_s^{+\infty} f_0(t) dt = 1 - F_0(s)$$

où f_0 est la densité de la variable $\Delta(\mathbf{X})$ sous l'hypothèse H_0 et F_0 sa fonction de répartition.

Il est important de noter que cette quantité est un paramètre intrinsèque du test et ne dépend pas de l'observation I mais seulement de la loi de la statistique de test $\Delta(\mathbf{X})$ sous l'hypothèse H_0 et du seuil s .

La fonction F_0 est croissante en tant que fonction de répartition. Ainsi, lorsque le seuil s diminue, la probabilité de fausses alarmes augmente.

Dans le cadre de nos travaux, nous souhaitons imposer à cette grandeur la valeur consigne α^* décidée par l'utilisateur.

$$\text{pfa}(T, s) = 1 - F_0(s) = \alpha^*.$$

Cette contrainte impose alors la valeur du seuil s à utiliser lors du test

$$s = F_0^{-1}(1 - \alpha^*).$$

Les tests ainsi construits sont dits à *taux de fausses alarmes constants* et notés TFAC [BF92]. C'est le cas du test de Neyman-Pearson de niveau α^* , $T_{NP}(\alpha^*)$, décrit précédemment.

1.3.2 Probabilité de détection

L'objectif principal d'un test de détection étant de détecter des cibles de différentes intensités, une des grandeurs caractéristiques d'un test est sa probabilité de détection.

La *probabilité de détection*, notée pd dans la suite du manuscrit, est définie comme la probabilité que le test décide de rejeter l'hypothèse H_0 quand H_0 est fautive. Elle dépend de l'intensité de la cible que l'on souhaite détecter.

$$pd(T, s, c) = \mathbb{P}(\Delta(\mathbf{X}) > s | H_1) = \int_s^{+\infty} f_{1,c}(t) dt = 1 - F_{1,c}(s)$$

où $f_{1,c}$ est la densité de la variable $\Delta(\mathbf{X})$ sous l'hypothèse $H_{1,c}$ pour une cible d'intensité c et $F_{1,c}$ sa fonction de répartition.

Là encore, il est à remarquer que cette quantité est un paramètre intrinsèque au test et ne dépend que de la loi de la statistique de test $\Delta(\mathbf{X})$ sous l'hypothèse $H_{1,c}$ et du seuil s . La figure 1.2 permet de visualiser graphiquement les probabilités de fausses alarmes et de détection associées à un seuil s . Puisque nous avons supposé les cibles additives, la densité sous l'hypothèse $H_{1,c}$ pour une intensité c est une translation de la densité sous l'hypothèse H_0 .

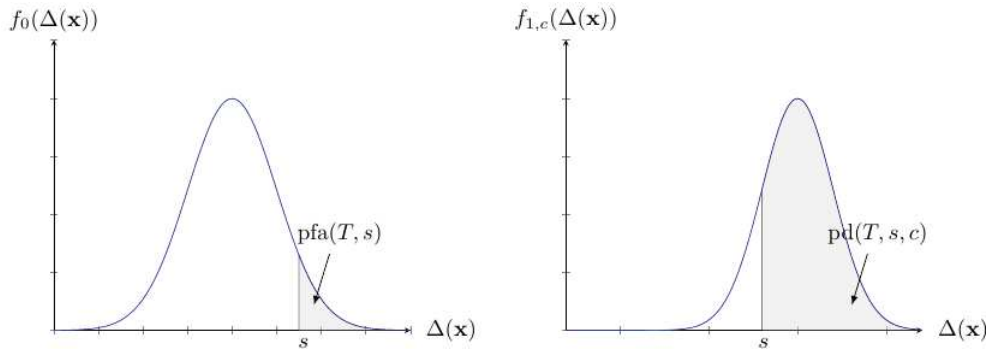


FIGURE 1.2 – Loi de la statistique de test sous chacune des hypothèses

Lorsque le seuil s est fixé, la quantité $pd(T, s, c)$ est croissante avec c : si l'intensité de la cible augmente, sa probabilité de détection aussi. À l'inverse, lorsque l'intensité c de la cible est fixée, la quantité $pd(T, s, c)$ est décroissante avec s : si le seuil augmente, la cible d'intensité c a une probabilité plus faible d'être détectée.

Si $pd(T, s, c) = 0$, les cibles d'intensité c ne seront pas détectées par le test T . Cela signifie que le seuil s est trop élevé. Pour détecter des cibles d'intensité c avec une probabilité de détection valant pd , il faudra alors abaisser le seuil

$$s = F_{1,c}^{-1}(1 - pd)$$

Cette diminution du seuil va entraîner une augmentation de la probabilité de fausses alarmes. Nous voyons alors qu'il n'est pas possible de régler le seuil s pour obtenir une probabilité de détection de 1 ainsi qu'une probabilité de fausses alarmes de 0. Il faudra réaliser un compromis entre ces deux grandeurs. La notion suivante va justement permettre de visualiser ce compromis.

1.3.3 Courbes COR

Lorsque la fonction F_0 est inversible, nous pouvons lier la probabilité de détection à la probabilité de fausses alarmes par l'intermédiaire du seuil s . Pour une intensité c , nous obtenons

$$\text{pd}(T, c) = 1 - F_{1,c} \left(F_0^{-1}(1 - \text{pfa}(T)) \right)$$

Courbes COR à intensité fixée

Pour une intensité c fixée, la courbe ainsi obtenue en traçant la probabilité de détection en fonction de la probabilité de fausses alarmes est appelée courbe COR du test T (COR pour *Caractéristique Opérationnelle de Réception*). Il y a une courbe COR pour chaque intensité. La figure 1.3 issue du manuscrit de Genin [Gen13] présente la forme générale d'une courbe COR.

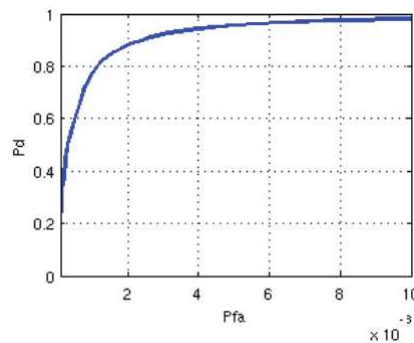


FIGURE 1.3 – Exemple de courbe COR à intensité fixée

Plusieurs remarques sont à faire sur les courbes COR :

- toutes les courbes commencent par le point (0,0), qui correspond en pratique à un seuil s qui tend vers $+\infty$. Il n'y a alors aucune fausse alarme mais aussi aucune détection.
- toutes les courbes finissent au point (1,1), qui correspond à un seuil s qui tend vers $-\infty$. La probabilité de détection vaut alors 1 mais la probabilité de fausses alarmes aussi.
- ces courbes sont croissantes. En effet, si la probabilité de fausses alarmes augmente alors le seuil diminue. Ainsi, le taux de détection augmente.

Chaque point de la courbe est obtenu pour un seuil particulier et correspond à un couple (pfa,pd) atteignable par le test de détection. Le test T ne pourra jamais atteindre un couple (pfa,pd) qui n'est pas sur la courbe (comme le couple (0,1) par exemple). Nous voyons alors que pour augmenter la probabilité de détection, l'utilisateur devra être prêt à augmenter la probabilité de fausses alarmes également.

Lorsque l'on veut comparer les performances de deux tests, nous pouvons comparer leur courbe COR pour différentes valeurs de l'intensité des cibles. Celui dont la courbe COR est supérieure a de meilleures performances de détection. La figure 1.4 issue du manuscrit de thèse de Vasquez [Vas11] donne un exemple d'une telle comparaison.

Si l'utilisateur souhaite atteindre une probabilité de détection de 0.8 (segment horizontal rouge), il devra accepter une probabilité de fausses alarmes de presque 0.01 avec le test en trait plein et une probabilité de fausses alarmes de 0.4 avec le test en trait pointillés. De même, si l'utilisateur souhaite imposer une probabilité de fausses alarmes de 0.001 (segment vertical bleu), il devra se contenter d'une probabilité de détection inférieure à 0.01 avec le test en trait pointillé et pourra atteindre une probabilité de détection supérieure à 0.5 avec le test en trait plein. Le test en trait plein a donc de meilleures performances que celui en trait pointillé.

Si les courbes se coupent, nous pourrions simplement dire que tel test est meilleur que tel autre pour une probabilité de fausses alarmes fixée et pas qu'il est meilleur pour toutes les probabilités de fausses alarmes.

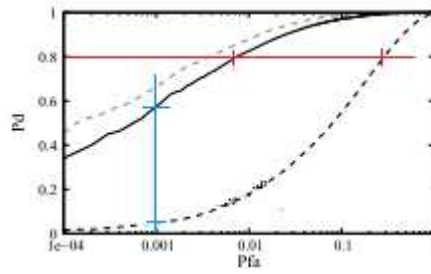


FIGURE 1.4 – Comparaison des courbes COR de deux tests

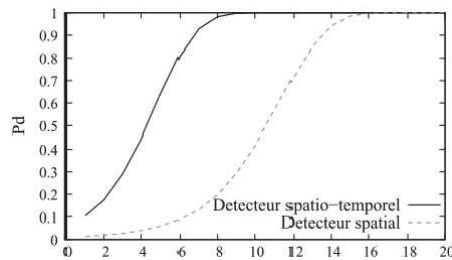
Le test le plus simple (le plus naïf) consiste à choisir au hasard entre les deux hypothèses. Pour ce test, les lois sous les deux hypothèses sont les mêmes. Ainsi, $pd(T)=pfa(T)$. La courbe COR de ce test est donc l'identité. Un test que l'on espère plus élaboré devra donc avoir une courbe COR au-dessus de l'identité. Sinon, il serait plus performant de choisir au hasard entre les deux hypothèses.

Courbes COR à pfa fixée

Dans le cadre de nos travaux, nous souhaitons travailler avec des tests TFAC, c'est-à-dire des tests à probabilité de fausses alarmes constante. Il peut être intéressant de tracer, pour une probabilité de fausses alarmes fixée, l'évolution de la probabilité de détection en fonction de l'intensité de la cible. Pour une probabilité de fausses alarmes fixée, $pfa=10^{-4}$ par exemple, le seuil correspondant est $s_{pfa} = F_0^{-1}(1 - pfa)$. La courbe qui nous intéresse est la suivante

$$pd(c) = 1 - F_{1,c}(s_{pfa})$$

La figure 1.5 issue du manuscrit de thèse de Vasquez [Vas11] donne un exemple d'une telle courbe pour une probabilité de fausses alarmes fixée à 10^{-4} . Cette courbe permet de lire la probabilité de détection que l'on peut atteindre pour différentes intensités lorsque la probabilité de fausses alarmes est fixée par l'utilisateur. Par exemple, pour une cible d'intensité $c = 8$, le détecteur en trait plein atteint une probabilité de détection de presque 0.9.

FIGURE 1.5 – Exemple de courbe COR à pfa fixée à 10^{-4}

Cette courbe permet aussi de mesurer quantitativement les différences entre deux algorithmes. Par exemple, nous lisons sur ce graphique de la figure 1.5 qu'au point de fonctionnement ($pfa=10^{-4}$, $pd=0.5$), le détecteur en trait plein détecte une cible à partir d'une intensité $c = 4$ alors que pour le détecteur en trait pointillé, il faut une cible d'intensité au moins $c = 10$. Le détecteur en trait plein est donc plus performant à ce point de fonctionnement.

1.3.4 Estimation des grandeurs

Dans nos travaux, nous avons utilisé des images réelles de TOSA. Ces images sont enregistrées lors de campagnes d'acquisition. Il n'y a pas toujours de champ spatial \mathbf{X} sous-jacent ou, lorsqu'il y en a un, nous ne connaissons pas sa loi.

Or, les trois grandeurs présentées dans le paragraphe précédent se calculent à partir de la loi du champ \mathbf{X} sous chacune des hypothèses H_0 ou H_1 . Dans les applications, nous allons donc avoir besoin d'estimer ces grandeurs pour mesurer les performances de notre test T .

Taille nécessaire de l'échantillon

Nous allons, dans un cadre plus général d'estimation de proportion, donner une formule permettant de connaître la taille de l'échantillon nécessaire à une bonne estimation.

Soit p la probabilité d'apparition d'un phénomène. Sur un échantillon de taille N , nous calculons f la fréquence d'apparition de ce phénomène. Si $N \geq 30$, $Np \geq 5$ et $Np(1-p) \geq 5$, l'approximation de la loi binomiale par la loi gaussienne nous permet d'obtenir un intervalle de confiance de risque r pour la grandeur f

$$\mathbb{P}\left(p - q_r \frac{p(1-p)}{Np} \leq f \leq p + q_r \frac{p(1-p)}{N}\right) = r$$

où q_r est le quantile de niveau $1 - r$ de la loi gaussienne centrée réduite. Pour que f soit une estimation de p avec une précision e et un risque r , il faut alors que la taille de l'échantillon vérifie la condition

$$q_r \frac{p(1-p)}{N} \leq e \Rightarrow N \geq \frac{q_r^2 p(1-p)}{e^2}.$$

Nous utiliserons ce résultat dans le chapitre 4 lorsque nous aurons besoin de bâtir un échantillon de test.

Taux de fausses alarmes

Puisque nous ne connaissons pas la densité f_0 du champ \mathbf{X} sous l'hypothèse H_0 , nous ne pouvons pas calculer la probabilité de fausses alarmes du test T . Nous ne pouvons que l'estimer empiriquement à partir des observations. Si nous avons accès à un échantillon (I_1, \dots, I_n) d'images réelles **ne contenant pas de cibles**, nous définissons la probabilité de fausses alarmes empirique de l'échantillon par

$$\widehat{\text{pfa}}_n(I, T, s) = \frac{1}{n} \# \{1 \leq k \leq n \mid \Delta(I_k, i, j) > s\}.$$

C'est la proportion d'images de l'échantillon pour lesquelles le test a décidé de rejeter l'hypothèse H_0 bien que H_0 soit vraie.

Lorsque $n = 1$, c'est-à-dire lorsqu'on ne travaille qu'avec une seule image, cette probabilité empirique vaut alors soit 0 soit 1. Nous souhaitons pouvoir atteindre des probabilités comprises entre 10^{-5} et 10^{-4} . Nous allons donc en donner une autre définition pour $n = 1$. Au lieu d'avoir accès à un échantillon (I_1, \dots, I_n) pour lesquels on ne teste que le pixel (i, j) , nous allons tester tous les pixels de l'image I . Pour une image I de taille $M \times M$, sa probabilité de fausses alarmes est défini comme la proportion de pixels pour lequel le test a décidé de rejeter l'hypothèse H_0 .

$$\widehat{\text{pfa}}_1(I, T, s) = \frac{1}{M^2} \# \{1 \leq k, l \leq M \mid \Delta(I, k, l) > s\}.$$

Contrairement à la probabilité de fausses alarmes théorique, la version empirique dépend de l'image sur laquelle elle est calculée. Nous parlerons alors du *taux de fausses alarmes* de l'image I plutôt que de sa probabilité de fausses alarmes.

Pour un test T TFAC, nous souhaitons que le taux de fausses alarmes de chaque image soit proche de la probabilité de fausses alarmes consigne imposée par l'utilisateur. Un graphique que nous utiliserons plusieurs fois dans le chapitre 4 sera donc celui du taux de fausses alarmes d'une image en fonction de la probabilité de fausses alarmes consigne de l'utilisateur. Nous vérifierons que cette courbe soit proche de l'identité. Un exemple en est donnée à la figure 1.6.

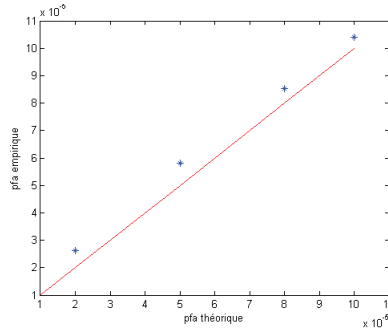


FIGURE 1.6 – Tracé du taux de fausses alarmes en fonction de la probabilité de fausses alarmes

Taux de détection

De même, lorsque la densité f_1 du champ \mathbf{X} sous l'hypothèse H_1 est inconnue, nous pouvons seulement estimer empiriquement la probabilité de détection à partir des données. Pour une intensité de la cible c , soit $(I_{c,1}, \dots, I_{c,n})$ un échantillon **contenant une cible d'intensité c** au pixel (i, j) . Le *taux de détection pour une intensité c* de l'échantillon I est défini par

$$\widehat{\text{pd}}_n(I, T, s, c) = \frac{1}{n} \# \{1 \leq k \leq n \mid \Delta(I_{c,k}, i, j) > s\}.$$

C'est la proportion d'images de l'échantillon pour lesquelles le test a décidé de rejeter l'hypothèse H_0 lorsque H_0 était fausse. C'est, là-aussi, un estimateur sans biais de $\text{pd}(T, c)$.

Lorsque $n = 1$, c'est-à-dire lorsqu'on ne travaille qu'avec une seule image I , ce taux ne prend lui aussi que deux valeurs : 0 ou 1. Nous allons alors utiliser l'image I pour fabriquer un échantillon d'images (J_1, \dots, J_{M^2}) contenant chacune une cible d'intensité c .

Nous avons supposé les cibles subpixelles et additives. La contribution de la cible n'est donc visible que dans le pixel où elle est ajoutée. Soit I une image ne contenant pas de cibles. Pour $(k, l) \in \llbracket 1, M \rrbracket^2$, si nous notons $J_{k,l}$ l'image obtenue après l'ajout d'une cible d'intensité c dans le pixel (k, l) , elle est donc définie de la façon suivante

$$\begin{cases} \forall (u, v) \neq (k, l), J_{k,l}(u, v) = I(s, t), \\ J_{k,l}(k, l) = I(k, l) + c; \end{cases} \quad (1.1)$$

En incrustant dans chacun des pixels de l'image I une cible d'intensité c , nous obtenons un échantillon (J_1, \dots, J_{M^2}) utile pour estimer la probabilité de détection du test T pour une intensité c . Pour l'image $J_{k,l}$, nous testons le pixel (k, l) .

$$\widehat{\text{pd}}_1(I, T, c, s) = \frac{1}{M^2} \# \{1 \leq k, l \leq M \mid \Delta(J_{k,l}, k, l) > s\}.$$

Modèle pour les cibles

Le modèle subpixelle et additif supposé pour les cibles est très, même trop, simple. D'autres modèles ont été envisagés dans la littérature. La cible, même en étant supposée plus petite qu'un pixel, est rarement comprise dans un seul pixel. Elle est plutôt contenue dans plusieurs pixels. La figure 1.7 présente différents modèles rencontrés dans la littérature pour des cibles de taille inférieure à un pixel.

0	0	0
0	$\frac{1}{2}$	$\frac{1}{2}$
0	0	0

0	0	0
0	$\frac{1}{4}$	0
0	$\frac{3}{4}$	0

0	$\frac{3}{16}$	0
$\frac{3}{16}$	$\frac{1}{4}$	$\frac{3}{16}$
0	$\frac{3}{16}$	0

FIGURE 1.7 – Exemples de signatures de cibles

Ces différents modèles sont plus réalistes pour les cibles que les algorithmes de TOSA cherche à détecter. Nous avons toutefois utilisé dans nos expériences le modèle subpixelle et additif.

Les performances que nous obtiendrons seront donc un peu optimistes. Pour se confronter à la réalité, nous appliquerons dans le chapitre 4 notre détecteur sur des cibles réelles, c'est-à-dire déjà présentes dans les images réelles de TOSA.

Courbes COR

Maintenant que nous savons estimer les probabilités de fausses alarmes et de détection pour un test T , nous allons pouvoir tracer des courbes COR empiriques pour chaque intensité c de la cible.

En réalité, ce n'est pas simplement l'intensité de la cible qui indique si elle est difficile à détecter ou non. C'est surtout son contraste vis-à-vis du fond. Nous définissons une grandeur que nous appelons *rapport signal sur bruit*, notée RSB dans la suite du manuscrit, et qui va mesurer le contraste de la cible vis-à-vis du fond.

$$RSB(c, i, j) = \frac{c}{\hat{\sigma}_{i,j}}$$

où $\hat{\sigma}_{i,j}$ est la variance empirique du fond estimée sur une fenêtre de taille 11×11 autour du pixel (i, j) . Nous reviendrons sur cette grandeur dans le chapitre 4.

Finalement, plutôt que de tracer des courbes COR pour une intensité fixée, nous allons tracer des courbes COR pour un rapport signal sur bruit fixé.

Nous disposons d'une image réelle I ne contenant pas de cibles. Pour chaque pixel (k, l) de cette image, nous calculons la statistique de test associé au test $T : \Delta(I, 1, 1), \dots, \Delta(I, M, M)$. Nous pouvons alors calculer le taux de fausses alarmes associé à cette image pour un seuil s

$$\widehat{\text{pfa}}_1(I, T, s) = \frac{1}{n} \# \{1 \leq k, l \leq M \mid \Delta(I, k, l) > s\}.$$

Nous souhaitons ensuite calculer le taux de détection de cette image pour un rapport signal bruit r fixé. Nous fabriquons comme précédemment, à partir de l'image I , un échantillon $(J_{1,1}, \dots, J_{M,M})$ de M^2 images contenant chacune une cible pour laquelle la valeur du rapport signal sur bruit vaut r . Nous pouvons alors calculer le taux de détection de cette image pour un rapport signal sur bruit r et un seuil s

$$\widehat{\text{pd}}_1(I, T, r, s) = \frac{1}{M^2} \# \{1 \leq k, l \leq M \mid \Delta(J_{k,l}, k, l) > s\}.$$

En faisant varier la valeur du seuil s , nous obtenons alors différents couples $(\widehat{\text{pfa}}_1(I, T, s), \widehat{\text{pd}}_1(I, T, r, s))$ qui correspondent à des points de fonctionnement du détecteur T . En traçant ces différents couples, nous obtenons ainsi une courbe COR empirique associée à la valeur r du rapport signal sur bruit. La figure 1.8 présente deux exemples de courbes COR empiriques pour le même détecteur T mais pour des rapports signal sur bruit différents.

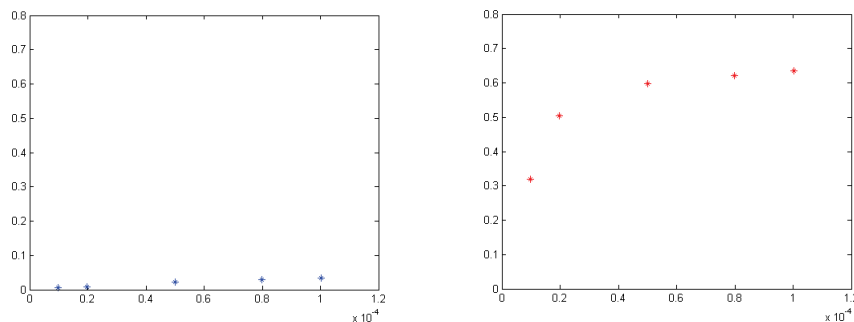


FIGURE 1.8 – Exemples de courbes COR empiriques pour différents rapports signal sur bruit

Les courbes COR empiriques ont bien le même aspect que la courbe COR théorique présentée sur la figure 1.3. Celle en bleu correspond à un rapport signal sur bruit de 3 tandis que celle en rouge,

à un rapport signal sur bruit de 10. Le détecteur est plus performant pour les cibles ayant un rapport signal sur bruit plus élevé, ce qui est le comportement attendu.

Nous pouvons aussi tracer la courbe COR à probabilité de fausses alarmes fixée. Cette courbe empirique représente le taux de détection d'un échantillon I en fonction du rapport signal sur bruit que l'on cherche à détecter. Un exemple est présentée sur la figure 1.9.

Comme expliqué précédemment, cette courbe va permettre de mesurer un gain de détection en terme de rapport signal sur bruit lorsque nous comparerons deux algorithmes.

Il est important de se rappeler que toutes ces courbes ont été obtenues à partir de l'image I . Ce sont des estimations des courbes COR théoriques et elles dépendent donc de l'image I sur laquelle elles ont été calculées.

Pour comparer les performances de deux algorithmes, nous allons utiliser les courbes COR empiriques obtenues par la méthode présentée ci-dessus. Nous ne pourrons comparer leurs courbes COR que si elles ont été obtenues à partir de la même image I . Les lectures des performances sur les courbes empiriques se feront comme expliqué au paragraphe 1.3.3.

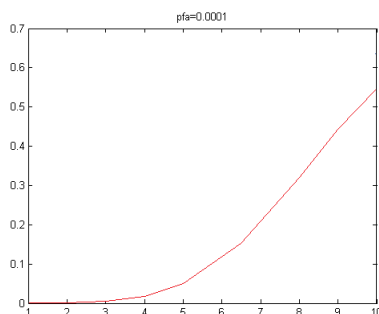


FIGURE 1.9 – Courbe COR pour une probabilité de fausses alarmes fixée

1.3.5 Echantillon d'images réelles

Les performances présentées au-dessus vont être mesurées sur différents échantillons d'images réelles fournies par TOSA. Ces images sont toutes des images infrarouges de taille 640×416 . La grande majorité ne contient pas de cibles. Quand des cibles sont présentes, nous parlons de cibles d'opportunités. Elles ont été labellisées par TOSA et nous pouvons ainsi faire tourner notre détecteur sur des cibles réelles. Les images ne contenant pas de cibles d'opportunités serviront à estimer les courbes COR empiriques.

La figure 4.8 représente quelques images réelles fournies par TOSA. Nous remarquons qu'elles sont de plusieurs natures : ciel bleu, nuages épais, nuages directionnels, ...

Ceci nous permettra de mesurer les performances de plusieurs détecteurs sur différents « types de fond ».

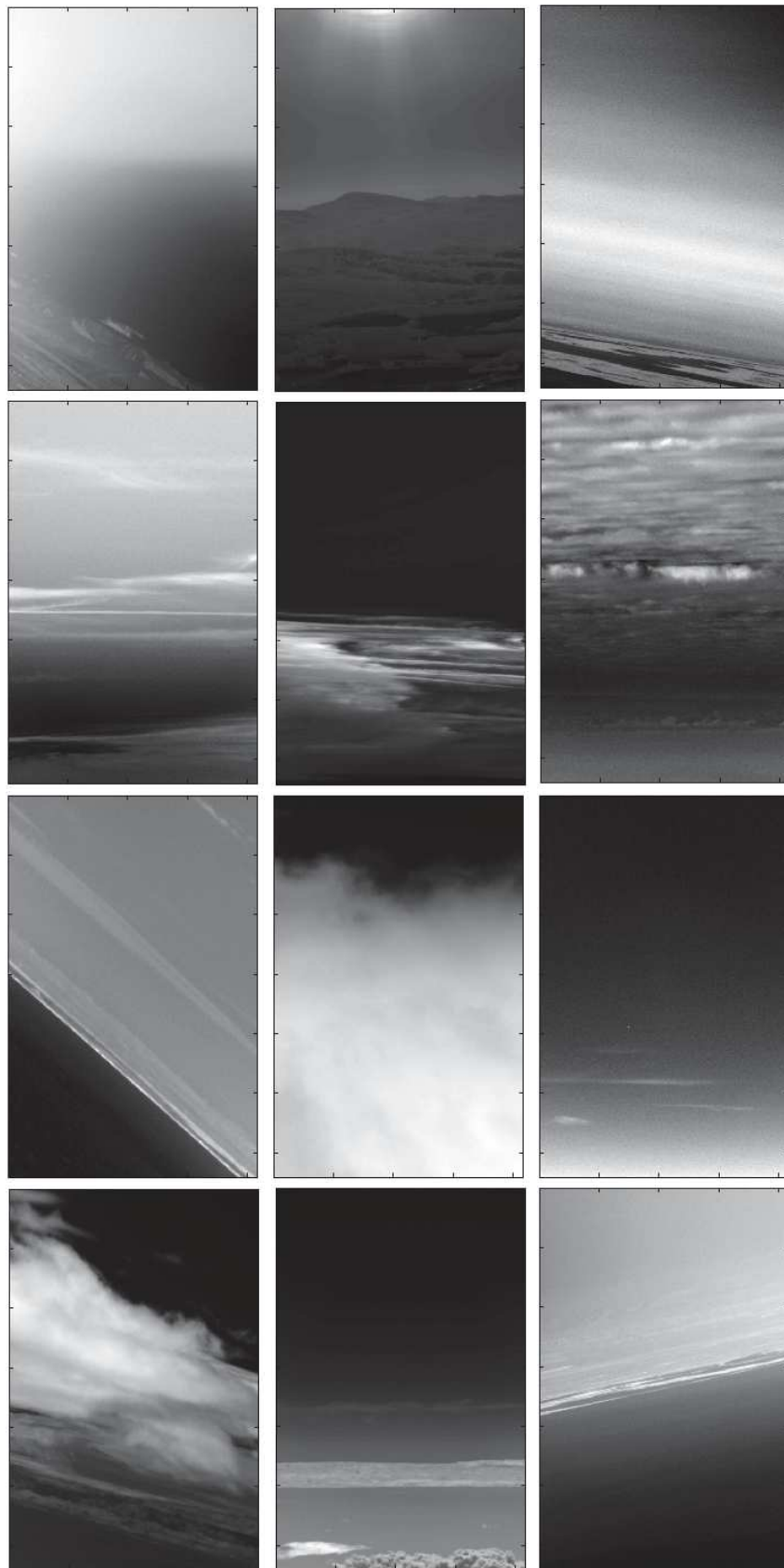


FIGURE 1.10 – Exemples d'images réelles de TOSA

Conclusion

Dans ce chapitre, nous avons présenté la stratégie de Neyman-Pearson. Cette stratégie a servi d'heuristique au développement du détecteur que nous proposons au chapitre 4.

Nous avons également proposé des grandeurs permettant de mesurer les performances d'un test de détection. Tous ces outils seront réutilisés dans le chapitre 4 lorsque nous comparerons notre détecteur à l'existant chez TOSA.

Pour mettre en place un test de détection, nous avons vu qu'il était très important de connaître le comportement des images réelles en absence et en présence de cibles. Puisque les cibles sont supposées additives, le comportement en présence de cibles se déduit de celui en absence de cibles. Dans le prochain chapitre, nous allons étudier le comportement des images en l'absence de cibles. Pour cela, nous allons définir plusieurs modèles statistiques qui permettront de les caractériser au mieux.

Chapitre 2

Modèles pour les textures

Sommaire

2.1 Premières modélisations	52
2.1.1 Estimation du vecteur moyenne	52
2.1.2 Hypothèses sur la matrice de covariance	55
2.2 Modèles indirects	56
2.2.1 Modèles auto-régressifs simultanés	56
2.2.2 Modèles auto-régressifs conditionnels	59
2.2.3 Intérêts des modèles SAR et CAR	60
2.2.4 Estimation des paramètres par maximum de vraisemblance	63
2.3 Modèles directs	64
2.3.1 Exemples de modèles directs	64
2.3.2 Estimation du maximum de vraisemblance sous contraintes	65
2.4 Modèles stationnaires par morceaux	67
2.4.1 Modèles SAR non stationnaires	68
2.4.2 Définition des modèles gaussiens finis stationnaires par morceaux	68
2.4.3 Estimation des paramètres par maximum de vraisemblance	71

Introduction

Le but principal de nos travaux est de proposer un algorithme de détection de cibles sous résolues dans des images de ciel. La connaissance (ou au moins l'estimation) d'un modèle pour représenter les images est indispensable à la mise en place d'un tel algorithme. Des expériences numériques ont montré qu'en l'absence de cibles, les images infra-rouges pouvaient être localement modélisées par des champs gaussiens stationnaires.

Nous allons, dans ce chapitre, présenter les modèles que nous avons choisi d'utiliser.

Nous observons une image I de taille $M \times M$. Cette image est vue comme la réalisation d'un champ gaussien spatial $\mathbf{X} = (X_{i,j}, i \in \llbracket 1, M \rrbracket, j \in \llbracket 1, M \rrbracket)$ de densité f devant la mesure de Lebesgue sur $\llbracket 1, M \rrbracket^2$ inconnues. Nous souhaitons estimer la densité f du champ \mathbf{X} .

Comme dans le chapitre précédent, nous n'allons plus considérer I comme un élément de $\mathbb{M}_M(\mathbb{R})$ mais comme un élément de \mathbb{R}^{M^2} . Nous allons de même supposer que le champ \mathbf{X} est un champ gaussien sur \mathbb{R}^{M^2} . Sa loi est donc entièrement caractérisée par un vecteur de moyenne $\mathbf{mo} \in \mathbb{R}^{M^2}$ et sa matrice de covariance $\Sigma \in \mathbb{M}_{M^2}(\mathbb{R})$.

Pour les notations, nous continuerons à utiliser la notation $X_{i,j}$ même s'il s'agit en réalité de la variable réelle $X_{(i-1)M+j}$. De même, pour une matrice A de taille $M^2 \times M^2$, nous noterons $A((i,j), (k,l))$ au lieu de $A((i-1)M+j, (k-1)M+l)$.

2.1 Premières modélisations

2.1.1 Estimation du vecteur moyenne

Selon la nature de l'image I , plusieurs hypothèses ont été proposées pour modéliser le vecteur \mathbf{mo} . Par exemple, lorsque I est une image ne contenant que du ciel bleu, Vasquez [Vas11] a proposé de modéliser le champ \mathbf{X} par la somme d'un plan et d'un bruit blanc gaussien.

$$\mathbf{X} = \mathbf{mo} + \sigma \mathbf{E} \text{ où } \begin{cases} \forall (i,j) \in \llbracket 1, M \rrbracket^2, mo_{i,j} = ai + bj + c \\ \mathbf{E} \sim N(0, Id) \end{cases} \quad (2.1)$$

Plus précisément, elle a supposé que cette hypothèse « plan + bruit blanc » était vraie localement. Pour tout pixel (i,j) , il existe une zone $\Omega_{r(i,j)}$, centrée en (i,j) , telle que sur cette zone, le modèle « plan+bruit blanc » soit vérifié.

Les paramètres a, b, c et σ sont estimés sur chacune des zones par maximisation de la vraisemblance. La figure 2.1a présente une image réelle de ciel bleu. La figure 2.1b présente le vecteur moyenne obtenu lorsqu'il est estimé localement par une fonction linéaire des coordonnées. L'histogramme de la figure 2.1c et la corrélation empirique de la figure 2.1d sont ceux du résidu \mathbf{E} obtenu comme différence entre l'image I et le vecteur moyenne estimé \widehat{mo} .

Nous voyons que pour ce type d'images, le modèle proposé par Vasquez est tout à fait adapté. Les résidus ont une corrélation proche de l'identité et leur histogramme est bien celui d'un champ gaussien. Essayons maintenant ce modèle sur une image réelle contenant à la fois du ciel bleu et des nuages.

Les résultats sont présentés par la figure 2.2. Sur cet exemple, l'histogramme des résidus est loin d'un histogramme gaussien. D'autre part, la matrice de covariance ne peut pas être supposée diagonale, de la forme $\sigma^2 Id$. Les résidus sont corrélés. Cela signifie que même localement, l'hypothèse « plan + bruit blanc » n'est plus vraie.

Dans un objectif d'estimation du vecteur moyenne, nous n'avons pas besoin de supposer un modèle pour le vecteur \mathbf{mo} . Une autre approche fréquemment utilisée et qui ne nécessite pas d'hypothèses sur le vecteur \mathbf{mo} est d'estimer les coordonnées $(mo_{i,j})_{1 \leq i,j \leq M}$ par une combinaison linéaire des valeurs du champ aux pixels voisins du pixel (i,j) .

$$\forall (i,j) \in \llbracket 1, M \rrbracket^2, \widehat{mo}_{i,j} = \sum_{(k,l) \in \llbracket 1, M \rrbracket^2} \theta(i,j,k,l) X_{k,l}. \quad (2.2)$$

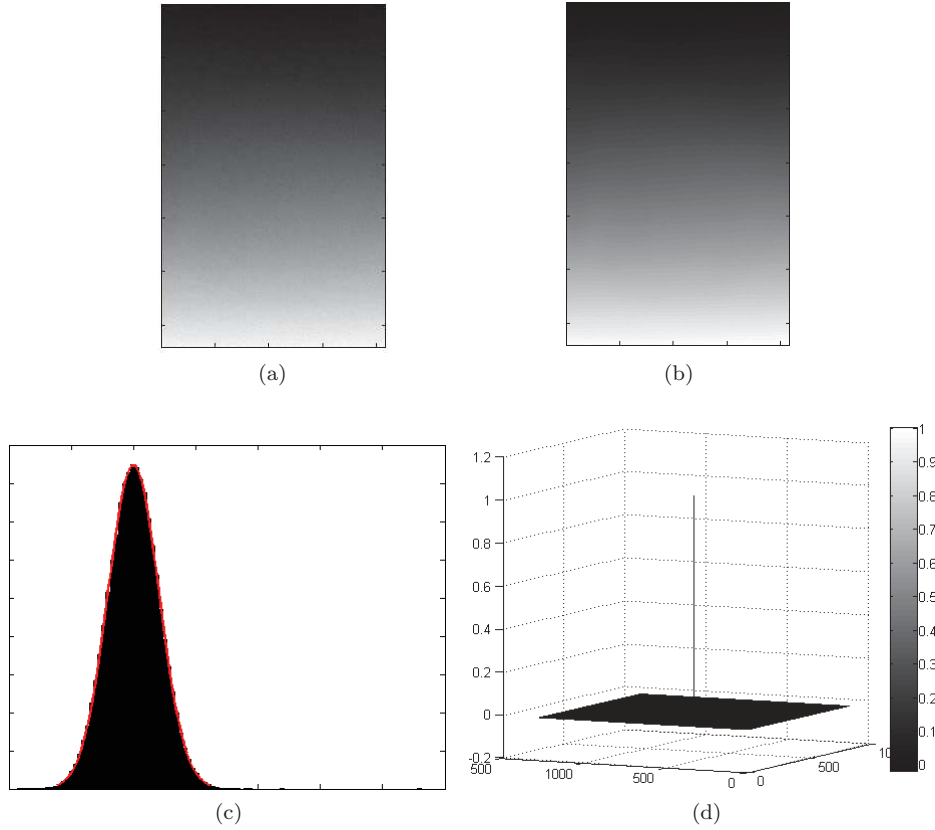


FIGURE 2.1 – Exemple de modélisation de mo par une fonction linéaire des coordonnées : (a) image réelle, (b) vecteur mo estimé, (c) histogramme des résidus et (d) Corrélations empiriques des résidus

Les coefficients $\theta(i, j, k, l)$ mesurent la pertinence du pixel (k, l) dans l'estimation de l'espérance du pixel $X_{i,j}$: $E(X_{i,j}) = mo_{i,j}$. Dans une approche locale, ils sont souvent nuls lorsque le pixel (k, l) est loin du pixel (i, j) , les seuls coefficients non nuls sont ceux correspondant aux 4 plus proches voisins.

$$\forall (i, j) \in \llbracket 1, M \rrbracket^2, \widehat{mo}_{i,j} = \theta_{0,1}X_{i,j+1} + \theta_{0,-1}X_{i,j-1} + \theta_{1,0}X_{i+1,j} + \theta_{-1,0}X_{i-1,j} \quad (2.3)$$

Cette formule pour définir l'estimateur entraîne immédiatement des problèmes de bords. Nous verrons au paragraphe 2.2.1 comment les résoudre.

Dans une approche non locale comme celle des *non-local means* (NL-means) proposée par Buades [BCM05], le coefficient $\theta(i, j, k, l)$ est nul lorsque le comportement du champ autour du pixel (k, l) est différent du comportement du champ autour du pixel (i, j) . Cette approche est basée sur le fait que dans une image, nous savons qu'il existe des zones similaires. Par exemple, lors de ses travaux de thèse, Genin [Gen13] a utilisé comme poids

$$\theta(i, j, k, l) = \exp\left(-\frac{d^2(P_{i,j}, P_{k,l})}{h}\right)$$

où $P_{i,j}$ est le patch carré centré en (i, j) et de taille $N = 5$ et d est une distance qui mesure la similarité entre les deux patches. Ce ne sont plus les pixels voisins du pixel (i, j) qui sont utilisés pour l'estimation de $mo_{i,j}$ mais plutôt les pixels dont le comportement est semblable à celui du pixel (i, j) , même s'ils sont loin.

Comme rappelé dans l'introduction, nous souhaitons travailler localement. Nous avons donc décidé de travailler avec la formule (2.3). Après plusieurs expériences sur les images réelles de TOSA, nous

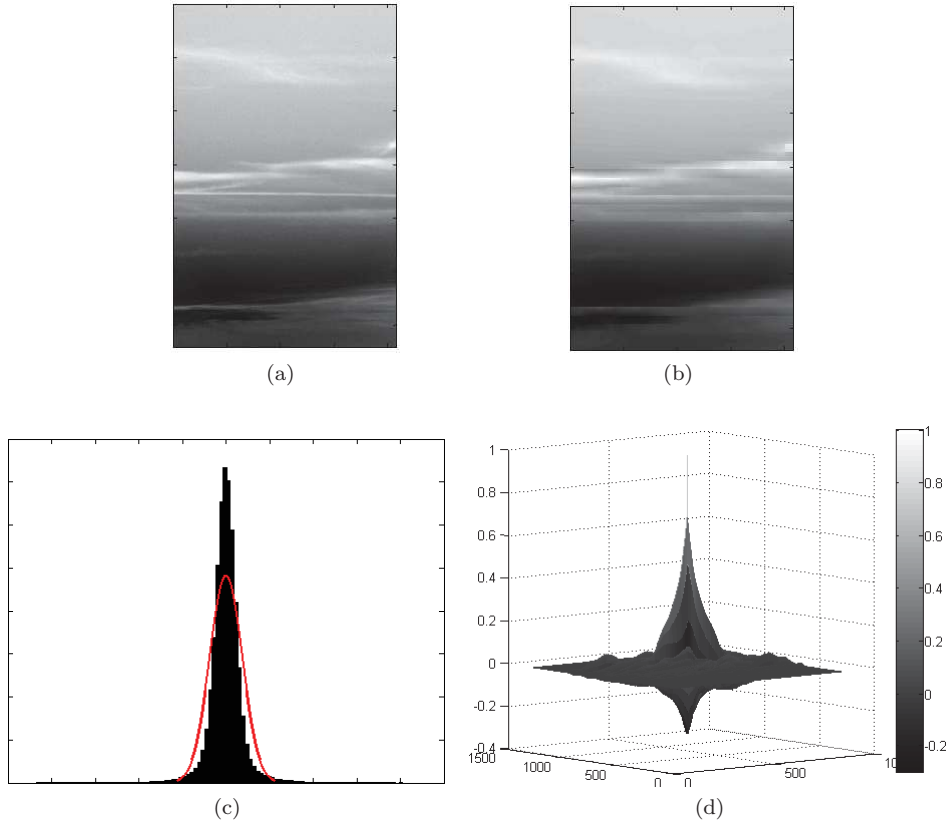


FIGURE 2.2 – Exemple de modélisation de mo par une fonction linéaire des coordonnées : (a) image réelle, (b) vecteur mo estimé, (c) histogramme des résidus et (d) Corrélacion empirique des résidus

avons décidé de supposer les coefficients $\theta(i, j, k, l)$ nuls lorsque le pixel (k, l) n'était pas un des quatre plus proches voisins du pixel (i, j) . Nous avons alors estimé le vecteur \mathbf{mo} par la moyenne des valeurs des pixels pour les quatre plus proches voisins

$$\forall (i, j) \in \llbracket 1, M \rrbracket^2, \widehat{m}_{o_{i,j}} = \frac{X_{i-1,j} + X_{i+1,j} + X_{i,j-1} + X_{i,j+1}}{4} \quad (2.4)$$

Cet estimateur ne suppose pas de modèle sur le vecteur \mathbf{mo} . De plus, si le champ \mathbf{X} suit le modèle supposé par Vasquez et rappelé en (2.1), nous obtenons un estimateur sans biais de l'espérance de $X_{i,j}$.

$$\begin{aligned} \mathbb{E}(\widehat{m}_{o_{i,j}}) &= \frac{a(i-1) + bj + c + a(i+1) + bj + c + ai + b(j-1) + c + ai + b(j+1) + c}{4} \\ &= ai + bj + c. \end{aligned}$$

C'est donc un estimateur plus général que celui que nous pourrions construire en supposant le modèle proposé par Vasquez.

La figure 2.3 présente l'estimation de \mathbf{mo} obtenue par la formule (2.4) ainsi que l'histogramme et la corrélation empirique des résidus pour l'image réelle de la figure 2.2. Nous pouvons alors faire plusieurs remarques. Tout d'abord, l'estimation de la moyenne visible sur la figure est visuellement bonne. D'autre part, l'histogramme des résidus est plus proche de l'histogramme gaussien que celui de la figure 2.2c. Enfin, la corrélation spatiale des résidus est plus faible que celle de la figure 2.2d. les résidus sont moins corrélés que dans le cas de l'estimation proposée par Vasquez.

Il reste toutefois de la dépendance locale et nous allons essayer dans la suite du chapitre de la modéliser.

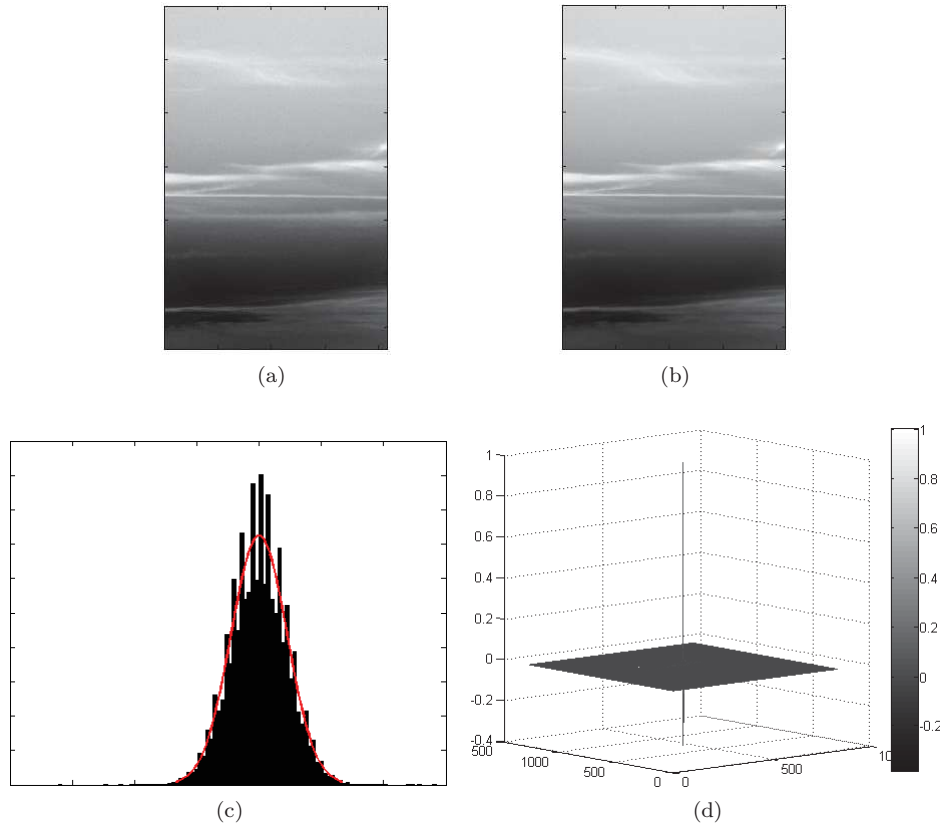


FIGURE 2.3 – Exemple de modélisation de mo par une fonction linéaire des coordonnées : (a) image réelle, (b) vecteur mo estimé, (c) histogramme des résidus et (d) Corrélation empirique des résidus

2.1.2 Hypothèses sur la matrice de covariance

Maintenant que nous avons estimé le vecteur moyenne, nous pouvons travailler avec le champ centré \mathbf{Y}

$$Y_{i,j} = X_{i,j} - \frac{X_{i,j-1} + X_{i,j+1} + X_{i-1,j} + X_{i+1,j}}{4}$$

Pour connaître entièrement la loi de \mathbf{Y} (et donc celle de \mathbf{X}), il reste à estimer la matrice de covariance de ce champ centré \mathbf{Y} ou, de manière équivalente, sa matrice de précision $Q = \Sigma^{-1}$. Ces matrices sont symétriques et de taille $M^2 \times M^2$. Sans aucune hypothèse sur les matrices Σ ou Q , il y a $\frac{M^2(M^2+1)}{2}$ coefficients à estimer. Or, nous ne travaillons qu'avec une image. Nous avons donc M^2 données pour estimer $\frac{M^2(M^2+1)}{2}$ coefficients.

Nous allons donc faire des hypothèses assez fortes sur la matrice Σ (ou sur la matrice Q) afin de réduire le nombre de paramètres à estimer. Chaque jeu d'hypothèses définira un modèle pour la structure de la matrice Σ (ou pour la matrice Q). L'idée générale est de supposer que la matrice Σ (ou la matrice Q) ne dépend que d'un petit nombre de paramètres.

$$\begin{cases} \Sigma = \Sigma(\theta) \\ Q = Q(\theta) \end{cases} \quad \text{où } \theta \in \mathbb{R}^d \text{ et } d \ll \frac{M^2(M^2+1)}{2}$$

d est alors la dimension du modèle.

La première hypothèse que nous pouvons faire pour réduire la dimension du modèle pour Σ est de supposer le champ \mathbf{Y} *stationnaire*.

Définition 1 (Stationnarité) On dit qu'un champ \mathbf{Y} sur \mathbb{Z}^2 centré est stationnaire si

$$\forall (i, j, k, l) \in \mathbb{Z}^4 \text{ Cov}(Y_{i,j}, Y_{k,l}) = C(i - k, j - l)$$

où C est une fonction définie sur \mathbb{Z}^2 .

Cela signifie que la covariance entre le champ \mathbf{Y} au pixel (i, j) et le champ \mathbf{Y} au pixel (k, l) ne dépend que de la position relative du pixel (i, j) par rapport au pixel (k, l) . La matrice de covariance devient alors une matrice de Toeplitz symétrique. Le nombre de coefficients à estimer a fortement diminué : il passe de $\frac{M^2(M^2+1)}{2}$ à M^2 . C'est un gain majeur pour des situations comme la notre où le nombre de données est assez faible.

Pour modéliser le champ \mathbf{Y} , nous pourrions nous intéresser à la fonction C directement ou au variogramme associé

$$\forall h \in \mathbb{Z}^2, \gamma(h) = C(0) - C(h).$$

Plusieurs modèles sont utilisés dans la littérature [Guy07] ou [Cre93]. Nous allons plutôt travailler avec la matrice Σ .

Dans la suite, nous allons présenter les différents modèles paramétriques stationnaires utilisés pour la matrice Σ ou pour la matrice Q . Nous appellerons *modèles directs* les modèles définis par la matrice de covariance Σ et *modèles indirects* ceux définis par la matrice de précision $Q = \Sigma^{-1}$.

2.2 Modèles indirects

Les modèles que nous allons présenter dans cette partie sont des modèles définis à travers la matrice de précision Q . Cette matrice a donc une structure particulière et connue. De plus, puisque nous souhaitons réduire le nombre de paramètres, cette structure sera souvent parcimonieuse. Nous en avons étudiés deux en particulier. Ce sont des modèles inspirés des modèles auto-régressifs en dimension un. Ils ont déjà été utilisés pour modéliser des fonds gaussiens dans un objectif de détection. Nous pouvons citer Denney et Figueiredo [DF00], Goldman et Cohen [GC05] ou encore Vezzosi [Vez78].

2.2.1 Modèles auto-régressifs simultanés

Le premier modèle indirect qui nous a intéressés est le modèle Auto-Régressif Simultané, noté SAR en abrégé.

Définition 2 (SAR stationnaire) Soit N une partie finie de \mathbb{Z}^2 . On dit qu'un champ \mathbf{Y} gaussien sur \mathbb{Z}^2 est un champ SAR stationnaire pour le voisinage N si

$$\forall (i, j) \in \mathbb{Z}^2, Y_{i,j} = \sum_{(k,l) \in N} a_{k,l} Y_{i-k, j-l} + \sigma \varepsilon_{i,j} \quad (2.5)$$

où $\sum_{(k,l) \in N} |a_{k,l}| < 1$ et ε est un bruit blanc gaussien.

Ces modèles ont été introduits par Whittle en 1954 [Whi54] et redéfinis par Guyon [Guy07]. Ils sont dits *simultanés* car toutes les équations sont nécessaires pour connaître les lois jointes, marginales ou conditionnelles associées.

Un des voisinages les plus souvent utilisés est le voisinage des quatre plus proches voisins $N_4 = \{(-1, 0), (1, 0), (0, -1), (0, 1)\}$. Nous verrons plus en détails au chapitre suivant des travaux de Verzelen [Ver09] ou de Kashyap et Challapa [KC83] permettant de choisir pour une image donnée le voisinage le plus adapté.

Si nous supposons de plus que les coefficients $a_{k,l}$ sont tous égaux à a avec $|a| < \frac{1}{4}$, nous obtenons alors un *modèle SAR isotrope aux quatre plus proches voisins*

$$\forall (i, j) \in \mathbb{Z}^2, Y_{i,j} = a(Y_{i-1,j} + Y_{i+1,j} + Y_{i,j-1} + Y_{i,j+1}) + \sigma \varepsilon_{i,j}. \quad (2.6)$$

Nous avons beaucoup utilisé ce modèle dans la suite de nos travaux.

Un autre modèle SAR souvent rencontré est le modèle qui pourrait être appelé modèle SAR stationnaire directionnel. Pour tout pixel (i, j) , les coefficients devant les voisins horizontaux $(i, j - 1)$ et $(i, j + 1)$ ont la même valeur. De même, les coefficients devant les voisins verticaux $(i - 1, j)$ et $(i + 1, j)$ ont la même valeur.

$$\forall (i, j) \in \mathbb{Z}^2 \quad Y_{i,j} = a_1(Y_{i-1,j} + Y_{i+1,j}) + a_2(Y_{i,j-1} + Y_{i,j+1}) + \sigma \varepsilon_{i,j}. \quad (2.7)$$

La définition et les deux exemples vus précédemment supposent que le champ \mathbf{Y} est défini sur \mathbb{Z}^2 . Pour obtenir des réalisations de taille finie de tels champs gaussiens stationnaires, nous devons d'abord traiter les problèmes de bords liés au passage de la grille infinie \mathbb{Z}^2 à une grille finie. Que devient la définition précédente si nous restreignons le champ \mathbf{X} à une grille $\llbracket 1, M \rrbracket^2$? La question concerne surtout la définition du voisinage N pour les pixels aux bords de la grille. Par exemple, quels sont les quatre plus proches voisins du pixel $(1, 1)$? Deux réponses sont possibles.

Voisinage périodisé : le champ est supposé périodique et il en est de même pour le voisinage.

Le voisinage du pixel $(1,1)$ est donc composé de quatre pixels : $(1,2)$, $(2,1)$, $(1,M)$ et $(M,1)$.

Un exemple est donné par le pixel et les croix bleus sur la figure 2.4.

Voisinage non périodisé : le champ n'est pas périodique. Le voisinage est limité aux pixels spatialement voisins. Le voisinage du pixel $(1,1)$ est donc composé de deux pixels : $(1,2)$ et $(2,1)$. Un exemple est donné par le pixel et les croix noirs sur la figure 2.4.

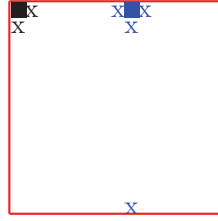


FIGURE 2.4 – Choix de voisinage pour des champs de taille finie

Il est important de noter que dans le cas d'un voisinage non périodisé, le champ \mathbf{Y} n'est plus stationnaire au sens de la définition 1. C'est toutefois cette définition du voisinage que nous allons conserver. En effet, pour des images réelles, il n'y a pas de raison de penser que la relation entre des pixels éloignés comme $(1,1)$ et $(1,M)$ soit la même que celle entre des pixels proches comme $(1,1)$ et $(1,2)$. Les problèmes de bords ont été étudiés par Dalhaus et Künsch [DK87]. Ce passage d'une grille infinie \mathbb{Z}^2 à une grille finie $\llbracket 1, M \rrbracket^2$ a aussi des conséquences sur l'estimation des paramètres [Guy07]. Dans la suite, nous utiliserons la définition 2 lorsque le champ sera défini sur \mathbb{Z}^2 . Pour une taille finie, nous utiliserons le voisinage non périodisé.

Dans la suite de nos travaux, nous avons supposé que le champ ε était un champ gaussien. Il est toutefois possible de supposer une autre loi pour le champ ε . La figure 2.5 présente deux simulations de champs SAR pour des bruits de lois différentes. A gauche, le champ ε suit une loi gaussienne tandis qu'à droite, il suit une loi de mélange de gaussiennes.

Dans la suite, nous supposons que le champ ε est un champ gaussien.

Pour voir que le modèle SAR gaussien stationnaire est bien un modèle indirect, il faut s'intéresser à la matrice de précision du champ \mathbf{Y} . Pour un modèle défini par les paramètres $a = (a_{k,l}, (k, l) \in N)$ et σ , nous notons $Q_s(a, \sigma)$ cette matrice de précision. Alors, on a la relation

$$Q_s(a, \sigma) = \frac{1}{\sigma^2} {}^t(I - A_a)(I - A_a)$$

où A_a est la matrice définie par

$$A_a((i, j), (i - k, j - l)) = \begin{cases} a_{k,l} & \text{si } (k, l) \in N \\ 0 & \text{sinon.} \end{cases}$$

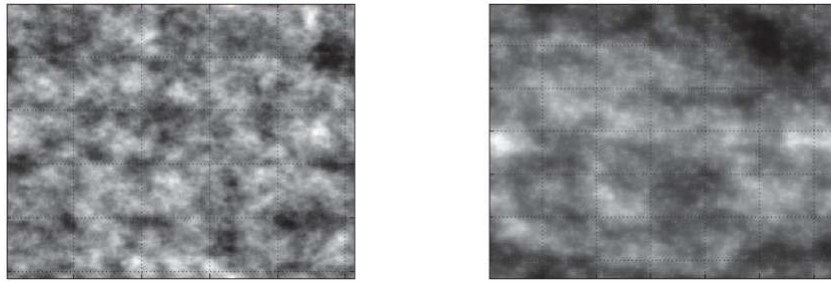


FIGURE 2.5 – Simulations de champs SAR pour différents bruits

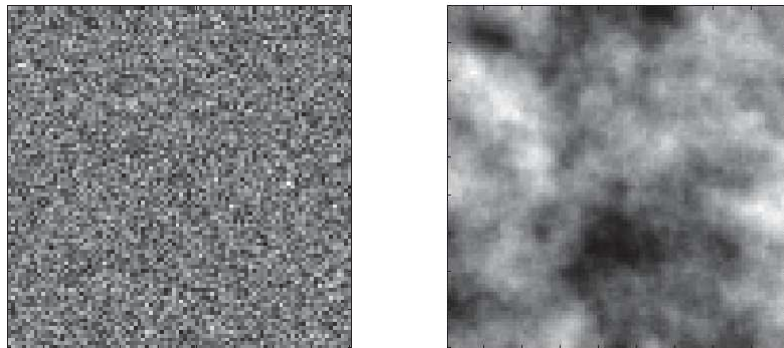
Cette matrice est bien symétrique. La condition sur les coefficients $(a_{k,l}, (k,l) \in N)$ assure qu'elle est définie positive [RH05].

Dans le cas d'un champ SAR isotrope aux quatre plus proches voisins (2.6), la condition sur a devient $|a| < \frac{1}{4}$ et la matrice A_a a la forme suivante

$$A_a = aW_4 \text{ où } W_4 = \begin{pmatrix} \ddots & & & & & & & & \\ & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ & & \ddots & & \ddots & & \ddots & & \ddots & \\ & & & \ddots & & \ddots & & \ddots & & \ddots \\ & & & & \ddots & & \ddots & & \ddots & \\ & & & & & \ddots & & \ddots & & \ddots \\ & & & & & & \ddots & & \ddots & \\ & & & & & & & \ddots & & \ddots \\ & & & & & & & & \ddots & \\ & & & & & & & & & \ddots \end{pmatrix} \quad (2.8)$$

Chaque ligne n'a que quatre coefficients non nuls.

Bien qu'il n'ait que deux paramètres, ce modèle permet une grande variabilité. La figure 2.6 montre deux réalisations de modèles SAR isotrope aux 4 plus proches voisins avec à gauche $a = 0$ et à droite $a = 0.249$. Ce sont des réalisations comme celle de droite qui nous ont motivés à utiliser les modèles SAR pour représenter les images de nuages.

FIGURE 2.6 – Exemples de réalisations de modèles SAR isotropes : à gauche avec $a = 0$ et à droite avec $a = 0.249$

Dans le cas d'un modèle SAR stationnaire directionnel (2.7), la condition sur les coefficients (a, b) est alors $|a| + |b| < \frac{1}{2}$ et la matrice A_a est de la forme

$$A_a = a_1W_{2,h} + a_2W_{2,v} \quad \text{où}$$

$$W_{2,h} = \begin{pmatrix} \ddots & & & & & & & & \\ & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ & & \ddots & & \ddots & & \ddots & & \ddots & \\ & & & \ddots & & \ddots & & \ddots & & \ddots \\ & & & & \ddots & & \ddots & & \ddots & \\ & & & & & \ddots & & \ddots & & \ddots \\ & & & & & & \ddots & & \ddots & \\ & & & & & & & \ddots & & \ddots \\ & & & & & & & & \ddots & \\ & & & & & & & & & \ddots \end{pmatrix} \quad \text{et} \quad W_{2,v} = \begin{pmatrix} \ddots & & & & & & & & & \\ & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ & & \ddots & & \ddots & & \ddots & & \ddots & & \\ & & & \ddots & & \ddots & & \ddots & & \ddots & \\ & & & & \ddots & & \ddots & & \ddots & & \\ & & & & & \ddots & & \ddots & & \ddots & \\ & & & & & & \ddots & & \ddots & & \\ & & & & & & & \ddots & & \ddots & \\ & & & & & & & & \ddots & & \\ & & & & & & & & & \ddots & \\ & & & & & & & & & & \ddots \end{pmatrix}$$

La figure 2.7 présente deux réalisations de modèles SAR stationnaires directionnels. Dans l'image de gauche, les coefficients valent $a_1 = 0$ et $a_2 = 0.48$. Pour l'image de droite, nous avons $a_1 = 0.48$ et $a_2 = 0$. L'aspect directionnel est très visible. Ainsi, en faisant varier la valeur des paramètres, nous pensons pouvoir modéliser aussi les nuages directionnels par exemple.

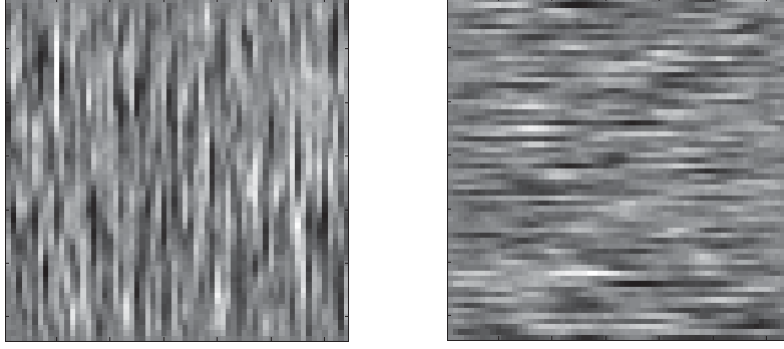


FIGURE 2.7 – Exemples de réalisations de modèles SAR stationnaire directionnels

2.2.2 Modèles auto-régressifs conditionnels

Le second modèle indirect qui nous a intéressés est le modèle Auto-Régressif Conditionnel, noté CAR.

Définition 3 (CAR stationnaire) Soit N une partie symétrique finie de \mathbb{Z}^2 . On dit qu'un champ \mathbf{Y} gaussien sur \mathbb{Z}^2 est un champ CAR stationnaire pour le voisinage N si

$$\forall (i, j) \in \mathbb{Z}^2, L(Y_{i,j} | Y_{k,l}, (k, l) \neq (i, j)) = N\left(\sum_{(k,l) \in N} a_{k,l} Y_{i-k, j-l}, \sigma^2\right) \quad (2.9)$$

où, pour tout couple (k, l) , $a_{k,l} = a_{-k, -l}$ et $\sum_{(k,l) \in N} |a_{k,l}| < 1$.

La notation $L(\mathbf{X} | \mathbf{Y})$ représente la loi conditionnelle de \mathbf{X} sachant \mathbf{Y} .

Ce modèle a été introduit par Besag en 1974 [Bes74] et la définition est issue de Rue et Held [RH05]. Il est dit *conditionnel* car chaque ligne permet de connaître la loi conditionnelle d'une variable. Il y d'autres informations données par ces lignes. Par exemple, pour un pixel (i, j) , la quantité $\sum_{(k,l) \in N} a_{k,l} Y_{i-k, j-l}$ est la meilleure prédiction au sens L^2 de $Y_{i,j}$.

Ce modèle est en fait très proche du modèle SAR présenté précédemment. Pour le voir, nous allons donner une définition équivalente des modèles CAR introduite par Guyon [Guy07].

Définition 4 Soit N une partie symétrique finie de \mathbb{Z}^2 . On dit qu'un champ \mathbf{Y} gaussien sur \mathbb{Z}^2 est un champ CAR stationnaire pour le voisinage N si

$$\forall (i, j) \in \mathbb{Z}^2, Y_{i,j} = \sum_{(k,l) \in N} a_{k,l} Y_{i-k, j-l} + \sigma \varepsilon_{i,j} \quad (2.10)$$

où pour tout couple (k, l) , $a_{k,l} = a_{-k, -l}$, $\sum_{(k,l) \in N} |a_{k,l}| < 1$ et pour tout couple (i, j) , $\varepsilon_{i,j}$ est décorrélé des variables $Y_{k,l}$ lorsque $(k, l) \neq (i, j)$.

Avec cette définition, la ressemblance avec le modèle SAR est plus marquée. Il y a toutefois des différences importantes à noter. Tout d'abord, le voisinage N d'un champ CAR doit être symétrique. D'autre part, le bruit ε est décorrélé pour un modèle SAR alors qu'il ne l'est pas pour un modèle CAR. Dans un modèle CAR, la matrice de covariance du champ ε est donnée par $(I - A_a)$. Ces différences se traduisent sur la matrice de précision du champ \mathbf{Y} . Si on note $Q_c(a, \sigma)$ la matrice de précision d'un champ CAR stationnaire, nous avons

$$Q_c(a, \sigma) = \frac{1}{\sigma^2} (I - A_a)$$

où A_a est la matrice définie en (2.8). Malgré ces différences, il existe un lien important entre les modèles SAR et les modèles CAR. C'est l'objet du résultat suivant énoncé par Guyon [Guy07].

Proposition 1 (Liens modèles SAR-modèles CAR)

1) Soit \mathbf{Y} un champ SAR stationnaire pour un voisinage N . Alors, \mathbf{Y} est aussi un champ CAR stationnaire pour un voisinage N' .

2) En dimension 2, la classe des champs CAR stationnaires est plus grande que la classe des champs SAR stationnaires.

Ce résultat se démontre en utilisant la densité spectrale du champ \mathbf{Y} . Reprenons l'exemple du champ SAR isotrope aux quatre plus proches voisins

$$\forall (i, j) \in \mathbb{Z}^2, Y_{i,j} = a(Y_{i-1,j} + Y_{i+1,j} + Y_{i,j-1} + Y_{i,j+1}) + \sigma \varepsilon_{i,j}.$$

En identifiant les densités spectrales, nous pouvons réécrire ce champ sous forme de champ CAR stationnaire

$$\begin{aligned} \forall (i, j) \in \llbracket 1, M \rrbracket^2, Y_{i,j} &= \frac{2a}{1 + 4a^2} (Y_{i-1,j} + Y_{i+1,j} + Y_{i,j-1} + Y_{i,j+1}) \\ &- \frac{a^2}{1 + 4a^2} (Y_{i-2,j} + Y_{i+2,j} + Y_{i,j+2} + Y_{i,j-2}) \\ &- \frac{2a^2}{1 + 4a^2} (Y_{i-1,j+1} + Y_{i+1,j+1} + Y_{i+1,j-1} + Y_{i-1,j-1}) + \frac{\sigma^2}{1 + 4a^2} \varepsilon_{i,j} \end{aligned} \quad (2.11)$$

Nous en concluons qu'un champ SAR isotrope aux 4 plus proches voisins est aussi un champ CAR stationnaire mais pour le voisinage formé des 12 plus proches voisins. La figure 2.8 présente les deux voisinages mis en jeu.



FIGURE 2.8 – Voisinages des 4 et des 12 plus proches voisins

Nous remarquons alors que le voisinage du champ CAR est une convolution du voisinage du champ SAR. Or, le voisinage est caractérisé par la position des coefficients non nuls dans la matrice de précision Q . Nous aurions donc pu faire cette remarque dès l'étude des matrices de précision de chacun des deux modèles. Pour un modèle SAR, cette matrice est de la forme $Q_s = \frac{1}{\sigma^2} {}^t(I - A_a)(I - A_a)$ où chaque ligne de la matrice A_a a seulement quatre coefficients non nuls correspondant aux quatre plus proches voisins. Le produit matriciel ${}^t(I - A_a)(I - A_a)$ donne donc une matrice Q_s dans laquelle chaque ligne possède douze coefficients non nuls correspondant aux douze plus proches voisins. La représentation équivalente sous forme de modèle CAR correspond à la même matrice de covariance. La matrice Q_c est donc une matrice de la forme $Q_c(a', \sigma') = \frac{1}{\sigma'} (I - A_{a'})$ qui correspond à un voisinage formé des douze plus proches voisins.

2.2.3 Intérêts des modèles SAR et CAR

Maintenant que nous savons que tout modèle SAR stationnaire peut être aussi représenté par un modèle CAR stationnaire, nous pouvons nous demander quel est l'intérêt d'utiliser l'une plutôt que l'autre des deux représentations pour un champ \mathbf{Y} sur \mathbb{Z}^2 .

En analysant des définitions de chacun des deux modèles, il semble que le modèle SAR soit défini plus simplement. En effet, dans le cas d'un modèle SAR, le bruit ε est un bruit gaussien blanc et la paramétrisation est plus simple. Nous allons voir qu'en terme de synthèse aussi, le modèle SAR est plus approprié que le modèle CAR. En revanche, en terme d'estimation ou de définition du voisinage, c'est le modèle CAR qui est le plus utile.

Méthodes de synthèse.

Contrairement aux modèles auto-régressifs en une dimension comme les champs ARMA par exemple, il n'y a pas toujours de représentation causale possible pour les modèles SAR-CAR. La synthèse d'un champ SAR stationnaire ou d'un champ CAR stationnaire ne peut donc pas se faire indice par indice comme c'est le cas en une dimension. D'autres méthodes sont toutefois disponibles. La première utilise la transformée de Fourier du champ. Elle convient lorsque le voisinage du champ est périodisé.

Repartons d'un champ SAR stationnaire périodisé.

$$\forall (i, j) \in \llbracket 1, M \rrbracket^2, Y_{i,j} = \sum_{(k,l) \in N} a_{k,l} Y_{i-k,j-l} + \sigma \varepsilon_{i,j}$$

Nous pouvons calculer la transformée de Fourier de cette expression. En notant $\hat{Y}_{i,j}$ la transformée de Fourier du champ \mathbf{Y} au point (i, j) , nous obtenons

$$\begin{aligned} \forall (i, j) \in \llbracket 1, M \rrbracket^2, \hat{Y}_{i,j} &= \sum_{(k,l) \in N} a_{k,l} \hat{Y}_{i-k,j-l} + \sigma \hat{\varepsilon}_{i,j} \\ &= \sum_{(k,l) \in N} a_{k,l} \hat{Y}_{i,j} e^{-2i\pi \frac{k}{M}} e^{-2i\pi \frac{l}{M}} + \sigma \hat{\varepsilon}_{i,j} \end{aligned}$$

Nous obtenons alors une expression pour la transformée de Fourier pour tous les points de la grille $\llbracket 1, M \rrbracket^2$

$$\forall (i, j) \in \llbracket 1, M \rrbracket^2, \hat{Y}_{i,j} = \frac{\sigma \hat{\varepsilon}_{i,j}}{1 - \sum_{(k,l) \in N} a_{k,l} \exp(-2i\pi \frac{k}{M}) \exp(-2i\pi \frac{l}{M})} \quad (2.12)$$

En pratique, cela permet de synthétiser une réalisation d'un champ SAR stationnaire de paramètres (a, σ) .

- On synthétise le bruit blanc ε
- On calcule sa transformée de Fourier discrète $\hat{\varepsilon}$
- On calcule $\hat{\mathbf{Y}}$ par la formule (2.12)
- On lui applique la transformée de Fourier inverse pour obtenir une réalisation d'un modèle SAR de paramètres a et σ .

Dans le cas d'un champ de voisinage non périodisé, il faut appliquer une autre méthode de synthèse, une synthèse matricielle. Pour un champ SAR stationnaire de paramètres a et σ ,

- Calcul de la matrice de covariance $\Sigma(a, \sigma) = Q_s^{-1}(a, \sigma)$.
- Calcul de $Z(a, \sigma)$ la décomposition de Cholesky de la matrice $\Sigma(a, \sigma)$.
- Simulation d'un bruit blanc ε .
- $\mathbf{Y} = Z(a, \sigma) \cdot \varepsilon$ est alors une réalisation d'un champ SAR stationnaire de paramètres a et σ .

Ces deux méthodes de synthèse sont plus adaptées aux champs SAR qu'aux champs CAR. En effet, dans le cas d'un modèle CAR, il faut alors remplacer la synthèse d'un bruit blanc par la synthèse d'un bruit de matrice de covariance $(I - A_a)$.

Ainsi, en terme de simulation, le modèle SAR semble plus pertinent que le modèle CAR.

Estimation des paramètres

Dans le cadre de la régression linéaire, la méthode la plus couramment utilisée est la méthode des moindres carrés ordinaires.

Appliquée aux modèles SAR et CAR stationnaires, cette méthode consiste à minimiser la quantité suivante

$$S(a) = \sum_{i,j=1}^M \left(Y_{i,j} - \sum_{(k,l) \in N} a_{k,l} Y_{i-k,j-l} \right)^2.$$

En dérivant par rapport à chacune des variables $a_{k,l}$ pour $(k,l) \in N$, nous obtenons un système de $\text{card}(N)$ équations à $\text{card}(N)$ inconnues. Nous pouvons donc le résoudre et ainsi obtenir l'estimateur des moindres carrés ordinaires.

Par exemple, dans le cas d'un modèle SAR isotrope aux 4 plus proches voisins défini en (2.6), l'estimateur par moindre carré ordinaire du coefficient a est donné par [Guy07]

$$\hat{a}_{MCO} = \frac{\sum_{i,j=1}^M (Y_{i-1,j} + Y_{i+1,j} + Y_{i,j-1} + Y_{i,j+1}) Y_{i,j}}{\sum_{i,j=1}^M (Y_{i-1,j} + Y_{i+1,j} + Y_{i,j-1} + Y_{i,j+1})^2}$$

Si nous utilisons le modèle CAR associé défini en (2.11), la quantité à minimiser est alors

$$\begin{aligned} C(a) = \sum_{i,j=1}^M \left(Y_{i,j} - \frac{2a}{1+4a^2} (Y_{i-1,j} + Y_{i+1,j} + Y_{i,j-1} + Y_{i,j+1}) \right. \\ \left. - \frac{a^2}{1+4a^2} (Y_{i-2,j} + Y_{i+2,j} + Y_{i,j+2} + Y_{i,j-2}) \right. \\ \left. - \frac{2a^2}{1+4a^2} (Y_{i-1,j+1} + Y_{i+1,j+1} + Y_{i+1,j-1} + Y_{i-1,j-1}) \right)^2. \end{aligned}$$

Son expression est plus compliquée que celle de $S(a)$. Toutefois, l'estimateur des moindres carrés ordinaires est consistant dans le cas d'un modèle CAR mais il ne l'est pas dans le cas d'un modèle SAR [Guy07]. Ce résultat est dû à l'indépendance (ou non) de chacun des résidus $Y_{i,j} - \sum_{(k,l) \in N} \hat{a}_{k,l} Y_{i-k,j-l}$ avec les variables $Y_{s,t}$ pour $(s,t) \neq (i,j)$.

Ainsi, dans le cas d'un modèle SAR isotrope aux quatre plus proches voisins, il vaudra mieux utiliser comme estimateur celui associé au modèle CAR.

Pour l'estimation des paramètres, la représentation sous forme de modèle CAR est donc plus appropriée.

Représentation sous forme de champs de Markov

Il existe une dernière raison qui justifie l'utilisation de la représentation CAR plutôt que la représentation SAR pour un champ \mathbf{Y} . Pour la comprendre, nous avons besoin de la notion de champ de Markov gaussien stationnaire.

Définition 5 (Champ de Markov gaussien stationnaire) *Un champ gaussien \mathbf{Y} sur \mathbb{Z}^2 est un champ de Markov gaussien stationnaire pour le voisinage N si*

$$\forall (i,j) \in \mathbb{Z}^2, L(Y_{i,j} | Y_{k,l}, (k,l) \neq (i,j)) = L(Y_{i,j} | Y_{(k,l)}, (k,l) \in (i,j) + N)$$

où $L(A|B)$ représente la loi conditionnelle de A sachant B .

Cela signifie que, sachant la valeur du champ \mathbf{Y} à tous les pixels autres que (i,j) , la loi du champ \mathbf{Y} au pixel (i,j) ne dépend en fait que d'un ensemble restreint de pixels. Cet ensemble est appelé *le voisinage du champ*. On dit alors que deux pixels (i,j) et (k,l) sont voisins si $(i-k, j-l) \in N$. Ces champs markoviens sont très souvent utilisés pour modéliser le contenu des images I [CJ83].

Pour \mathbf{Y} un champ de Markov gaussien stationnaire, le voisinage N se lit directement sur la matrice de précision. Deux pixels (i,j) et (k,l) seront voisins si, et seulement si, $\Sigma^{-1}((i,j), (k,l)) \neq 0$.

Pour un modèle CAR de voisinage N , la matrice de précision est de la forme $Q = \frac{1}{\sigma^2} (I - A_a)$ où la matrice A_a est telle que $A_a((i,j), (i-k, j-l)) = 0$ si, et seulement si, $(k,l) \in N$. Ainsi, pour les modèles CAR aussi, le voisinage se lit sur la matrice de précision. Les ressemblances entre les modèles CAR et les modèles de Markov ne s'arrêtent pas là puisqu'en fait, un champ CAR de voisinage N est un champ de Markov pour ce même voisinage N .

Puisque tous les modèles SAR sont des modèles CAR, nous en déduisons que les modèles SAR sont, eux aussi, des champs de Markov. Toutefois, le voisinage n'est pas celui associé à la représentation sous forme de modèle SAR mais celui associé à la représentation sous forme de modèle CAR. Par exemple, le modèle SAR isotrope aux quatre plus proches voisins est un champ de Markov. Son voisinage n'est pas celui formé des quatre plus proches voisins mais celui formé des douze plus proches voisins.

La représentation sous forme de modèle CAR est donc celle qui permet de définir le plus simplement le champ de Markov associé.

Dans les paragraphes précédents, nous avons étudiés des modèles indirects particuliers. Ces modèles sont tous paramétriques et nous allons nous intéresser à l'estimation de leurs paramètres.

2.2.4 Estimation des paramètres par maximum de vraisemblance

Soit S un modèle paramétrique indirect, comme le modèle SAR ou le modèle CAR par exemple. Nous supposons que l'image I suit le modèle S . Nous notons θ_S les paramètres à estimer. Le modèle S est alors un modèle gaussien de moyenne nulle et de matrice de précision $Q(\theta_S)$.

Dans nos applications, nous souhaitons estimer les paramètres par maximum de vraisemblance

$$\hat{\theta}_S(I) = \operatorname{argmax}_{\theta_S} \frac{\sqrt{\det(Q(\theta_S))}}{\sqrt{2\pi}^{M^2}} \exp\left(-\frac{1}{2} {}^t I Q(\theta_S) I\right).$$

Cet estimateur est non biaisé et consistant. Même si nous travaillons avec peu de données, il est intéressant de travailler avec un estimateur consistant.

Il n'y a pas de formule générale pour l'estimateur du maximum de vraisemblance $\hat{\theta}_S$. Cela dépend bien sûr du modèle S . En réalité, même une fois fixé le modèle S , il n'existe pas toujours de formule exacte pour $\hat{\theta}_S$. Prenons l'exemple d'un modèle SAR isotrope aux quatre plus proches voisins comme en (2.6). Dans ce cas, $\theta_S = (a, \sigma)$ et la densité $f_S(x|\theta_S)$ a la forme suivante

$$f_S(\mathbf{x}|a, \sigma) = \frac{\det(\operatorname{Id} - A_a)}{(\sqrt{2\pi}\sigma)^{M^2}} \exp\left(-\frac{1}{2\sigma^2} {}^t \mathbf{x} (\operatorname{Id} - A_a)^2 \mathbf{x}\right)$$

La présence du déterminant $\det(\operatorname{Id} - A_a)$ nous empêche de déduire une formule exacte pour $\hat{\theta}_S$. Dans le cas d'images multispectrales, Bennett et Khotanzad [BK99] ont proposé un algorithme d'estimation du maximum de vraisemblance. Dans nos travaux, nous ne travaillons qu'avec des images en niveau de gris et nous allons proposer une estimation adaptée aux modèles indirects étudiés.

Revenons à notre modèle S initial. La recherche de l'estimateur du maximum de vraisemblance $\hat{\theta}_S$ se fait alors de façon itérative. Nous partons d'une valeur initiale θ_S^0 . A chaque itération i , nous cherchons une valeur θ_S^{i+1} , proche de θ_S^i telle que $f_S(I|\theta_S^{i+1}) > f_S(I|\theta_S^i)$. La recherche s'arrête soit lorsqu'on a atteint le nombre initial d'itérations souhaitées soit lorsque l'écart $f_S(X, \theta_S^{i+1}) - f_S(X, \theta_S^i)$ a atteint un seuil initialement fixé. Cette recherche itérative est coûteuse en temps de calcul puisqu'il faut évaluer souvent les quantités $\det(Q_S(\theta))$ et ${}^t X Q_S(\theta) X$. Nous essayons alors, modèle par modèle, d'accélérer ces évaluations. Par exemple, si nous reprenons le modèle SAR isotrope aux quatre plus proche voisins, nous pouvons exprimer autrement le déterminant. Pour ce modèle, nous avons vu que la matrice A_a est de la forme $A_a = aW_4$. Si nous notons $(e_i)_{1 \leq i \leq M^2}$ les valeurs propres de la matrice W_4 , nous pouvons alors exprimer autrement la vraisemblance d'une image I sous la forme suivante

$$f_S(I|a, \sigma) = \frac{1}{(\sqrt{2\pi})^{M^2}} \prod_{i=1}^{M^2} (1 - ae_i) \exp\left(-\frac{1}{2\sigma^2} ({}^t I I + 2a {}^t I W_4 I - a^2 {}^t I W_4^2 I)\right)$$

Les valeurs propres $(e_i)_{1 \leq i \leq M^2}$ ainsi les quantités ${}^t I I$, ${}^t I W_4 I$ et ${}^t I W_4^2 I$ sont calculées une seule fois au début de la recherche du maximum de vraisemblance. Nous avons alors pu programmer la recherche itérative de l'estimateur du maximum de vraisemblance en un temps qui est devenu raisonnable. Toutefois, ce temps est encore trop long pour les applications en temps réel visées par TOSA.

Nous avons vu précédemment que pour des modèles S définis linéairement, comme les modèles SAR ou CAR, il pouvait être naturel d'utiliser l'estimateur des moindres carrés ordinaires.

$$\tilde{\theta}_S = \operatorname{argmin}_{\theta} {}^t I Q_S(\theta) I.$$

Dans ce cas, l'évaluation du déterminant $\det(Q(\theta))$ ne pose plus de problème et le temps de calcul est très rapide. Nous allons alors comparer les valeurs obtenues par l'estimateur des moindres carrés ordinaires et par l'estimateur du maximum de vraisemblance pour un modèle SAR isotrope aux quatre plus proches voisins. Pour l'estimateur des moindres carrés ordinaires, nous avons utilisé le modèle CAR équivalent afin d'obtenir un estimateur consistant.

La figure 2.9 compare l'estimateur des moindres carrés ordinaires et celui du maximum de vraisemblance dans le cas d'un modèle SAR isotrope aux quatre plus proches voisins lorsque la taille de l'échantillon augmente. Pour ce modèle, nous avons une formule exacte pour l'estimateur des moindres carrés ordinaires et la recherche itérative du maximum de vraisemblance est assez rapide.

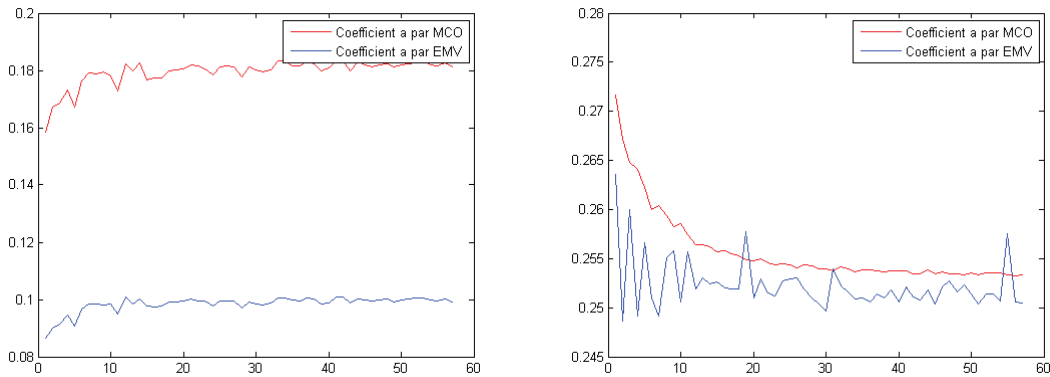


FIGURE 2.9 – Comparaison des estimateurs du maximum de vraisemblance et des moindres carrés ordinaires pour $a = 0.1$ et $a = 0.249$.

Nous retrouvons bien le caractère biaisé de l'estimateur des moindres carrés ordinaires. Ce biais dépend du paramètre a qui est inconnu. Nous avons donc décidé de travailler dans la suite de nos travaux avec l'estimation itérative du maximum de vraisemblance.

Les modèles SAR et CAR présentés sont deux cas particuliers de modèles indirects, c'est-à-dire pour lesquels la structure de covariance est imposée sur la matrice de précision. Cette matrice de précision est souvent parcimonieuse comme pour le modèle SAR isotrope aux quatre plus proches voisins par exemple. La matrice de covariance correspondante, comme inverse d'une matrice parcimonieuse, a donc très peu de coefficients nuls. On dit que le champ \mathbf{Y} a une covariance à longue portée.

2.3 Modèles directs

Nous avons introduit ensuite les modèles directs dans le but de modéliser les situations opposées, c'est-à-dire lorsque la matrice de covariance a beaucoup de coefficients nuls.

2.3.1 Exemples de modèles directs

Définition 6 (Modèle directs) Soit \mathbf{Y} un champ gaussien sur une grille $\llbracket 1, M \rrbracket^2$. On dit que \mathbf{Y} suit un modèle gaussien direct si sa matrice de covariance est de la forme suivante

$$\Sigma \in \mathcal{R} = \left\{ \sigma^2 \cdot Id + \sum_{i=1}^{d-1} c_i T_i \right\}$$
 où les matrices T_i sont des matrices ne possédant que deux diagonales de 1, c'est-à-dire de la forme

En utilisant la commutativité de la trace, nous pouvons réécrire le deuxième terme

$${}^t I \Sigma^{-1} I = \text{tr}({}^t I \Sigma^{-1} I) = \text{tr}(\Sigma^{-1} I {}^t I) = \text{tr}(\Sigma^{-1} S_e)$$

où S_e est la matrice de covariance empirique $S_e = I {}^t I$. La fonction que nous cherchons à maximiser s'écrit donc

$$g(S_e, \Sigma) = -\log(\det(\Sigma)) - \frac{1}{2} \text{tr}(\Sigma^{-1} S_e).$$

Comme c'est souvent le cas dans les méthodes du maximum de vraisemblance, nous avons cherché un zéro de la différentielle de la fonction à maximiser.

$$\begin{aligned} \forall R \in GL_n(\mathbb{R}), \forall H \in \mathcal{R}_m, D_R(\log(\det))(H) &= \text{tr}(R^{-1} H) \\ D_R(\text{inv})(H) &= \text{tr}(R^{-1} H R^{-1}) \end{aligned}$$

Par linéarité de la trace, nous obtenons alors

$$\forall H \in \mathcal{R}_m, D_\Sigma(g(S_e, \cdot))(H) = \text{tr}[(\Sigma^{-1} S_e \Sigma^{-1} - \Sigma^{-1}) H]$$

et l'estimateur du maximum de vraisemblance est alors un zéro de la différentielle sur \mathcal{R}_m . Enfin, puisque la structure \mathcal{R}_m est une structure linéaire, il suffit alors de résoudre le système suivant

$$\hat{\Sigma}_m \in \mathcal{R}_m \text{ et } \forall i \in \llbracket 1, d \rrbracket, \text{tr}[(\hat{\Sigma}_m^{-1} S_e \hat{\Sigma}_m^{-1} - \hat{\Sigma}_m^{-1}) T_i] = 0.$$

où $T_d = \text{Id}$. Toutefois, il n'existe pas de formule exacte pour résoudre ce système et nous allons procéder de manière itérative.

Pour l'initialisation, nous pouvons utiliser la matrice $\hat{\Sigma}_{m,0} = S_e$. Cette matrice n'a pas la structure \mathcal{R}_m souhaitée mais elle vérifie les d égalités précédentes.

Notons ensuite $\hat{\Sigma}_{m,p}$ la matrice obtenue après la $p^{\text{ème}}$ itération. Nous construisons la matrice $\hat{\Sigma}_{m,p+1}$ en résolvant le système suivant

$$\forall i \in \llbracket 1, d \rrbracket, \text{tr}[(\hat{\Sigma}_{m,p}^{-1} S \hat{\Sigma}_{m,p}^{-1} - \hat{\Sigma}_{m,p}^{-1} \hat{\Sigma}_{m,p+1} \hat{\Sigma}_{m,p}^{-1}) T_i] = 0.$$

Une idée de preuve pour justifier cette construction est présentée dans l'appendice A. Ce n'est pas une définition explicite de la matrice $\hat{\Sigma}_{m,p+1}$ en fonction de la matrice $\hat{\Sigma}_{m,p}$ mais ce système est lui solvable.

La matrice $\hat{\Sigma}_{m,p+1}$ doit appartenir à la structure \mathcal{R}_m . Elle est donc de la forme

$$\hat{\Sigma}_{m,p+1} = \sum_{j=1}^d c_{j,p+1} T_j$$

En substituant cette écriture dans le système précédent, nous obtenons

$$\forall i \in \llbracket 1, d \rrbracket, \sum_{j=1}^d c_{j,p+1} \text{tr}[\hat{\Sigma}_{m,p}^{-1} T_j \hat{\Sigma}_{m,p}^{-1} T_i] = \text{tr}[\hat{\Sigma}_{m,p}^{-1} S \hat{\Sigma}_{m,p}^{-1} T_i] \quad (2.14)$$

Cela devient alors un système linéaire de d équations à d inconnues que l'on peut résoudre. Le vecteur solution $(\hat{c}_{i,p+1})_{1 \leq i \leq d}$ permet d'obtenir la matrice $\hat{\Sigma}_{m,p+1} = \sum_{i=1}^d \hat{c}_{i,p+1} T_i$. Nous sommes alors sûrs que la matrice estimée a bien la structure souhaitée. Rien ne nous garantit en revanche qu'elle soit bien définie positive (ce qui est indispensable pour une matrice de covariance) et nous devons donc nous en assurer lors de la mise en place de l'algorithme. Finalement, la procédure générale est donc la suivante

1. A l'initialisation, $\hat{\Sigma}_{m,0} = S_e$.
2. Pour chaque itération p , le système (2.14) permet de définir $\hat{\Sigma}_{m,p+1}$ à partir de $\hat{\Sigma}_{m,p}$.
 - (a) Si $g(S, \hat{\Sigma}_{m,p+1}) > g(S, \hat{\Sigma}_{m,p})$ et si $\hat{\Sigma}_{m,p+1}$ est une matrice définie positive, on retourne à l'étape 2 pour l'itération $p+1$.

- (b) Si une des deux conditions n'est pas vérifiée, on pose $R' = \hat{\Sigma}_{m,p} + q(\hat{\Sigma}_{m,p} - \hat{\Sigma}_{m,p+1})$ avec $q = \frac{1}{2}$. On divise par 2 la valeur de q jusqu'à obtenir une matrice R' qui vérifie les deux conditions de l'étape 2. On retourne ensuite à l'étape 2 avec $\hat{\Sigma}_{m,p+1} = R'$.

Nous arrêtons l'algorithme après N itérations. Cet algorithme permet d'obtenir une estimation de la matrice de covariance qui soit définie positive et qui ait la structure souhaitée. Nous l'avons utilisé sur quelques exemples.

Exemple. Reprenons une matrice Σ qui a la même structure que celle présentée en (2.13) avec comme coefficients $\sigma^2 = 20.3$, $\gamma = -7$ et $\alpha = -3.3$. Ces valeurs garantissent que la matrice soit bien définie positive. Soit \mathbf{Y} un champ gaussien sur la grille $\llbracket 1, 16 \rrbracket^2$ de matrice de covariance Σ . La figure 2.10 présente les valeurs obtenues pour les estimations des trois coefficients aux différentes itérations de l'algorithme.

N	$\hat{\sigma}^2$	$\hat{\gamma}$	$\hat{\alpha}$
	20.3	-7	-3.3
1	21.9636	-6.8290	-1.9274
2	22.1618	-7.0112	-3.7022
3	22.2583	-6.9826	-3.9593
4	22.2747	-6.9540	-4.0198
5	22.2815	-6.9542	-4.0285
6	22.2823	-6.9531	-4.0309
7	22.2826	-6.9531	-4.0312
8	22.2826	-6.9531	-4.0313
9	22.2826	-6.9531	-4.0313
10	22.2826	-6.9531	-4.0313

FIGURE 2.10 – Estimation des paramètres σ^2 , γ et α aux différentes itérations.

Nous pouvons d'abord remarquer que les valeurs obtenues ne sont pas très bonnes. Il faut cependant noter que nous n'avons utilisé qu'un échantillon de taille 1 pour estimer ces coefficients et que finalement, les résultats sont corrects pour les applications souhaitées.

L'autre remarque importante à faire est qu'il semble y avoir une convergence des valeurs à partir de $N = 5$. Nous avons noté le même comportement pour d'autres structures. Dans toute la suite, nous avons donc choisi de n'effectuer que $N = 5$ itérations lors de l'utilisation de l'algorithme.

Dans les deux paragraphes précédents, nous avons présenté deux classes de modèles stationnaires ainsi que des outils pour estimer leurs paramètres par maximum de vraisemblance. Ces modèles ont été introduits pour modéliser les images réelles fournies par TOSA.

Toutefois, l'image réelle visible sur la figure 2.2 est loin de ressembler *globalement* à la réalisation d'un champ SAR aux quatre plus proches voisins présentée à la figure 2.6. Les modèles proposés semblent mieux adaptés pour modéliser *localement* le comportement des images réelles. Nous avons donc étendu les notions précédentes pour définir des modèles non plus stationnaires mais simplement *stationnaires par morceaux*.

2.4 Modèles stationnaires par morceaux

Dans l'image présentée à la figure 2.2, c'est en réalité l'hypothèse de stationnarité globale qui est mise en défaut. En effet, le comportement du pixel (5,5) vis-à-vis de ses voisins n'est pas le même que celui du pixel (155,400) vis-à-vis des siens. En revanche, le champ \mathbf{Y} peut être supposé localement stationnaire. Dahlhaus [Dah12] a étudié les processus de dimension 1 localement stationnaires. Nous souhaitons segmenter l'image I en zones dans lesquelles nous pourrions supposer le champ stationnaire. En dimension un, ce problème a déjà été abordé par Chambaz [Cha02] pour détecter des changements dans la moyenne du signal, par Lavielle [Lav98], Lavielle et Ludena [LL00] ou encore Basseville et Nikiforov [BN93] lorsqu'il s'agit de changements de loi. Ils cherchent à détecter

de possibles ruptures dans la valeur des paramètres d'un signal en une dimension. En dimension deux, [Lee98] a proposé un algorithme pour segmenter des images. Galland et coauteurs [GBR03] ont également proposé un algorithme itératif pour segmenter des images de nuages. Dans tous les cas, les modèles sous-jacents sont des modèles non stationnaires. Kleiber et Nychka [KN12] ont proposé d'utiliser des modèles de Matern pour la matrice de covariance. Dans nos travaux, pour effectuer la segmentation de nos images en zones stationnaires, nous allons étendre les modèles stationnaires présentés précédemment à des modèles stationnaires par morceaux.

2.4.1 Modèles SAR non stationnaires

Reprenons les modèles SAR stationnaires définis au paragraphe 2.2.1. Il existe plusieurs façons d'étendre cette définition pour obtenir un modèle non stationnaire. Tout d'abord, nous pouvons supposer que le voisinage N n'est plus le même pour tous les pixels de la grille $\llbracket 1, M \rrbracket^2$. La notation $N(i, j)$ représente le voisinage du pixel (i, j) .

$$\forall (i, j) \in \llbracket 1, M \rrbracket^2, Y_{i,j} = \sum_{(k,l) \in N(i,j)} a_{k,l} Y_{i-k,j-l} + \sigma \varepsilon_{i,j}$$

Nous pouvons aussi supposer que le voisinage est toujours le même pour chaque pixel mais qu'en revanche la valeur des coefficients $(a_{k,l}, (k,l) \in N)$ dépend du pixel (i, j) . Nous les notons alors $(a_{k,l}^{i,j}, (k,l) \in N)$.

$$\forall (i, j) \in \llbracket 1, M \rrbracket^2, Y_{i,j} = \sum_{(k,l) \in N} a_{k,l}^{i,j} Y_{i-k,j-l} + \sigma \varepsilon_{i,j}$$

Enfin, en faisant varier à la fois le voisinage et la valeur des coefficients, nous obtenons la définition d'un modèle SAR non stationnaire. Nous donnons la définition pour un champ défini sur \mathbb{Z}^2 [Guy07]. Elle s'adapte comme précédemment au cas d'un champ de taille finie.

Définition 7 (SAR non stationnaire) Soit $N = (N(i, j), (i, j) \in \mathbb{Z}^2)$ une famille de parties finies de \mathbb{Z}^2 . On dit qu'un champ \mathbf{Y} gaussien sur \mathbb{Z}^2 est un champ SAR pour la famille de voisinage N si

$$\forall (i, j) \in \mathbb{Z}^2, Y_{i,j} = \sum_{(k,l) \in N(i,j)} a_{k,l}^{i,j} Y_{i-k,j-l} + \sigma \varepsilon_{i,j}$$

où $\forall (i, j) \in \mathbb{Z}^2, \sum_{(k,l) \in N(i,j)} |a_{k,l}^{i,j}| < 1$ et ε est un bruit blanc gaussien.

Dans le cas d'un modèle fini de taille M , le nombre de paramètres à estimer est alors majoré par $\sum_{i,j=1}^M N(i, j)$. Puisque nous travaillons avec une seule image, cela fait beaucoup trop de paramètres à estimer pour peu de données.

Nous nous sommes donc intéressés à des modèles non stationnaires particuliers, ayant moins de paramètres à estimer. Pour cela, nous avons étudié plus en détails les images réelles fournies par TOSA. La figure 2.2 est représentative de la base d'images réelles de TOSA. Elles sont composées de zones qui sont de différentes textures : ciel bleu, nuages épais, nuages directionnels, ... Dans chacune des ces textures, l'hypothèse de stationnarité est raisonnable.

Nous avons donc supposé que les images étaient *stationnaires par morceaux* : la restriction de l'image I à chacune des zones (des textures) est la réalisation d'un champ spatial stationnaire. Ce modèle a déjà été proposé et utilisé par Kim, Mallick et Holmes [KMH05] dans un cadre de segmentation de cartes de sols pour en étudier la perméabilité.

2.4.2 Définition des modèles gaussiens finis stationnaires par morceaux

Dans ce paragraphe, I représente toujours une image finie de taille M . Une fois recentrée, c'est une réalisation d'un champ gaussien \mathbf{Y} sur la grille $\llbracket 1, M \rrbracket^2$. Avant de donner une définition exacte, nous allons expliquer ce que nous entendons par *finis stationnaires par morceaux*.

Par morceaux : Soit $R = (R_1, \dots, R_P)$ une partition de taille P de la grille $\llbracket 1, M \rrbracket^2$. Nous nous intéressons aux restrictions du champ \mathbf{Y} à chacune des zone R_i pour $i \in \{1, \dots, P\}$. Nous notons ces différentes restrictions $\mathbf{Y}_{R_1}, \dots, \mathbf{Y}_{R_P}$. S'il n'y a aucune ambiguïté sur la partition R utilisée, nous les noterons plus simplement $\mathbf{Y}_1, \dots, \mathbf{Y}_P$.

\mathbf{Y}_i est alors un champ gaussien sur la zone R_i .

Stationnaire fini : Soit S un modèle stationnaire défini pour un champ \mathbf{Y} sur \mathbb{Z}^2 . Soit R une zone de la grille $\llbracket 1, M \rrbracket^2$. La notation $\#(R)$ représente le nombre de pixels contenus dans la zone R . Nous notons $S(R)$ la restriction du modèle S à la zone R . Cela signifie que nous avons réglé les problèmes de bords liés au passage de l'espace \mathbb{Z}^2 à la zone R . Par exemple, pour le cas du modèle SAR stationnaire, nous avons choisi de ne pas périodiser le voisinage lors de la restriction à une zone de taille finie.

Pour caractériser ce modèle fini $S(R)$, nous aurons besoin de connaître sa matrice de covariance Σ de taille $\#(R) \times \#(R)$.

Nous pouvons maintenant donner la définition d'un modèle fini stationnaire par morceaux.

Définition 8 (Modèle stationnaire fini par morceaux) Soit \mathbf{Y} un champ gaussien centré sur la grille $\llbracket 1, M \rrbracket^2$. On dit que \mathbf{Y} suit un modèle fini stationnaire par morceaux s'il existe une partition $R = (R_1, \dots, R_P)$ de $\llbracket 1, M \rrbracket^2$ et une liste de modèles (S_1, \dots, S_P) stationnaires sur \mathbb{Z}^2 telles que

$\forall i \in \llbracket 1, P \rrbracket$, la restriction du champ \mathbf{Y} à la zone R_i , notée \mathbf{Y}_{R_i} , suit le modèle stationnaire fini $S_i(R_i)$.

Puisque \mathbf{Y} est un vecteur gaussien centré, nous pouvons caractériser sa loi par une matrice Σ . Cette matrice a pour blocs diagonaux les différentes matrices Σ_i , matrice de covariance du champ \mathbf{Y}_{R_i} .

La figure 2.11 présente un exemple de modèle stationnaire fini par morceaux pour une partition en quatre rectangles et pour des modèles stationnaires sur \mathbb{Z}^2 notés (S_1, \dots, S_4) . La matrice de covariance correspondante Σ est aussi représentée.

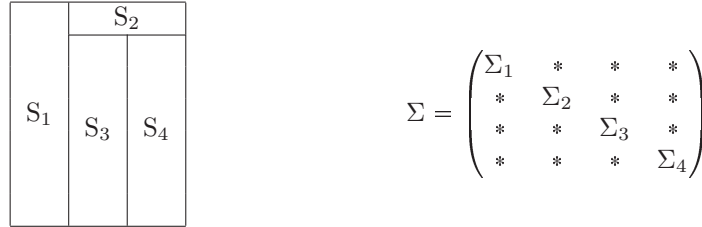


FIGURE 2.11 – Modèle SAR fini stationnaire par morceaux et sa matrice de covariance

Nous voyons alors que la définition 8 ne permet pas de définir entièrement la matrice Σ et donc pas non plus la loi de \mathbf{Y} . En effet, nous devons compléter les blocs non diagonaux. La matrice Σ est une matrice de covariance, elle doit donc être inversible et définie positive. Ce sont les seules contraintes pour remplir les blocs non diagonaux.

Le choix le plus naturel consiste à compléter les blocs * par des blocs de 0. Σ sera toujours inversible et définie positive et diagonale par blocs.

Ainsi, nous avons décidé d'imposer la complétion de la matrice par des blocs de 0, ce qui nous conduit à une définition complète d'un champ gaussien fini stationnaire par morceaux.

Définition 9 (Modèle fini strictement stationnaire par morceaux) Soit \mathbf{Y} un champ gaussien centré sur la grille $\llbracket 1, M \rrbracket^2$. On dit que \mathbf{Y} suit un modèle fini strictement stationnaire par morceaux s'il existe une partition $R = (R_1, \dots, R_P)$ de $\llbracket 1, M \rrbracket^2$ et une liste de modèles S_1, \dots, S_P stationnaires sur \mathbb{Z}^2 telles que

$\forall i \in \llbracket 1, P \rrbracket$, la restriction du champ \mathbf{Y} à la zone R_i , notée \mathbf{Y}_{R_i} , suit le modèle stationnaire fini $S_i(R_i)$.

Les restrictions $(\mathbf{Y}_{R_1}, \dots, \mathbf{Y}_{R_P})$ sont des champs indépendants, c'est-à-dire que les blocs non diagonaux de Σ sont des blocs de 0.

La partition R appartient à l'ensemble des partitions de la grille $\llbracket 1, M \rrbracket^2$. Les modèles (S_1, \dots, S_P) sont tous des modèles stationnaires. Ils peuvent être directs ou indirects comme présentés précédemment. Les situations souvent rencontrées sont les suivantes

- C'est le même modèle S dans chacune des zones : $\forall i \in \llbracket 1, P \rrbracket, S_i = S$.
- Chaque modèle S_i est un modèle SAR stationnaire mais leur voisinage $(N_i, i \in \llbracket 1, P \rrbracket)$ sont différents.

La figure 2.12 représente plusieurs réalisations de modèles SAR finis et stationnaires par morceaux.

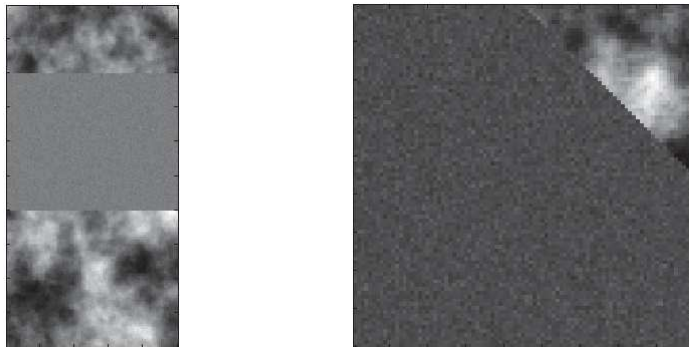
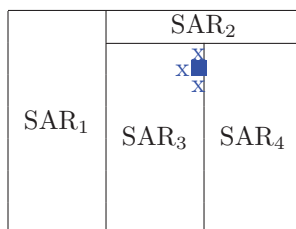


FIGURE 2.12 – Réalisations de modèles SAR finis et stationnaires par morceaux

Le modèle que nous retrouverons lors de la construction de l'algorithme de détection dans le chapitre 4 est celui pour lequel la partition R correspond à une segmentation de l'image en carré de taille 16×16 . Dans chacune des zones, le modèle supposé est un modèle direct aux quatre plus proches voisins comme en (2.13). Ce modèle est bien un modèle gaussien fini strictement stationnaire par morceaux.

Le choix de blocs de 0 pour compléter la matrice Σ entraîne l'indépendance des restrictions du champ \mathbf{Y} dans chaque zone de la partition. Dans le cas d'une image de ciel bleu et de nuages, des considérations physiques tendent à appuyer l'hypothèse d'indépendance entre la zone de ciel et la zone de nuages. Ce choix permet aussi de résoudre les problèmes de frontière apparus suite à l'hypothèse de stationnarité par morceaux. Reprenons l'exemple d'un modèle SAR par morceaux avec dans chacune des zones le voisinage formé des quatre plus proches voisins.

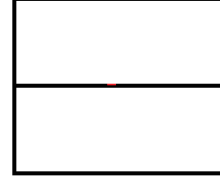


Le pixel représenté par un carré bleu n'est pas aux bords de l'image. Ses quatre plus proches voisins sont donc bien définis. Pourtant, suite à l'hypothèse d'indépendance entre les restrictions, son voisinage ne sera composé que des trois voisins appartenant à la même zone que lui (croix bleues).

Exemple de complétion pour un modèle SAR fini stationnaire par morceaux

Ce modèle est un modèle indirect et nous allons donc compléter la matrice de précision $Q = \Sigma^{-1}$ plutôt que la matrice Σ directement. Le matrice Q a la forme suivante

Nous allons proposer une autre complétion que celle avec des blocs de zéros. Pour l'expliquer, nous allons utiliser l'exemple d'un modèle SAR par morceaux avec deux zones séparées par une frontière horizontale. Dans la zone supérieure, l'image est constituée de nuages et le modèle est un modèle SAR isotrope aux quatre plus proches voisins avec comme paramètres a_1 et σ_1 . Dans la zone inférieure, il s'agit de ciel et d'un modèle SAR isotrope aux quatre plus proches voisins avec comme coefficients a_2 et σ_2 .



$$Q = \left(\begin{array}{cccc|cccc} 1 & -a & & -a & & & & \\ -a & \ddots & & & & & & \\ & & \ddots & & & & & \\ -a & & & -a & 1 & & & \\ \hline & & & & & 1 & -b & -b \\ & & & & & -b & \ddots & \ddots \\ & & & & & & \ddots & \ddots & -b \\ & & & & & -b & & -b & 1 \end{array} \right)$$

où les deux blocs non diagonaux sont à définir. Dans la définition 9, ces blocs sont des blocs de 0. Les deux zones de l'image sont alors indépendantes.

Nous proposons de compléter la matrice Q de la façon suivante

$$Q = \left(\begin{array}{cccc|cccc} 1 & -a_1 & & -a_1 & 0 & \dots & \dots & 0 \\ -a_1 & \ddots & & & -\frac{a_1+a_2}{2} & \ddots & & \vdots \\ & & \ddots & & 0 & \ddots & & \vdots \\ -a_1 & & & -a_1 & 1 & 0 & 0 & -\frac{a_1+a_2}{2} & 0 \\ \hline 0 & -\frac{a_1+a_2}{2} & 0 & 0 & 1 & -a_2 & & -a_2 \\ \vdots & \ddots & \ddots & 0 & -a_2 & \ddots & \ddots & \\ \vdots & & \ddots & -\frac{a_1+a_2}{2} & & \ddots & \ddots & -a_2 \\ 0 & \dots & \dots & 0 & -a_2 & & -a_2 & 1 \end{array} \right)$$

Avec ce choix de complétion, les zones ne sont plus supposées indépendantes. Nous faisons l'hypothèse qu'il y a des interactions entre la zone de ciel et la zone de nuages. Plus précisément, nous supposons que pour les pixels qui sont le long de la frontière horizontale, la covariance avec les voisins appartenant à l'autre zone est non nulle.

Cette complétion est peut-être plus réaliste. Toutefois, l'estimation des paramètres a_1 et a_2 va devenir plus coûteuse. Nous allons voir dans le prochain paragraphe pourquoi et comment essayer d'y remédier.

Nous aurions aussi pu la compléter en remplaçant $\frac{a_1+a_2}{2}$ par toute autre combinaison linéaire de a_1 et de a_2 tant que la matrice reste inversible et définie positive.

Finalement, pour chaque partition $R = (R_1, \dots, R_P)$ de la grille $\llbracket 1, M \rrbracket^2$ et pour chaque famille finie de modèles stationnaires (S_1, \dots, S_P) , nous pouvons définir un modèle fini strictement stationnaire par morceaux. Nous verrons dans le chapitre suivant comment choisir le « meilleur » modèle pour représenter l'image I une fois centrée.

2.4.3 Estimation des paramètres par maximum de vraisemblance

Soit I une image de taille $M \times M$ de densité inconnue f . Nous avons proposé différents modèles non stationnaires pour modéliser f . Nous allons voir comment estimer les paramètres sous chacun de ces modèles.

Modèles finis stationnaires par morceaux

Soit m un modèle stationnaire par morceaux défini par un partition $R = (R_1, \dots, R_P)$ et une liste de modèles stationnaires (S_1, \dots, S_P) . La matrice de covariance Σ est donc complétée par des blocs de 0. Les paramètres à estimer, notés θ_m , sont ceux des modèles finis stationnaires $S_1(R_1), \dots, S_P(R_P)$. Suite à l'hypothèse d'indépendance entre les restrictions du champ, le modèle $S_1(R_1)$ ne porte que sur la restriction de l'image I à la zone R_1 . L'estimation de ses paramètres ne se fait donc qu'à partir de la zone R_1 . Ainsi, l'estimation de l'ensemble des paramètres du modèle par morceaux m se fait indépendamment dans chacune des zones. De plus, nous appliquons dans chacune des zones la méthode d'estimation la plus adaptée : celle présentée au paragraphe 2.2.4 pour les modèles indirects ou celle du paragraphe 2.3.2 pour les modèles directs.

En notant $\hat{\theta}_i$ les paramètres estimés à partir de la zone R_i , nous avons donc $\hat{\theta}_m = (\hat{\theta}_1, \dots, \hat{\theta}_P)$. De même, si nous notons \hat{f}_i la densité estimée du modèle stationnaire $S_i(R_i)$, nous avons

$$\hat{f}_m(I, \hat{\theta}_m) = \prod_{i=1}^P \hat{f}_i(I|_{R_i}, \hat{\theta}_i).$$

Modèles non stationnaires

Dans le cas où nous n'aurions pas complété la matrice Σ par des blocs de 0, les restrictions du champ \mathbf{Y} à chacune des zones ne seraient plus supposées indépendantes et l'estimation des paramètres ne peut plus se faire zone par zone. Nous devons donc revenir à la méthode initiale qui consiste à chercher le maximum de vraisemblance de manière itérative à partir de la vraisemblance globale

$$\hat{\theta}_m = \operatorname{argmax}_{\theta} f_m(I, \theta).$$

L'interaction entre les zones rend le temps de calcul nécessaire à cette estimation très long. Certains auteurs ont alors proposé de remplacer la vraisemblance par une pseudo-vraisemblance. L'estimation des paramètres par maximum de cette pseudo vraisemblance est plus rapide que par maximisation de la vraisemblance. De plus, les estimateurs obtenus ont un comportement proche du maximum de vraisemblance.

Par exemple, Guyon [Guy82] propose d'utiliser la pseudo-vraisemblance conditionnelle définie pour un vecteur aléatoire $\mathbf{Y} \in \mathbb{R}^n$ par

$$\text{PVC}(\mathbf{Y}) = \prod_{i=1}^n L(Y_i | Y_k, k \neq i).$$

où $L(Y|\mathbf{Z})$ est la loi conditionnelle de Y sachant \mathbf{Z} . Cette pseudo-vraisemblance est le produit de n vraisemblances gaussiennes en dimension 1. Elle s'affranchit du terme $\det(\Sigma)$ que l'on retrouve dans une densité gaussienne en dimension n . C'est pour cette raison que la recherche du maximum de pseudo vraisemblance est rapide, comme pour l'estimateur des moindres carrés ordinaires.

Toutefois, des travaux menés par Xavier Guyon [Guy07] ont montré que le comportement de l'estimateur de pseudo-vraisemblance s'écartait de celui de l'estimateur du maximum de vraisemblance lorsque que la corrélation du fond augmentait. Or, nous souhaitons justement nous en servir dans un cadre d'images corrélées. Nous utiliserons cet estimateur plus en détail dans le chapitre suivant. Nous pourrons alors le comparer à l'estimateur du maximum de vraisemblance.

Conclusion

Dans ce chapitre, nous avons présenté des modèles stationnaires directs ou indirects. Nous avons porté une attention particulière aux modèles SAR et aux modèles CAR. Nous avons également présenté l'algorithme que nous avons implémenté pour l'estimation par maximum de vraisemblance des paramètres de ces modèles.

Nous avons ensuite étendu la définition de modèles stationnaires pour définir des modèles *stationnaires par morceaux* en supposant l'indépendance entre les différentes zones du champ. Un modèle m est donc entièrement défini par une partition $R = (R_1, \dots, R_P)$ de la grille $\llbracket 1, M \rrbracket^2$ ainsi qu'une liste de modèles stationnaires (S_1, \dots, S_P) .

Nous disposons alors d'une collection de modèles (de segmentation) pour représenter une image réelle I . Dans le prochain chapitre, nous allons appliquer des outils de sélection de modèles pour choisir le plus adapté pour représenter le contenu de I .

Chapitre 3

Sélection de modèles

Sommaire

3.1	Cadre général	76
3.1.1	Modèles oracles	76
3.1.2	Sélection pratique	77
3.1.3	Comparaison des estimateurs	77
3.2	Reformulation des hypothèses	79
3.2.1	Autre expression de l'hypothèse $(H_{\mathcal{M}(\mathcal{P}, \mathcal{S})})$	79
3.2.2	Autre expression des hypothèses (H_m)	80
3.2.3	Inégalité oracle pour les modèles gaussiens stationnaires par morceaux	81
3.3	Applications aux images réelles	81
3.3.1	Collections particulières de partitions	81
3.3.2	Ensemble \mathcal{S} particulier	84
3.3.3	Parcours des ensembles $\mathcal{M}(\mathcal{P}_1, \mathcal{S}_1)$ et $\mathcal{M}(\mathcal{P}_2, \mathcal{S}_1)$	84
3.3.4	Réglage de la pénalité	87
3.3.5	Exemples sur images simulées	87
3.3.6	Exemples sur images réelles	90
3.4	La pseudo log-vraisemblance	91
3.4.1	Définition	91
3.4.2	Exemples sur images simulées puis réelles	92

Introduction

Dans le chapitre précédent, nous avons construit une collection de modèles gaussiens par morceaux pour représenter une image infra-rouge I . Chaque modèle correspond à une segmentation de l'image I en zones rectangulaires non recouvrantes. Ces modèles sont tous paramétriques et la collection est grande. Nous souhaitons sélectionner parmi cette collection exhaustive le modèle le plus adapté pour modéliser l'image I . Une approche de segmentation d'images par sélection de modèles a déjà été envisagée dans les travaux de Galland et coauteurs [GBR03] ou de Vasquez [Vas11]. Le critère utilisé était la complexité stochastique [Ris86]. Dans ce chapitre, nous allons présenter le principe de sélection de modèles à partir du critère de log-vraisemblance pénalisée. Puis, nous appliquerons ce principe à la collection définie dans le chapitre précédent. Nous verrons que ces méthodes sont lourdes en temps de calcul et nous proposerons une alternative basée sur la pseudo log-vraisemblance.

3.1 Cadre général de la sélection de modèles

Dans un cadre très général, nous disposons d'un échantillon de taille n noté $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ de réalisations indépendantes d'un vecteur aléatoire \mathbf{X} de \mathbb{R}^d de loi inconnue (pas nécessairement gaussienne) de densité f par rapport à la mesure de Lebesgue sur \mathbb{R}^d . Nous cherchons alors à estimer f .

Si nous disposons d'un a priori sur la loi du vecteur \mathbf{X} comme par exemple que c'est une loi gaussienne alors, pour estimer f il suffit d'estimer son vecteur moyenne $\mathbf{m}\mathbf{o}$ et sa matrice de covariance Σ . On le fait par exemple par maximum de vraisemblance ce qui donne \hat{f} comme estimateur de f

$$\hat{f}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{Md}{2}} \det(\hat{\Sigma})^{\frac{1}{2}}} \exp\left(-\frac{1}{2} {}^t(\mathbf{x} - \widehat{\mathbf{m}\mathbf{o}}) \hat{\Sigma}^{-1} (\mathbf{x} - \widehat{\mathbf{m}\mathbf{o}})\right) \text{ où}$$

$$\widehat{\mathbf{m}\mathbf{o}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \text{ et } \hat{\Sigma} = \frac{1}{n} \sum_{i=1}^n {}^t \mathbf{x}_i \mathbf{x}_i.$$

En pratique, nous disposons plutôt d'une collection d'a priori portant par exemple sur la structure de $\mathbf{m}\mathbf{o}$ ou de Σ . Chacun de ces a priori correspond à un modèle statistique sous lequel la densité f est à estimer. Nous disposons ainsi d'une collection \mathcal{M} de modèles et nous allons choisir dans cette collection celui qui est le plus adapté pour estimer f .

Pour chacun des modèles $m \in \mathcal{M}$, nous pouvons supposer que la densité f appartient à ce modèle et fabriquer un estimateur \hat{f}_m de f . Dans nos travaux, nous avons construit \hat{f}_m par maximum de vraisemblance. Nous disposons alors d'une collection d'estimateurs $\{\hat{f}_m, m \in \mathcal{M}\}$.

3.1.1 Modèles oracles

Pour choisir un modèle dans la collection \mathcal{M} , nous voulons ensuite comparer ces estimateurs entre eux. Pour cela, nous leur associons un risque. Nous prenons comme fonction de risque la divergence de Kullback-Leibler définie, pour deux densités f et g par rapport à la mesure de Lebesgue, par

$$\text{KL}(f, g) = \begin{cases} \int \log\left(\frac{f}{g}\right) f d\lambda & \text{si } f d\lambda \ll g d\lambda. \\ +\infty & \text{sinon.} \end{cases}$$

où $P \ll Q$ signifie que la mesure P est absolument continue devant la mesure Q . Nous parlons aussi bien du risque de l'estimateur \hat{f}_m que du risque du modèle m directement.

Nous souhaitons minimiser cette fonction de risque et nous définissons les modèles dits *oracle* par

$$m^* \in \operatorname{argmin}_{m \in \mathcal{M}} \text{KL}(f, \hat{f}_m).$$

Il n'y a pas unicité du modèle oracle. En revanche, ils ont tous la même valeur de risque (que l'on appellera *risque oracle*) et cette valeur est une référence puisqu'elle correspond au plus petit risque possible. Malheureusement, f est inconnue et la recherche de modèles oracles comme définie ci-dessus est impossible.

Massart [Mas09] a alors proposé une procédure de sélection de modèles basée seulement sur les données $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$. Cette procédure sélectionne un modèle \hat{m} dont le risque est proche du risque oracle.

3.1.2 Sélection pratique

Puisque f est inaccessible, nous allons remplacer le risque $\text{KL}(f, \cdot)$ par un risque empirique ne dépendant que de l'échantillon $(\mathbf{x}_1, \dots, \mathbf{x}_n)$. Le but est de sélectionner un modèle dont le risque est proche du risque oracle. Nous choisissons donc un critère empirique, noté $\text{kl}_n(\mathbf{x}, \cdot)$, tel que pour tout estimateur \hat{f}_m , $\mathbb{E}_f(\text{kl}_n(\mathbf{X}, \hat{f}_m)) = \text{KL}(f, \hat{f}_m)$. Le choix le plus naturel est alors

$$\text{kl}_n(\mathbf{x}, \hat{f}_m) = -\frac{1}{n} \sum_{i=1}^n \log \left(\frac{\hat{f}_m(\mathbf{x}_i)}{f(\mathbf{x}_i)} \right) = -\frac{1}{n} \sum_{i=1}^n \log(\hat{f}_m(\mathbf{x}_i)) + \frac{1}{n} \sum_{i=1}^n \log(f(\mathbf{x}_i))$$

Cependant, f ne doit pas intervenir dans ce critère empirique et nous aurions envie de choisir comme risque empirique

$$\gamma_n(\mathbf{x}, \hat{f}_m) = -\frac{1}{n} \sum_{i=1}^n \log(\hat{f}_m(\mathbf{x}_i)).$$

Toutefois, nous pouvons montrer que $\mathbb{E}_f(\gamma_n(\mathbf{X}, \hat{f}_m)) < \text{KL}(f, \hat{f}_m)$. Le risque empirique $\gamma_n(\mathbf{x}, \hat{f}_m)$ est donc un estimateur biaisé du risque $\text{KL}(f, \hat{f}_m)$. Si nous voulions utiliser un estimateur non biaisé, la formule ferait forcément intervenir f . Massart [Mas09] a alors proposé de pénaliser $\gamma_n(\mathbf{x}, \hat{f}_m)$ par une quantité ne dépendant pas de f mais de la complexité du modèle m

$$\hat{m} = \underset{m \in \mathcal{M}}{\text{argmin}} [\gamma_n(\mathbf{x}, \hat{f}_m) + \text{pen}(m)]$$

où $\text{pen} : \mathcal{M} \rightarrow \mathbb{R}_+$ mesure la complexité du modèle. L'estimateur retenu pour f est alors $\hat{f} = \hat{f}_{\hat{m}}$.

La fonction de pénalité est souvent proportionnelle à la dimension du modèle, c'est-à-dire au nombre de paramètres à estimer dans ce modèle ($\text{pen}(m) = \lambda \dim(m)$). Le coefficient de proportionnalité dépend de l'échantillon $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ ([BM07],[Arl07]). Nous verrons plus loin comment le choisir.

3.1.3 Comparaison des estimateurs

Une fois construit l'estimateur $\hat{f} = \hat{f}_{\hat{m}}$, nous souhaitons comparer son risque à celui de l'estimateur oracle \hat{f}_{m^*} . Dans les applications suivantes, nous travaillerons avec une seule image, c'est-à-dire $n = 1$. Nous avons donc envie de comparer les risques de façon non asymptotique. Idéalement, nous voudrions montrer que $\mathbb{E}[\text{KL}(f, \hat{f})] = \mathbb{E}[\text{KL}(f, \hat{f}_{m^*})]$. En toute généralité, cette égalité n'est pas vraie. Le théorème suivant de Pascal Massart [Mas03] permet de donner des conditions sur la fonction de pénalité pour obtenir un résultat de la forme

$$\mathbb{E}[h^2(f, \hat{f}_{\hat{m}})] \leq C \left(\inf_{m \in \mathcal{M}} [\text{KL}(f, m) + \text{pen}(m)] \right) + \frac{\Gamma}{n} \quad (3.1)$$

où $h^2(f, g) = \|\sqrt{f} - \sqrt{g}\|_2^2$ est la distance d'Hellinger entre deux densités et $\text{KL}(f, m) = \inf_{t \in m} \text{KL}(f, t)$. Après l'énoncé du théorème, nous expliquerons comment ce résultat peut donner des renseignements sur le risque de l'estimateur $\hat{f}_{\hat{m}}$.

Le théorème que nous souhaitons appliquer nécessite des hypothèses sur la collection \mathcal{M} ainsi que sur chacun des modèles $m \in \mathcal{M}$. Plus particulièrement, ces hypothèses portent sur l'entropie à crochet des modèles m . Nous rappelons d'abord la définition.

Définition 10 (Entropie à crochets) Un crochet de taille ε est un couple de fonctions $[t^-, t^+]$ tel que $t^- \leq t^+$ et $h^2(t^-, t^+) \leq \varepsilon^2$.

Soit m un modèle statistique et $\varepsilon > 0$. On appelle entropie à crochet de taille ε du modèle m , et l'on note $H_{\square}(\varepsilon, m)$, le logarithme du nombre minimum de crochets de taille ε nécessaires pour recouvrir tout le modèle m .

$$H_{\square}(\varepsilon, m) = \log(N_{\square}(\varepsilon, m)).$$

Nous pouvons maintenant énoncer les hypothèses nécessaires ainsi que le théorème.

Hypothèse (H_m) Il existe une fonction croissante ϕ_m telle que la fonction $x \mapsto \frac{\phi_m(x)}{x}$ est décroissante et

$$\forall \sigma \in \mathbb{R}_+, \int_0^{\sigma} \sqrt{H_{\square}(\varepsilon, m)} \leq \phi_m(\sigma)$$

Hypothèse ($H_{\mathcal{M}}$) Il existe $(\rho_m, m \in \mathcal{M})$ une famille de nombres positifs telle que

$$\sum_{m \in \mathcal{M}} \exp(-\rho_m) = \Gamma < \infty$$

Théorème 1 Soit $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ un échantillon de même densité inconnue f . Soit \mathcal{M} une collection dénombrable de modèles. On considère la collection d'estimateurs du maximum de vraisemblance associé à chacun des modèles $\{\hat{f}_m, m \in \mathcal{M}\}$.

On suppose que l'hypothèse ($H_{\mathcal{M}}$) est vérifiée. On suppose aussi que pour tout modèle $m \in \mathcal{M}$, l'hypothèse (H_m) est vérifiée et on note σ_m l'unique solution positive de $\phi_m(y) = \sqrt{ny}^2$.

Alors, il existe des constantes K et C telles que pour toutes fonctions pen de la forme

$$\forall m \in \mathcal{M} \quad \text{pen}(m) \geq K \left(\sigma_m^2 + \frac{\rho_m}{n} \right)$$

on obtient

$$\mathbb{E}[h^2(f, \hat{f}_{\bar{m}})] \leq C \left(\inf_{m \in \mathcal{M}} [\text{KL}(f, m) + \text{pen}(m)] + \frac{\Gamma}{n} \right)$$

Dans la preuve, l'hypothèse (H_m) permet d'obtenir pour chacun des estimateurs \hat{f}_m une inégalité oracle de la forme

$$\mathbb{E}[h^2(f, \hat{f}_m)] \leq C \left(\inf_{m \in \mathcal{M}} [\text{KL}(f, m) + \text{pen}(m)] + \frac{\Gamma}{n} \right)$$

L'hypothèse ($H_{\mathcal{M}}$) permet de généraliser cette inégalité à l'estimateur $\hat{f}_{\bar{m}}$.

Remarque. La première remarque à faire concernant ce résultat est qu'il n'est pas asymptotique. Il est vrai quelque soit la taille n de l'échantillon utilisé. Ce n'est pas le cas du critère AIC par exemple qui s'appuie sur le comportement asymptotique de la log-vraisemblance d'un échantillon.

La deuxième remarque à faire est que cette inégalité ne ressemble pas tout à fait à l'égalité souhaitée $\mathbb{E}[\text{KL}(f, \hat{f}_{\bar{m}})] = \mathbb{E}[\text{KL}(f, \hat{f}_{m^*})]$. Concernant le membre de gauche, nous n'avons pas le risque de l'estimateur $\hat{f}_{\bar{m}}$ mais seulement la distance d'Hellinger $h^2(f, \hat{f}_{\bar{m}})$. Sous certaines hypothèses un peu plus fortes, nous pouvons appliquer le lemme 7.23 de [Mas03] qui minimise $h^2(f, \hat{f}_{\bar{m}})$ par $\text{KL}(f, \hat{f}_{\bar{m}})$ et ainsi obtenir

$$\mathbb{E}[\text{KL}(f, \hat{f}_{\bar{m}})] \leq \tilde{C} \left(\inf_{m \in \mathcal{M}} [\text{KL}(f, m) + \text{pen}(m)] + \frac{\Gamma}{n} \right)$$

Concernant le membre de droite, la démonstration permet de montrer que $\text{pen}(m)$ est en fait un majorant de $\mathbb{E}(\text{KL}(f, \hat{f}_m)) - \text{KL}(f, f_m)$. Ainsi,

$$\mathbb{E}(\text{KL}(f, \hat{f}_m)) = \text{KL}(f, f_m) + \mathbb{E}(\text{KL}(f, \hat{f}_m)) - \text{KL}(f, f_m) \leq \text{KL}(f, f_m) + \text{pen}(m)$$

Le résultat de Massart ne permet pas d'obtenir directement dans le terme de droite le risque oracle $\inf_{m \in \mathcal{M}} \text{KL}(f, f_m)$ mais un majorant de ce risque.

Ce résultat a été utilisé dans beaucoup de domaines. Nous pouvons citer, par exemple, Maugis et Michel [MM11] dans un cadre de sélection de variables et de classification non supervisée. Citons également Le Pennec et Cohen [LPC13] dans un cadre d'estimation de densité conditionnelle. Gendre, dans [Gen09] ou [Gen14], a utilisé ces outils dans un cadre de régression hétéroscédastique. Baraud et coauteurs [BHL03] ont appliqué ces résultats dans un cadre de tests statistiques. Parmi une collection de tests $\{T_m, m \in \mathcal{M}\}$, ils sélectionnent celui qui minimise un critère pénalisé. Enfin, Verzelen [Ver09] a utilisé ses résultats avec comme collection \mathcal{M} une liste de modèles CAR pour différents voisinages comme définis en 2.2.2. Pour un champ gaussien \mathbf{X} , il sélectionne le modèle CAR le plus adapté pour représenter les données.

3.2 Reformulation du théorème pour les modèles stationnaires par morceaux

Nous avons appliqué cette procédure d'estimation de densités via la sélection de modèles afin d'estimer la densité inconnue f d'une image infra-rouge I de taille $M \times M$.

Comme présenté dans le chapitre 1, nous avons travaillé avec des modèles (gaussiens) stationnaires par morceaux, c'est-à-dire que nous avons supposé que la densité f pouvait s'écrire comme le produit des densités des restrictions du champ à chacune des zones. Un modèle m est entièrement défini par une partition $R = (R_1, \dots, R_P)$ appartenant à une collection de partitions \mathcal{P} de la grille $\llbracket 1, M \rrbracket^2$ ainsi que P restrictions (aux zones R_i) de modèles stationnaires de l'ensemble \mathcal{S} . On note ces restrictions $(S_{i_1}, \dots, S_{i_P})$ au lieu de $(S_{i_1}(R_1), \dots, S_{i_P}(R_P))$ s'il n'y a aucune ambiguïté sur la partition R utilisée pour définir le modèle m . On note $\mathcal{M}(\mathcal{P}, \mathcal{S})$ la collection de modèles par morceaux fabriqués à partir des partitions de \mathcal{P} et des modèles stationnaires de \mathcal{S} .

Numériquement, la recherche du minimum pour définir \hat{m} est très coûteuse en temps de calcul. Nous souhaitons donc restreindre la collection $\mathcal{M}(\mathcal{P}, \mathcal{S})$ sur laquelle se fait la recherche du minimum. D'un point de vue théorique, nous devons toutefois nous assurer que l'hypothèse $(H_{\mathcal{M}(\mathcal{P}, \mathcal{S})})$ est vérifiée.

3.2.1 Autre expression de l'hypothèse $(H_{\mathcal{M}(\mathcal{P}, \mathcal{S})})$

Comme chacun des modèles m est défini par une partition R ainsi qu'une liste de modèles stationnaires S_{i_1}, \dots, S_{i_P} , nous avons cherché les coefficients ρ_m sous la forme

$$\rho_m = \rho_R + \rho_{i_1} + \dots + \rho_{i_P}$$

où chaque indice i_j correspond à un modèle de l'ensemble \mathcal{S} . Alors,

$$\begin{aligned} \sum_{m \in \mathcal{M}(\mathcal{P}, \mathcal{S})} \exp(-\rho_m) &= \sum_{R \in \mathcal{P}} \sum_{i_1, \dots, i_P \in \mathcal{S}^P} \exp(-\rho_R - \rho_{i_1} - \dots - \rho_{i_P}) \\ &= \sum_{R \in \mathcal{P}} e^{-\rho_R} \left(\sum_{j=1}^{|\mathcal{S}|} e^{-\rho_j} \right)^P \end{aligned}$$

Ainsi, si pour tout $j \in \mathcal{S}$, on pose $\rho_j = \log(|\mathcal{S}|)$, l'hypothèse à vérifier n'est plus $(H_{\mathcal{M}})$ mais seulement $(H_{\mathcal{P}})$.

Hypothèse $(H_{\mathcal{P}})$ Il existe une famille de nombres positifs $(\rho_R, R \in \mathcal{P})$ telle que

$$\sum_{R \in \mathcal{P}} \exp(-\rho_R) \leq 1$$

3.2.2 Autre expression des hypothèses (H_m)

Pour appliquer le théorème 1 à une collection de modèles $\mathcal{M}(\mathcal{P}, \mathcal{S})$, nous avons besoin d'exhiber pour chaque modèle m une fonction ϕ_m vérifiant l'hypothèse (H_m) et un réel σ_m solution de $\phi_m(y) = \sqrt{ny^2}$.

Le lemme suivant permet de donner une expression de ϕ_m dans le cas particulier d'un modèle stationnaire par morceaux.

Lemme 1 Soit $m = (R, S_{i_1}, \dots, S_{i_P})$ un modèle de la collection $\mathcal{M}(\mathcal{P}, \mathcal{S})$. Si pour tout $j \in \llbracket 1, P \rrbracket$

$$\forall \varepsilon > 0, H_{\square}(\varepsilon, S_{i_j}) \leq \dim(S_{i_j}) \left(C + \log \frac{\#(R_j)}{\varepsilon} \right)$$

alors la fonction ϕ_m définie par

$$\phi_m(\sigma) = \sigma \sqrt{\dim(m)} \left(\sqrt{C + \log(M^2)} + \sqrt{\pi} + \sqrt{\log \frac{1}{\sigma \wedge e^{-1/2}}} \right)$$

vérifie l'hypothèse (H_m).

Ce lemme est démontré dans l'appendice B.2.3 à partir d'un résultat énoncé et démontré par Le Pennec et Cohen dans [LPC13].

Dans la suite de nos travaux, nous avons donc remplacé l'hypothèse (H_m) par l'hypothèse (H'_m) suivante :

Hypothèse (H'_m) Soit $m = (R_1, \dots, R_P, S_{i_1}, \dots, S_{i_P})$ un modèle de $\mathcal{M}(\mathcal{P}, \mathcal{S})$, il existe une constante $C > 0$ telle que

$$\forall \varepsilon > 0 \quad \forall j \in \llbracket 1, P \rrbracket, H_{\square}(\varepsilon, S_{i_j}) \leq \dim(S_{i_j}) \left(C + \log \frac{\#(R_j)}{\varepsilon} \right)$$

Les hypothèses (H'_m) comme les hypothèses (H_m) ne sont pas des hypothèses faciles à lire ni à vérifier. Dans les applications présentées dans les chapitres suivants, l'ensemble \mathcal{S} est un ensemble composé de modèles gaussiens paramétriques. Cela signifie que pour une partition donnée $R = (R_1, \dots, R_P)$, les restrictions $(S_{i_j})_{j=1 \dots P}$ sont entièrement caractérisées par leur vecteur moyen mo_{i_j} et leur matrice de covariance Σ_{i_j} .

Nous avons alors démontré la proposition suivante qui permet de traduire l'hypothèse (H'_m) dans le cas particulier d'un modèle m gaussien fini stationnaire par morceaux.

Proposition 2 Soit $m \sim N(0, \Sigma)$ un modèle gaussien paramétrique ($\dim(m) = d$) sur \mathbb{R}^p tel que

$$\theta \mapsto \log(\det(\Sigma(\theta))) \text{ est } k.p\text{-lipschitz pour la norme infinie sur } \mathbb{R}^d \text{ et la norme } L1 \text{ sur } \mathbb{R}.$$

$$\theta \mapsto \Sigma(\theta) \text{ est } k\text{-lipschitz pour la norme infinie sur } \mathbb{R}^d \text{ et la norme triple sur } M_p(\mathbb{R}).$$

alors on peut majorer l'entropie à crochet du modèle m

$$\forall \varepsilon > 0, H_{\square}(\varepsilon, m) \leq \dim(m) \left(C + \log \frac{p}{\varepsilon} \right).$$

La démonstration est présentée en appendice B.2. Ce résultat nous permet, pour un modèle m gaussien stationnaire par morceaux de remplacer l'hypothèse (H'_m) par l'hypothèse (G_m) suivante

Hypothèse (G_m) Soit $m = (R_1, \dots, R_P, S_{i_1}, \dots, S_{i_P})$ tel que $\forall j \in \llbracket 1, P \rrbracket, S_{i_j} \sim N(m_{i_j}, \Sigma_{i_j})$.

$$\forall j \in \llbracket 1, P \rrbracket, \theta \mapsto \log(\det(\Sigma_{i_j}(\theta))) \text{ est } k.\#(R_j)\text{-lipschitz}$$

$$\theta \mapsto \Sigma_{i_j}(\theta) \text{ est } k\text{-lipschitz}$$

3.2.3 Inégalité oracle pour les modèles gaussiens stationnaires par morceaux

Maintenant que nous avons reformulé les hypothèses (H_m) et $(H_{\mathcal{M}})$, nous pouvons énoncer le théorème de Massart pour les modèles gaussiens stationnaires par morceaux.

Théorème 2 *Soit $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ un échantillon de même densité inconnue f . Soit \mathcal{P} une collection de partitions pour laquelle l'hypothèse $(H_{\mathcal{P}})$ est vérifiée. Soit \mathcal{S} un ensemble fini de modèles gaussiens stationnaires tel que pour tout $m = (R, S_{i_1}, \dots, S_{i_P}) \in \mathcal{M}(\mathcal{P}, \mathcal{S})$, l'hypothèse (G_m) soit vérifiée.*

Pour chaque modèle m , on note \hat{f}_m l'estimateur du maximum de vraisemblance. soit $\text{pen} : \mathcal{M}(\mathcal{P}, \mathcal{S}) \mapsto \mathbb{R}_+$ et considérons le critère de log-vraisemblance pénalisée

$$\text{crit}(m) = \sum_{i=1}^n -\log(\hat{f}_m(\mathbf{x}_i)) + \text{pen}(m)$$

Alors il existe des constantes K et C telles que si pour tous les modèles $m \in \mathcal{M}(\mathcal{P}, \mathcal{S})$

$$\text{pen}(m) \geq K \frac{\dim(m)}{n} \left[\sqrt{C + \log(M^2)} + \sqrt{\pi} + \sqrt{\log \frac{1}{\sqrt{\frac{\dim(m)}{n}} (\sqrt{C + \log(M^2)} + \sqrt{\pi}) \wedge e^{-1/2}}} \right]^2$$

alors le modèle \hat{m} qui minimise le critère pénalisé sur $\mathcal{M}(\mathcal{P}, \mathcal{S})$ est tel que

$$\mathbb{E}[\text{KL}(f, \hat{f}_{\hat{m}})] \leq \tilde{C} \inf_{m \in \mathcal{M}} \left(\mathbb{E}[\text{KL}(f, f_m) + \text{pen}(m) \right) + \frac{1}{n}.$$

La démonstration de ce théorème est présentée en appendice B. Plusieurs remarques sont à faire. Tout d'abord, ce résultat valide l'utilisation en pratique d'une fonction de pénalité proportionnelle à la dimension de chacun des modèles. Il montre aussi que le risque de l'estimateur $\hat{f}_{\hat{m}}$ est proche du risque oracle. Toutefois, la démonstration ne donne pas la valeur exacte des constantes K et C et le coefficient de proportionnalité entre la pénalité et la dimension ($\text{pen}(m) = \lambda \dim(m)$) du modèle m est à définir empiriquement [Arl07]. Nous verrons dans le paragraphe dédié à l'étude empirique la valeur choisie pour ce coefficient.

3.3 Applications aux images réelles

Dans la partie précédente, nous avons étudié les propriétés théoriques de la procédure proposée. Dans cette partie, nous allons appliquer cette procédure de sélection de modèles à des images simulées puis réelles.

Nous réglerons en parallèle le choix des constantes apparues dans le théorème 2.

3.3.1 Collections particulières de partitions

Pour des raisons de temps de calcul, nous allons restreindre la collection de modèles $\mathcal{M}(\mathcal{P}, \mathcal{S})$ à deux collections de modèles par morceaux définis pour des collections de partitions particulières. Après les avoir définies, nous montrerons qu'elles vérifient bien l'hypothèse $(H_{\mathcal{P}})$.

Partitions récursives dyadiques

Pour cette collection de partitions, les images doivent être de taille une puissance de 2, c'est à dire que $M = 2^k$.

Soit $R = (R_1, \dots, R_P)$ une partition récursive dyadique. Alors, toutes les zones R_i sont des carrés de côté de longueur 2^{k-j} et les frontières entre les différentes zones R_i sont à des lignes ou à des colonnes dont le numéro est la forme $c \cdot 2^{k-l}$. Nous pouvons nous référer à Sart [Sar14] ou à Le Pennec et Cohen [LPC13] pour une autre définition. La figure 3.1 présente deux exemples de partitions récursives dyadiques. On note \mathcal{P}_1 l'ensemble de toutes les partitions récursives dyadiques.

Cette collection de partitions possède une forte structure hiérarchique. En effet, chacune des partitions de la collection \mathcal{P}_1 est obtenue à partir de la partition R_0 ne contenant qu'une seule

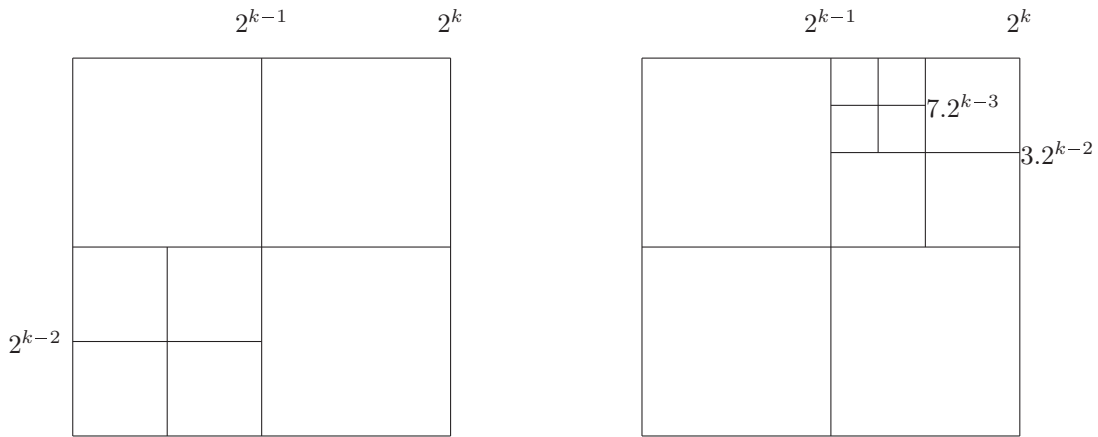


FIGURE 3.1 – Deux exemples de partitions récursives dyadiques

zone : à chaque étape, chacun des carrés d’une partition peut être divisé en 4 carrés de même taille. Grâce à cette structure hiérarchique, nous pouvons représenter une partition de la collection \mathcal{P}_1 par un arbre dans lequel chaque nœud a 0 ou 4 branches. Chaque feuille de l’arbre correspond à une zone de la partition. La figure 3.2 présente les arbres correspondants aux partitions de la figure 3.1. Le nombre de feuilles de ce sous-arbre correspond au nombre de zones dans la partition. Cet arbre peut ensuite être codé en binaire où un 0 représente un nœud sans enfant et 1 un nœud avec 4 enfants. Les codes pour chacune des partitions de la figure 3.1 sont donnés dans la figure 3.2.

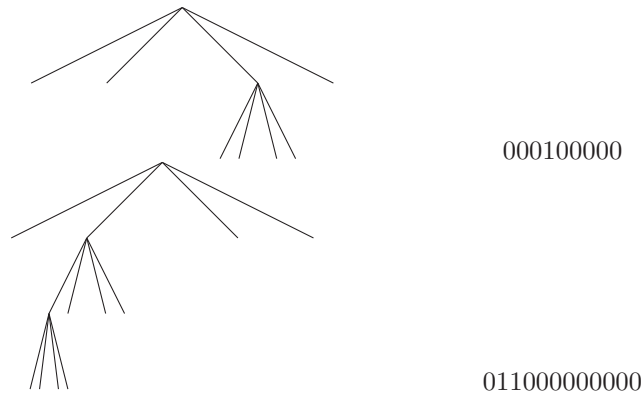


FIGURE 3.2 – Arbres dyadiques et codes associés pour les partitions de la figure 3.1

Deux arbres différents ne peuvent pas être codés par la même séquence binaire. On dit alors que ce codage est décodable. Une propriété importante d’un codage décodable est qu’il vérifie l’inégalité de Kraft-Milman [McM56].

Proposition 3 (Inégalité de Kraft-Milman) *Soit c un code décodable sur un alphabet à D éléments. Alors, les longueurs des mots $(l_n)_{n \geq 0}$ doivent vérifier*

$$\sum_{n \geq 0} D^{-l_n} \leq 1.$$

Pour coder les partitions dyadiques, nous avons utilisé l’alphabet binaire donc $D = 2$. On note $l(R)$ la longueur du code pour la partition R . Alors,

$$\sum_{R \in \mathcal{P}_1} 2^{-l(R)} \leq 1 \Rightarrow \sum_{R \in \mathcal{P}_1} e^{-\log(2)l(R)} \leq 1 \tag{3.2}$$

Nous n'avons utilisé qu'un symbole par nœud donc $l(R)$ est aussi le nombre total de nœuds dans l'arbre. Essayons aussi de relier la longueur du code $l(R)$ au nombre de zones dans la partition R , que l'on note $|R|$. Il y a deux sortes de nœuds dans l'arbre : ceux n'ayant pas d'enfants, dont le cardinal est $|R|$, et ceux ayant des enfants, de cardinal $l(R) - |R|$. Exceptée la racine, les autres nœuds sont tous l'enfant d'un nœud donc

$$\begin{aligned} 4(l(R) - |R|) &= l(R) - 1 \\ l(R) &= \frac{4|R| - 1}{3} \geq |R| \end{aligned}$$

En utilisant l'inégalité (3.2), on obtient alors

$$\sum_{R \in \mathcal{P}_1} e^{-\log(2)|R|} \leq \sum_{R \in \mathcal{P}_1} e^{-\log(2)l(R)} \leq 1.$$

La collection des partitions récursives dyadiques vérifie donc bien l'hypothèse $(H_{\mathcal{P}_1})$ pour la famille de poids $\rho_R = \log(2)|R|$. Elle permet ainsi de fabriquer une collection de modèles à laquelle on pourra appliquer le théorème 2. En particulier, le risque de l'estimateur obtenu sera proche du risque oracle. Toutefois, en utilisant les partitions récursives dyadiques, l'image sera segmentée en zones carrées. Cette contrainte est trop forte et nous avons alors opté pour une autre collection de partitions.

Partitions récursives split

Pour cette collection, l'image n'est plus nécessairement de taille une puissance de 2. On trouve dans cette collection toutes les partitions de la grille $\llbracket 1, M \rrbracket^2$ en rectangles. La figure 3.3 donne deux exemples de telles partitions. On note \mathcal{P}_2 la collection des partitions récursives dyadiques split. On peut tout de suite remarquer que les partitions récursives dyadiques sont une sous-famille des partitions récursives split.

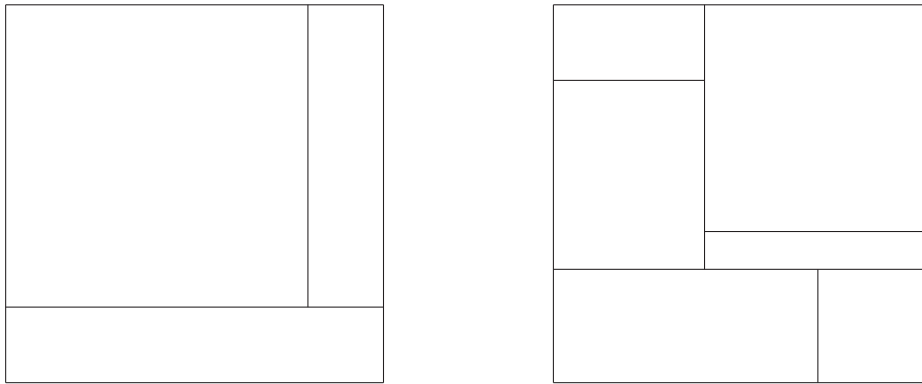


FIGURE 3.3 – Deux exemples de partitions récursives split

Pour cette collection aussi, il existe une structure hiérarchique : à chaque étape, un rectangle peut être découpé en deux rectangles pas nécessairement de même taille. De plus, ce découpage peut se faire parallèlement à la longueur ou à la largeur du rectangle initial. On peut encore coder ces partitions en binaire. Le codage de chaque nœud sera toutefois plus complexe que pour une partition récursive dyadique. En effet, en plus de coder si le rectangle est découpé ou non, le code doit aussi définir les nouveaux rectangles obtenus en donnant la direction (horizontale ou verticale) ainsi que la position de la découpe. Le codage de chaque nœud est donc de longueur $1 + 1 + \lceil \log_2(M) \rceil$. Pour une partition récursive dyadique split R dont l'arbre dyadique a $N(R)$ nœuds, on a donc $l(R) = N(R)(2 + \lceil \log_2(M) \rceil)$. En utilisant de nouveau l'inégalité de Kraft-Milman [McM56] pour les codes décodables, nous obtenons

$$\sum_{R \in \mathcal{P}_2} 2^{-N(R)(2 + \lceil \log_2(M) \rceil)} \leq 1 \Rightarrow \sum_{R \in \mathcal{P}_2} e^{-N(R) \log(2)(2 + \lceil \log_2(M) \rceil)} \leq 1.$$

Comme dans le cas des partitions récursives dyadiques, nous pouvons montrer que $N(R) \geq |R|$. L'hypothèse $(H_{\mathcal{P}})$ est donc bien vérifiée pour la collection des partitions récursives dyadiques split avec la famille de poids $\rho_R = |R| \log(2)(2 + \lceil \log_2(M) \rceil)$.

3.3.2 Ensemble \mathcal{S} particulier

Les modèles stationnaires de l'ensemble \mathcal{S} correspondent aux modèles suivis par les restrictions du champ \mathbf{X} à chacune des zones de la partition R . Il peut s'agir de modèles directs ou indirects comme présentés au chapitre 2.

Dans nos travaux, nous supposons que dans chacune des zones d'une partition R , la restriction du champ suit un modèle gaussien SAR isotrope aux 4 plus proches voisins défini en 2.6 :

$$\forall (i, j) \in P, X_{i,j} = a_P(X_{i-1,j} + X_{i+1,j} + X_{i,j-1} + X_{i,j+1}) + \sigma_P \varepsilon_{i,j}.$$

où ε est un champ gaussien centré et réduit. Seuls les paramètres a_P et σ_P varient d'une zone à l'autre.

Cette situation correspond donc à un ensemble \mathcal{S} réduit à un seul élément : le modèle SAR aux 4 plus proches voisins. Nous avons appelé cet ensemble \mathcal{S}_1 .

Vérification des hypothèses (G_m)

Nous voulons vérifier que les modèles des collections $\mathcal{M}(\mathcal{P}_1, \mathcal{S}_1)$ et $\mathcal{M}(\mathcal{P}_2, \mathcal{S}_1)$ vérifient bien les hypothèses (H_m) pour pouvoir ensuite appliquer le théorème 2. Puisque ce sont des modèles gaussiens, nous allons plutôt vérifier l'hypothèse (G_m) .

Soit $m \in \mathcal{M}(\mathcal{P}_1, \mathcal{S}_1)$. Le modèle m est donc défini par une partition $R = (R_1, \dots, R_P) \in \mathcal{P}_1$ et des modèles locaux $(S_{i_1}, \dots, S_{i_P})$ avec

$$\begin{aligned} \forall j \in \llbracket 1, P \rrbracket, S_{i_j} &= N(0, \Sigma_{i_j}) \text{ où} \\ \Sigma_{i_j} &= \sigma_{i_j} (I - a_{i_j} W_{i_j})^{-2} \end{aligned}$$

où W_{i_j} est la matrice des 4 plus proches voisins adaptée à la taille de la zone R_{i_j} . La proposition suivante permet de montrer que le modèle m vérifie bien l'hypothèse (G_m) .

Proposition 4 *Soit P une zone de la grille $\llbracket 1, M \rrbracket^2$. Alors,*

$$\begin{aligned} (a_P, \sigma_P) &\mapsto \log(\det(\sigma_P(I - a_P W_P)^{-2})) \text{ est } k.\#(P)\text{-lipschitz} \\ (a_P, \sigma_P) &\mapsto \sigma_P(I - a_P W_P)^{-2} \text{ est } k\text{-lipschitz} \end{aligned}$$

Nous avons démontré cette proposition dans l'appendice C.1 en utilisant le fait que la densité d'un modèle SAR isotrope a une forme particulière.

L'application du théorème 2 permet d'affirmer, qu'en théorie au moins, l'estimateur obtenu par la procédure de Massart pour les collections $\mathcal{M}(\mathcal{P}_1, \mathcal{S}_1)$ ou $\mathcal{M}(\mathcal{P}_2, \mathcal{S}_1)$ a un comportement proche de l'estimateur oracle.

Ceci nous motive pour l'utiliser en pratique.

3.3.3 Parcours des ensembles $\mathcal{M}(\mathcal{P}_1, \mathcal{S}_1)$ et $\mathcal{M}(\mathcal{P}_2, \mathcal{S}_1)$

Une fois fixé l'ensemble $\mathcal{M}(\mathcal{P}, \mathcal{S})$ sur lequel se fera la recherche du minimum

$$\hat{m} = \operatorname{argmin}_{m \in \mathcal{M}(\mathcal{P}, \mathcal{S})} \left(\log(\hat{f}_m(I)) + \operatorname{pen}(m) \right)$$

nous devons programmer cette recherche.

L'approche la plus naturelle consiste à calculer le critère $\operatorname{crit}(m) = \log(\hat{f}_m(I)) + \operatorname{pen}(m)$ pour chacun des modèles m puis à sélectionner celui qui le minimise. Cette méthode coûte cher en temps de calcul. Grâce à la structure *par morceaux* de nos modèles, nous allons pouvoir appliquer une méthode par arbre de tri pour estimer \hat{m} .

Un modèle m est de la forme $m = (R_1, \dots, R_P, S_{i_1}, \dots, S_{i_P})$. Puisque nous avons supposé l'indépendance du champ dans chacune des zones, nous avons $\hat{f}_m = \hat{f}_{R_1} \times \dots \times \hat{f}_{R_P}$ où \hat{f}_{R_i} est l'estimateur de la densité de la restriction du champ à la zone R_i sous le modèle m . La log-vraisemblance globale s'écrit comme la somme des log-vraisemblances de chacune des zones.

$$\log(\hat{f}_m(I)) = \sum_{i=1}^P \log(\hat{f}_{R_i}(I_i)).$$

De même, la dimension du modèle m , définie comme le nombre de paramètres à estimer, est la somme de la dimension des modèles dans chacune des zones. Puisqu'on utilise comme seul modèle local le modèle SAR aux 4 plus proches voisins, il y a deux paramètres à estimer dans chacune des zones.

Enfin, la pénalité est choisie proportionnelle à la dimension avec un coefficient de proportionnalité λ . Ainsi, le calcul du critère pour le modèle m se décompose en calculs indépendants portant chacun sur une zone de la partition R .

$$\begin{aligned} \text{crit}(m) &= \log(\hat{f}_m(I)) + \lambda \dim(m) \\ &= \sum_{i=1}^P \left(\log(\hat{f}_{R_i}(I_i)) + 2\lambda \right) \end{aligned}$$

Nous appelons *coût* de la zone R_i la quantité $\log(\hat{f}_{R_i}(I_i)) + 2\lambda$. La conséquence principale de cette décomposition est que pour comparer deux modèles, il suffira de comparer la somme des coûts des zones non communes à ces deux modèles.

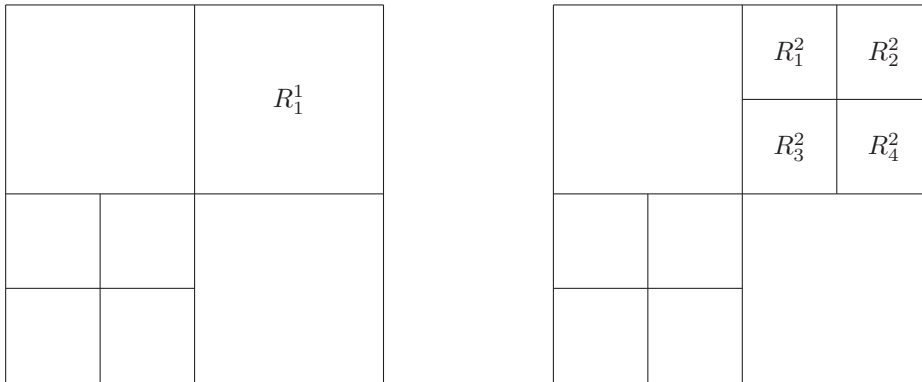


FIGURE 3.4 – Deux exemples de partitions de la collection \mathcal{P}_1 .

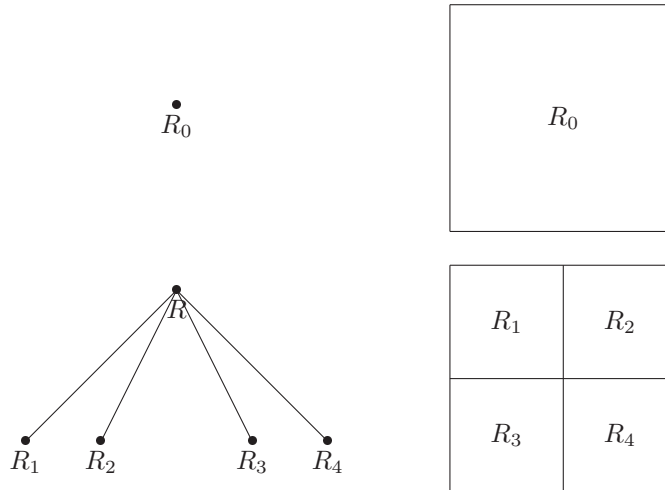
La figure 3.4 présente deux partitions de la collection \mathcal{P}_1 qui ont des zones en commun. Elles définissent deux modèles m_1 et m_2 . Pour choisir entre ces deux modèles, nous devons seulement comparer le coût de la zone R_1^1 sous le modèle m_1 donné par $\log(\hat{f}_{R_1^1}(I_1^1)) + \lambda$ et la somme des coûts des quatre zones R_i^2 sous le modèle m_2 donnée par $\sum_{i=1}^4 \log(\hat{f}_{R_i^2}(I_i^2)) + 4\lambda$.

C'est sur ce principe qu'est basé l'algorithme CART (pour Classification And Regression Trees) [BFSO84]. Il fonctionne en deux étapes :

- Fabrication d'un arbre dit maximum de profondeur k pour une image de taille $M = 2^k$. Chaque feuille de cet arbre correspond à un pixel de l'image.
- Sélection du sous-arbre de l'arbre maximum pour lequel le critère crit est le minimum.

Arbre maximum

La racine de l'arbre maximum correspond à l'image toute entière. Nous associons à ce nœud la valeur de $\log(\hat{f}_{R_0}(I)) + 2\lambda$ où R_0 correspond à la partition ne contenant qu'une seule zone.



Ensuite, pour la collection \mathcal{P}_1 , ce nœud donne 4 nœuds enfants, chacun représentant un des carrés obtenus après le découpage en quatre de la partition initiale R_0 . Pour la collection \mathcal{P}_2 , ce nœud ne donnera que 2 enfants.

On associe au nœud (ou à la partition) R_i la valeur $\log(\hat{s}_{R_i}(I_i)) + 2\lambda$. Nous continuons à associer à chaque nœud 2 ou 4 enfants jusqu'à obtenir un arbre de profondeur k . Il y a alors autant de feuilles que de pixels.

Chaque partition R de la collection \mathcal{P}_1 (ou \mathcal{P}_2) est représentée par un sous-arbre de l'arbre maximum. Pour obtenir la valeur du critère pour le modèle m défini par cette partition R , il faut additionner les valeurs des nœuds correspondants aux zones de la partition R .

Sous-arbre de critère minimum

Pour obtenir l'arbre pour lequel le coût est minimum, nous partons des feuilles de l'arbre. On décide de fusionner les quatre (ou les deux) nœuds « frères » si la somme de leur coût est plus élevée que le coût de leur « père ». Dans ce cas, les feuilles enfants disparaissent. Si la somme des coûts est plus faible, on ne les fusionne pas. Les branches restent donc dans l'arbre. En revanche, le coût associé au père change : on remplace son coût par la somme des coûts de ses nœuds « enfants ».

La figure 3.5 présente un exemple de recherche de sous-arbre de critère minimum. Dans cet exemple, la somme des coûts des enfants est plus élevée pour le nœud 1 ($c_1 < c_{11} + c_{12} + c_{13} + c_{14}$) et le nœud 4 ($c_4 < c_{41} + c_{42} + c_{43} + c_{44}$). Les quatre enfants des nœuds 1 et 4 sont donc fusionnés. Pour les nœuds 2 et 3, les sommes des coûts de leurs enfants est plus faible. Il n'y a pas de fusion mais le coût associé aux nœuds 2 et 4 sont remplacés respectivement par $\tilde{c}_2 = c_{21} + c_{22} + c_{23} + c_{24}$ et $\tilde{c}_3 = c_{31} + c_{32} + c_{33} + c_{34}$. Enfin, puisque la somme des coûts $c_1 + \tilde{c}_2 + \tilde{c}_3 + c_4$ est plus faible que le coût c , il n'y a pas de nouvelle fusion. Le modèle sélectionné contient alors 10 zones qui forment la partition de droite de la figure 3.4.

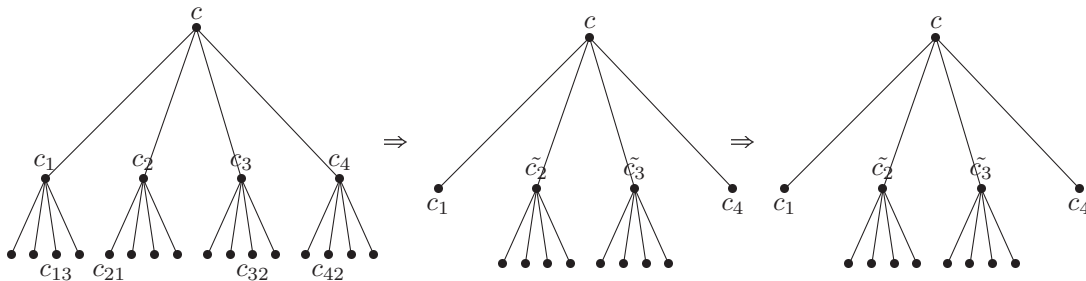


FIGURE 3.5 – Exemple de fusion possible

Pour les partitions de la collection \mathcal{P}_2 , nous appliquons le même algorithme mais cette fois les

nœuds ont 0 ou 2 enfants.

Cette méthode de recherche de minimum permet de diminuer le temps de calcul puisqu'elle évite de calculer plusieurs fois les mêmes coûts.

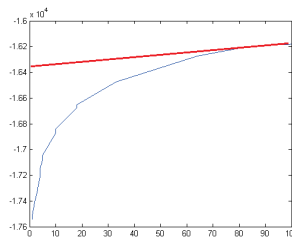
3.3.4 Réglage de la pénalité

Nous avons présenté dans le paragraphe précédent comment sélectionner, en pratique et assez rapidement, le modèle \hat{m} qui minimise le critère $\text{crit}(m) = \gamma_n(\mathbf{x}, \hat{f}_m) + \lambda \dim(m)$.

Nous allons maintenant voir comment choisir la valeur de λ . Cette valeur doit permettre de pénaliser les modèles complexes et ainsi de choisir un modèle qui réalise un bon compromis entre l'explication des données (le biais) et la complexité du modèle (sa variance). Une des méthodes très souvent utilisées est celle de l'heuristique de pente ([Arl07], [BMM11]). Elle est basée sur le fait que le biais se stabilise avec l'augmentation de la dimension du modèle et qu'à partir d'une certaine dimension, seule la variance compte.

Heuristique de pente

1. On trace la courbe qui fait correspondre à la dimension de chaque modèle m sa log-vraisemblance $\gamma_n(\mathbf{x}, \hat{f}_m)$.
2. On estime ensuite la pente de la courbe pour les grandes dimensions. On note cette pente a .



3. On utilise comme constante $\lambda = 2a$. Le modèle sélectionné est alors donné par

$$\hat{m} = \underset{m \in M}{\operatorname{argmin}} [\gamma_n(\mathbf{x}, \hat{f}_m) + 2a \cdot \dim(m)].$$

Il est important de noter que la valeur de la pente a ne dépend que de l'échantillon considéré. Le choix de la pénalité et le calcul du critère sont donc entièrement basés sur les données. Dans la littérature, on parle de pénalité *data-driven* ([Ver09], [AM09]).

Malheureusement, avec les modèles par morceaux que nous utilisons, le biais ne se stabilise pas pour des modèles de grandes dimensions. Nous ne pouvons donc pas appliquer l'heuristique de pente. Nous avons toutefois besoin de pénaliser la log-vraisemblance. Le choix de la valeur de λ se fait donc de manière empirique, c'est-à-dire en testant plusieurs valeurs et en conservant celle qui permettait de sélectionner un modèle pas trop complexe. Après une étude sur les images réelles, nous avons décidé d'utiliser $\text{pen}(m) = 2\dim(m)$.

Nous reviendrons sur cette méthode d'heuristique de pente à la fin du chapitre lors de l'utilisation d'un autre critère que la vraisemblance pour lequel le biais se stabilise.

Nous avons maintenant tous les outils pour pouvoir lancer notre procédure de sélection de modèles (de segmentation) sur des images simulées puis réelles.

3.3.5 Exemples sur images simulées

Nous allons maintenant lancer notre algorithme de sélection de modèles sur des images simulées. Pour ces images, le *vrai* modèle m_0 est connu.

Dans un premier exemple présenté dans la figure 3.6a, le modèle m_0 ne contient que 2 zones. Dans

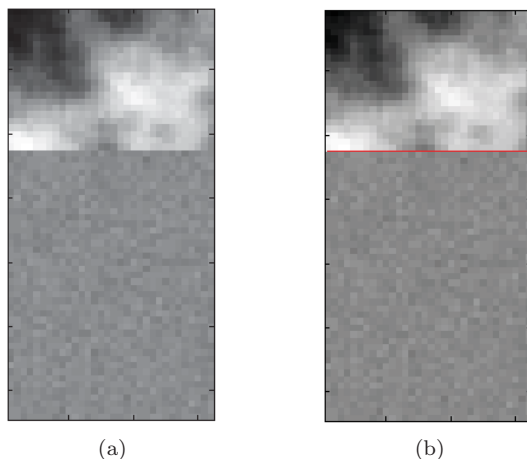


FIGURE 3.6 – Modèles SAR par morceaux simulés avec 1 zone corrélée et une zone non corrélée : (a) réalisation d'un tel modèle, (b) estimation obtenue après sélection de modèle

la partie supérieure de l'image, le champ X est un champ gaussien SAR avec comme paramètres $a_1 = 0.249$ et $\sigma_1 = 1$. Dans la partie inférieure, il s'agit aussi d'un champ gaussien SAR mais avec pour paramètres $a_2 = 0$ et $\sigma_2 = 1$.

On lance alors la procédure d'estimation. Ici, nous avons supposé connu le fait que l'image ne soit composée que de 2 zones et nous avons donc utilisé comme collection de partitions celles ne contenant que deux zones avec une frontière horizontale. Pour les modèles locaux, nous avons conservé l'ensemble \mathcal{S}_1 . Sur la figure 3.6b, nous voyons le modèle sélectionné. La frontière est bien celle du modèle m_0 . D'autre part, pour la partie supérieure du modèle, les paramètres estimés sont $\hat{a}_1 = 0.2490$ et $\hat{\sigma}_1 = 0.9712$. Pour la partie inférieure, les paramètres estimés sont $\hat{a}_2 = 0.02$ et $\hat{\sigma}_2 = 0.9826$.

Avec les paramètres choisis, la corrélation est très forte dans la partie supérieure de l'image et est nulle dans la partie inférieure. On peut alors se demander ce qu'il se passe si la corrélation est forte dans chacune des deux zones.

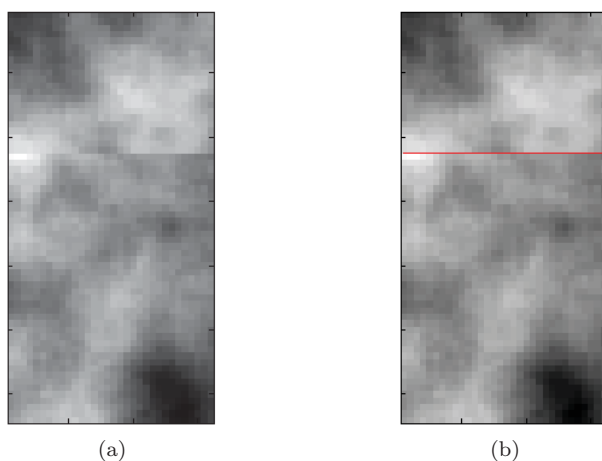


FIGURE 3.7 – Modèle SAR par morceaux avec 2 zones fortement corrélées : (a) réalisation d'un tel modèle, (b) estimation obtenue après sélection de modèle

La figure 3.7a présente un tel exemple. Les paramètres sont $a_1 = 0.249$ et $\sigma_1 = 1$ pour la zone supérieure et $a_2 = 0.249$ et $\sigma_2 = 1$ pour la zone inférieure. La figure 3.7b présente le modèle estimé. On remarque que dans ce cas aussi, la frontière du modèle estimé correspond à la frontière du vrai

modèle. De plus, les paramètres estimés pour chacune des zones sont $\hat{a}_1 = 0.249$, $\hat{\sigma}_1 = 0.9712$, $\hat{a}_2 = 0.249$ et $\hat{\sigma}_2 = 1.0981$. On peut donc dire que ce n'est pas que la différence de corrélation entre les 2 zones qui aide l'algorithme à choisir la meilleure frontière. En revanche, nous avons supposé que les champs dans chacune des zones étaient indépendants. C'est plutôt cette indépendance qui est détectée par l'algorithme de segmentation.

Dans un troisième exemple dont une réalisation est présentée dans la figure 3.8a, le modèle m_0 contient cette fois 4 zones indépendantes. Dans les zones en haut à gauche et en bas à droite, le champ \mathbf{X} est un champ gaussien SAR avec comme paramètres $a_1 = 0.249$ et $\sigma_1 = 1$. Dans les zones en bas à gauche et en haut à droite, il s'agit aussi d'un champ gaussien SAR mais avec pour paramètres $a_2 = 0$ et $\sigma_2 = 1$.

Pour cette image, nous avons lancé l'algorithme d'estimation de modèles avec les deux collections de partitions \mathcal{P}_1 et \mathcal{P}_2 et l'ensemble de modèles locaux \mathcal{S}_1 . Sur la figure 3.8b est représenté le modèle sélectionné pour la collection de partitions récursives dyadiques. Sur la figure 3.8c est représenté le modèle sélectionné pour la collection de partitions récursives split.

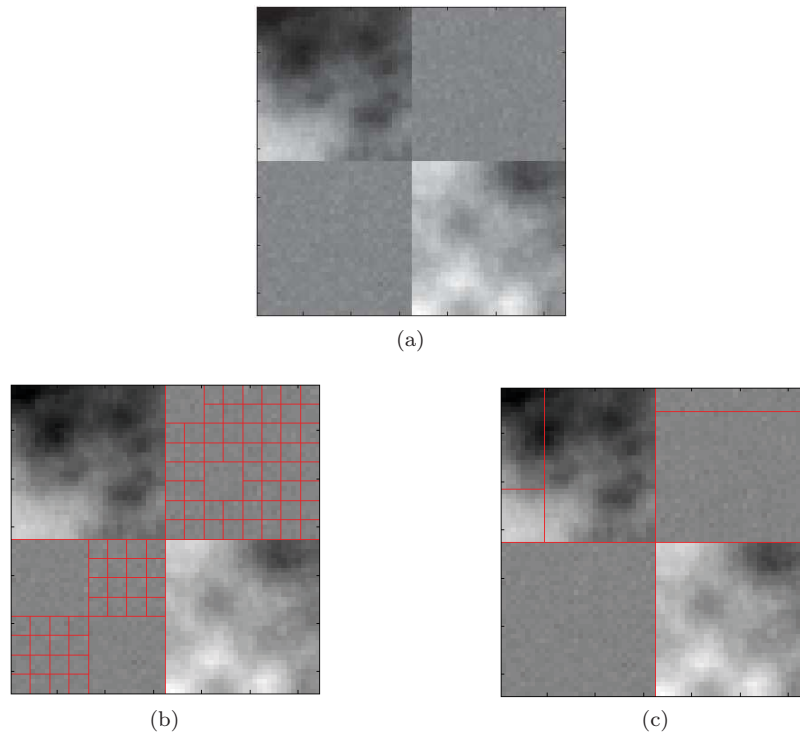


FIGURE 3.8 – Segmentation obtenue avec l'utilisation des partitions récursives dyadiques

La première remarque importante à faire est que le *vrai* modèle m_0 est dans les deux collections. Pourtant, ce n'est pas lui qui est sélectionné. Cela ne veut pas dire que la procédure de sélection de modèles est fautive. Cela veut plutôt dire que pour cette image (pour cette réalisation du modèle m_0), ce n'est pas le modèle m_0 qui représente le mieux les données.

Dans le cas de la collection recursive split, le modèle sélectionné contient le vrai modèle m_0 . En effet, dans les deux zones en haut à droite et en bas à gauche, nous avons utilisé un modèle SAR avec un coefficient $a_2 = 0$, ce qui correspond à supposer les pixels indépendants. Le modèle sélectionné suppose, lui, les pixels dans chacune des zones indépendants, ce qui est bien vrai vu le modèle m_0 choisi. Une pénalité plus élevée aurait peut-être conduit à sélectionner m_0 .

Pour la collection recursive dyadic, le modèle sélectionné contient bien les frontières du modèle m_0 . Dans la zone en haut à gauche, ce modèle suppose deux zones indépendantes qui ne le sont pas sous le modèle m_0 . Le modèle sélectionné est toutefois très proche du vrai modèle m_0 .

3.3.6 Exemples sur images réelles

Regardons maintenant le résultat de l'algorithme de sélection sur des images réelles. Nous avons utilisé l'algorithme avec les deux collections de modèles $\mathcal{M}(\mathcal{P}_1, \mathcal{S}_1)$ et $\mathcal{M}(\mathcal{P}_2, \mathcal{S}_1)$. La figure 3.9 présente quatre images réelles ainsi que les modèles sélectionnés pour chacune d'elles. La colonne centrale correspond aux partitions récursives dyadiques, c'est-à-dire à l'utilisation de la collection $\mathcal{M}(\mathcal{P}_1, \mathcal{S}_1)$. La colonne de droite correspond aux partitions récursives split, c'est-à-dire à la collection $\mathcal{M}(\mathcal{P}_2, \mathcal{S}_1)$.

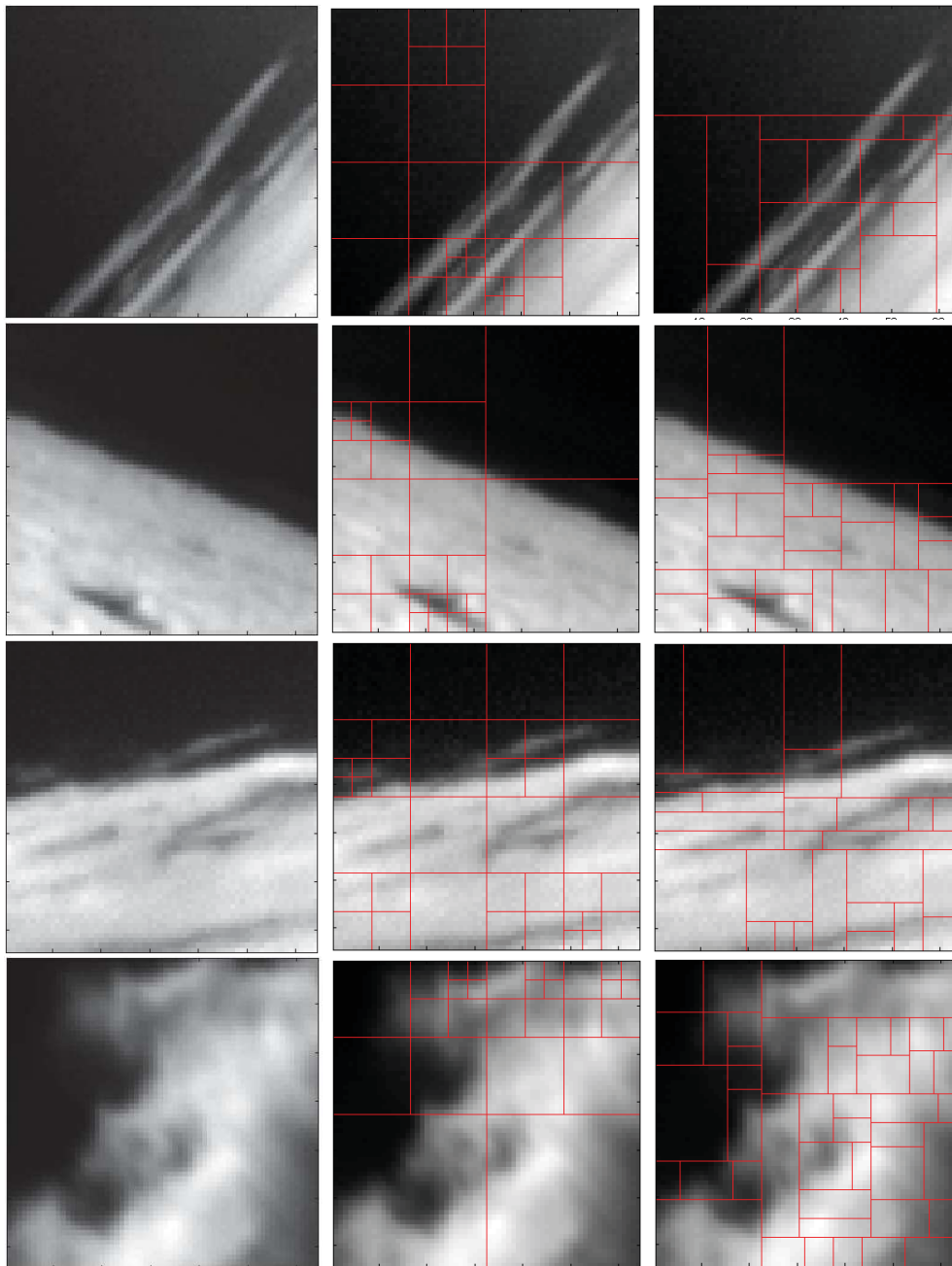


FIGURE 3.9 – Exemples de sélection de partition pour des images réelles.

Pour ces images, le *vrai* modèle m_0 n'est pas connu et nous ne pouvons donc pas comparer les estimateurs obtenus au modèle idéal. Nous pouvons simplement juger de la pertinence du modèle

sélectionné visuellement.

Par exemple, nous pouvons noter que sur l'exemple 3 de la figure 3.9 la frontière horizontale la plus haute correspond bien à la limite supérieure du nuage. De même, sur l'exemple 4, la frontière verticale la plus à gauche pour la collection des partitions récursives split est proche de la frontière naturelle du nuage.

Visuellement, toutefois, nous sommes d'accord pour dire que les partitions des collections \mathcal{P}_1 ou \mathcal{P}_2 sont peu pertinentes pour expliquer les quatre exemples étudiés qui ont des frontières entre les textures obliques ou courbées. En d'autres termes, le modèle inconnu m_0 est assez éloigné des deux collections $\mathcal{M}(\mathcal{P}_1, \mathcal{S}_1)$ et $\mathcal{M}(\mathcal{P}_2, \mathcal{S}_1)$. ceci explique la difficulté de modéliser ces exemples par des modèles des collections envisagées.

La collection des partitions récursives dyadiques est incluse dans la collection des partitions récursives split. Nous savons donc que le modèle récursif split correspond à une valeur du critère plus faible que le modèle récursif dyadique. Autrement dit, il est meilleur pour expliquer les données. Cependant, le temps d'estimation est beaucoup plus faible pour le modèle récursif dyadique. Il faut 34 secondes pour obtenir le modèle récursif dyadique contre 45 minutes pour le modèle récursif split. Nous allons voir dans le prochain paragraphe comment améliorer ce temps de calcul pour un résultat similaire.

3.4 Autre critère : la pseudo log-vraisemblance

Dans l'utilisation du critère de log-vraisemblance pénalisée, la majorité du temps de calcul est utilisée pour calculer la log-vraisemblance. Si \mathbf{Y} est un champ gaussien sur \mathbb{R}^d de moyenne nulle et de matrice de covariance Σ , la log-vraisemblance d'une réalisation \mathbf{y} du champ \mathbf{Y} vaut

$$lv(\mathbf{y}, \Sigma) = -\frac{1}{2} \log(\det(\Sigma)) - \frac{1}{2} {}^t \mathbf{y} \Sigma^{-1} \mathbf{y} = \frac{1}{2} \log(\det(Q)) - \frac{1}{2} {}^t \mathbf{y} Q \mathbf{y}. \quad (3.3)$$

où $Q = \Sigma^{-1}$. C'est le calcul du déterminant qui rend le temps de calcul si long.

En supposant les modèles *par morceaux*, nous avons réduit ce temps de calcul. Pour une partition $R = (R_1, \dots, R_P)$, la vraisemblance devient

$$lv(\mathbf{y}, \Sigma) = \sum_{i=1}^P lv(\mathbf{y}_i, \Sigma_i) = \sum_{i=1}^P \left(-\frac{1}{2} \log(\det(\Sigma_i)) - \frac{1}{2} {}^t \mathbf{y}_i \Sigma_i^{-1} \mathbf{y}_i \right)$$

où les différentes matrices Σ_i correspondent aux blocs diagonaux de la matrice Σ . Ces matrices sont toutes de taille plus petite que celle de Σ et nous réduisons ainsi le temps nécessaire au calcul des déterminants et donc au calcul de $lv(\mathbf{y}, \Sigma)$. Cependant, le gain n'est pas assez fort pour les utilisations en temps réel souhaitées par TOSA.

Dans le but de réduire encore ce temps de calcul, nous allons utiliser un autre critère que celui de la log-vraisemblance.

3.4.1 Définition

Cet autre critère a été proposé par Besag en 1974 [Bes74] : il s'agit du *critère de pseudo log-vraisemblance*.

Définition 11 Pour un champ \mathbf{Y} gaussien sur \mathbb{R}^d de densité f , la pseudo log-vraisemblance d'une réalisation \mathbf{y} est définie comme la somme des log-vraisemblances conditionnelles.

$$PLV(\mathbf{y}, f) = \sum_{i=1}^d \log(f(\mathbf{y}_i | \mathbf{y}_j, j \neq i)).$$

où $f(Y_i | \mathbf{Z})$ représente la densité conditionnelle de la variable Y_i sachant la variable \mathbf{Z} .

La pseudo log-vraisemblance est le plus souvent introduite comme estimateur de la log-vraisemblance

[Bes77]. Elle est très souvent utilisée dans un cadre d'estimation de paramètres. Pour un modèle gaussien paramétrique, il est souvent difficile d'obtenir une formule exacte pour les estimateurs du maximum de vraisemblance. On utilise alors les estimateurs du maximum de pseudo vraisemblance qui sont généralement plus faciles à obtenir.

Reprenons l'exemple d'un champ \mathbf{Y} gaussien de moyenne nulle et de matrice de covariance Σ . Pour une réalisation \mathbf{y} du champ \mathbf{Y} , nous connaissons la valeur de la vraisemblance conditionnelle en fonction de la matrice de précision $Q = \Sigma^{-1}$

$$\forall i \in \llbracket 1, d \rrbracket, f(\mathbf{y}_i | \mathbf{y}_j, j \neq i) = \sqrt{\frac{q_{ii}}{2\pi}} \exp\left(\frac{-q_{ii}}{2} \left(\mathbf{y}_i + \sum_{\substack{j=1 \\ j \neq i}}^d q_{ij} \mathbf{y}_j\right)^2\right)$$

Nous pouvons alors en déduire la valeur de la pseudo log-vraisemblance de la réalisation \mathbf{y}

$$PLV(\mathbf{y}, \phi_{0,\Sigma}) = \sum_{i=1}^d \frac{1}{2} \log\left(\frac{q_{ii}}{2\pi}\right) - \frac{1}{2} \sum_{i=1}^d q_{ii} \left(\mathbf{y}_i + \sum_{\substack{j=1 \\ j \neq i}}^d q_{ij} \mathbf{y}_j\right)^2$$

En comparant cette formule à celle de la log-vraisemblance donnée par la formule (3.3), nous pouvons remarquer que le gain en temps de calcul est dû à l'absence de déterminant dans le calcul de la pseudo log-vraisemblance.

Pour un modèle *par morceaux*, ce critère se simplifie comme dans le cas de la vraisemblance

$$PLV(\mathbf{y}, f) = \sum_{i=1}^P PLV(\mathbf{y}_i, f_i)$$

et nous réduisons encore le temps de calcul.

Sélection de modèles

Nous pouvons alors définir une nouvelle procédure de sélection de modèle basée sur le critère de la pseudo log-vraisemblance pénalisée. Pour une image I et un modèle $m = (R_1, \dots, R_P, S_1, \dots, S_P)$, nous notons $\tilde{f}_i = \operatorname{argmax}_{f_i \in S_1(R_i)} PLV(I_i, f_i)$ l'estimateur par maximum de pseudo-vraisemblance de la

densité de la restriction du champ \mathbf{X} à la zone R_i . Le modèle sélectionné est celui qui minimise la pseudo log-vraisemblance pénalisée.

$$\tilde{m} = \operatorname{argmax}_{m \in \mathcal{M}(\mathcal{P}, \mathcal{S})} \left(\sum_{i=1}^P \log(PLV(I_i, \tilde{f}_i)) + \operatorname{pen}(m) \right)$$

où pen est toujours proportionnelle à la dimension du modèle. Comme pour le critère de la vraisemblance pénalisée, nous avons implémenté cette procédure sous Matlab.

Hypothèse d'indépendance

Pour un modèle stationnaire par morceaux, l'hypothèse d'indépendance entre les zones a été ajoutée pour diminuer le temps de calcul du critère de log-vraisemblance pénalisée et le rendre raisonnable pour une étude empirique.

Avec l'utilisation du critère de pseudo log-vraisemblance, nous ne considérons que les lois conditionnelles $f(\mathbf{y}_i | \mathbf{y}_j, j \neq i)$ qui sont des lois gaussiennes en dimension 1. La structure de covariance définie à travers la matrice Σ n'intervient pas dans le calcul. Nous n'avons donc plus besoin de supposer les modèles *strictement par morceaux*. Nous pourrions ainsi utiliser des modèles dans lesquels la matrice de covariance n'est pas complétée, en dehors de la diagonale, par des blocs de 0.

3.4.2 Exemples sur images simulées puis réelles

Comme nous l'avons expliqué précédemment, cette procédure a l'avantage d'être beaucoup plus rapide en temps de calcul. Nous allons le voir sur quelques exemples d'images simulées puis d'images réelles.

Images simulées

Reprenons un des exemples traités précédemment, celui pour lequel le vrai modèle m_0 est un modèle par morceaux avec dans chaque zone un modèle gaussien SAR aux 4 plus proches voisins. L'appendice C.2 présente l'estimation des paramètres a et σ par maximum de pseudo log-vraisemblance pour le modèle SAR aux 4 plus proches voisins. La critère de pseudo log-vraisemblance vaut alors

$$\text{crit}(m) = \sum_{i=1}^P \frac{\#(R_i)}{2} \left[\log \left(\frac{1 + 4\hat{a}_i^2}{2\pi\hat{\sigma}_i^2} \right) - 1 \right] + 2\lambda|R|.$$

Comme dans le cas de la log-vraisemblance, le paramètre λ est à déterminer. Nous le faisons avec la méthode de l'heuristique de pentes présentée au paragraphe 3.3.4.

La figure 3.10 présente le modèle estimé par le critère de pseudo log-vraisemblance pénalisée pour l'image de la figure 3.8a. Sur la partie droite de l'image, on peut voir le modèle estimé par le critère de log-vraisemblance pour comparaison.

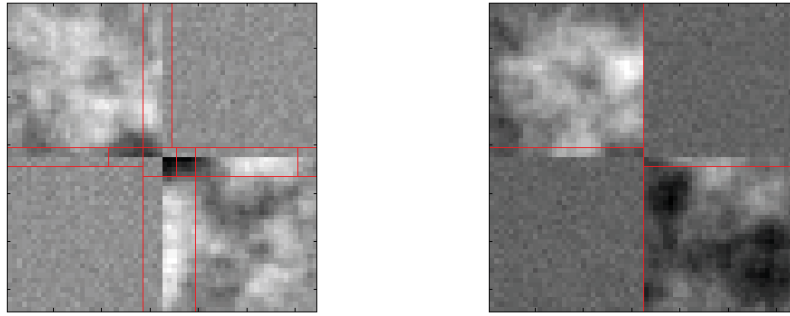


FIGURE 3.10 – Modèle estimé par le critère de pseudo log-vraisemblance pénalisée et par le critère de log-vraisemblance pour l'image de la figure 3.8a

Nous pouvons d'abord remarquer que les deux segmentations proposées sont proches de la segmentation du vrai modèle m_0 . La segmentation obtenue par le critère de log-vraisemblance est toutefois plus précise : même si les frontières ne sont pas toutes exactes, il y a bien 4 zones comme dans le vrai modèle.

En temps de calcul en revanche, c'est le critère de pseudo log-vraisemblance qui est le plus performant. Il faut 45 minutes pour effectuer la recherche du modèle minimisant le critère de log-vraisemblance pénalisée. Il ne faut que 30 secondes pour effectuer cette recherche dans le cas du critère de pseudo log-vraisemblance pénalisée.

Sur cet exemple simulé comme sur les autres que nous avons testés, le critère de pseudo log-vraisemblance pénalisée semble réaliser un bon compromis entre un temps de calcul raisonnable et un résultat proche de celui obtenu par le critère de log-vraisemblance pénalisée.

Images réelles

Revenons maintenant sur les quatre exemples déjà traités pour le critère de log-vraisemblance. Nous avons déjà utilisé la collection des partitions récursives split pour segmenter ces images réelles. La colonne de droite de la figure 3.11 rappelle les résultats obtenus pour le critère de log-vraisemblance pénalisée. Sur la colonne de gauche, se trouvent les modèles sélectionnés par le critère de pseudo log-vraisemblance pénalisée.

Comme nous ne connaissons pas le vrai modèle m_0 , nous ne pouvons pas comparer les modèles sélectionnés au modèle m_0 . En revanche, nous pouvons les comparer entre eux par chacune des procédures et pour une même image initiale.

Les modèles sélectionnés par le critère de pseudo log-vraisemblance pénalisée sont moins complexes que ceux sélectionnés par le critère de log-vraisemblance. L'exemple 4 de la figure 3.11 en est un bon exemple. La pénalité appliquée dans le critère de log-vraisemblance (choisie par nous de la

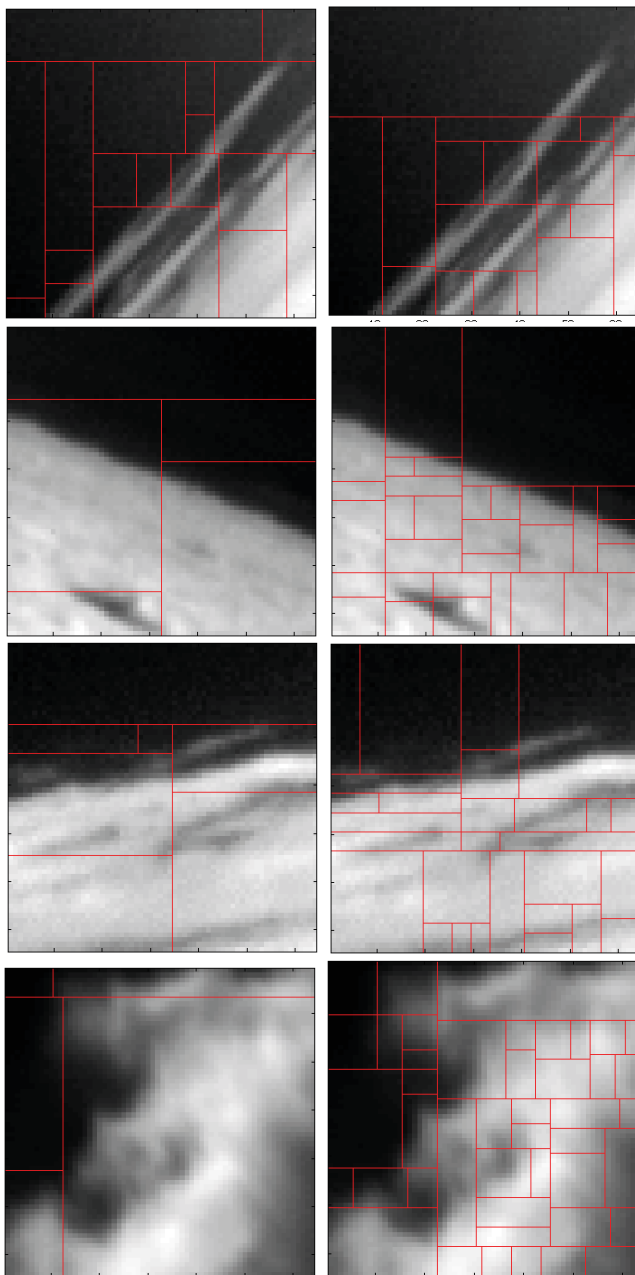


FIGURE 3.11 – Sélection de modèles par pseudo log-vraisemblance pour la colonne de gauche et par log-vraisemblance sur la colonne de droite.

forme $\text{pen}(m)=2\text{dim}(m)$) pourrait sûrement être augmentée pour diminuer le nombre de zones de la segmentation sélectionnée. A l'inverse, celle utilisée dans le critère de pseudo log-vraisemblance (qui est choisie à partir des données seulement) pourrait être plus faible pour favoriser les modèles plus complexes.

Finalement, le modèle sélectionné par le critère de pseudo log-vraisemblance pénalisée est raisonnable visuellement, tout autant que celui sélectionné par le critère de log-vraisemblance pénalisée. De plus, le temps de calcul est bien plus faible. On retrouve donc sur les images réelles le compromis que l'utilisation de la pseudo log-vraisemblance réalisait déjà sur les images simulées entre le temps de calcul et la pertinence du modèle obtenu.

Conclusion

Dans ce chapitre, nous avons présenté les méthodes de sélection de modèles avec le critère de log-vraisemblance pénalisée. Nous avons ensuite reformulé les hypothèses d'un théorème énoncé par Massart. Le théorème 2 que nous en déduisons a permis de justifier le choix d'une pénalité proportionnelle à la dimension du modèle pour obtenir un estimateur dont le risque est proche du risque oracle (c'est-à-dire le plus petit risque possible).

Nous avons également implémenté la procédure de sélection de modèles proposée par Massart après avoir accéléré le parcours de la collection de modèles utilisée. Le temps de calcul est encore trop long pour une application en temps réel. Nous avons donc remplacé le critère de log-vraisemblance pénalisée par un critère de pseudo log-vraisemblance pénalisée. Toutefois, nous n'avons pas de résultats théoriques pour l'utilisation de ce critère.

Dans le chapitre suivant, nous allons présenter la chaîne de détection que nous avons construite à partir des études menées dans les chapitre 2 et 3. Ses performances ainsi que la comparaison à l'existant chez TOSA sont également proposées.

Chapitre 4

Algorithme de détection

Sommaire

4.1 Filtrage proposé	98
4.1.1 Etape 1 : Retrait de la moyenne	98
4.1.2 Etape 2 : Estimation de la matrice de covariance	100
4.1.3 Etape 3 : Retrait de la structure de covariance	101
4.1.4 Définition de la statistique de test	102
4.2 Apprentissage des seuils	104
4.2.1 Etape hors-ligne d'estimation de quantiles	104
4.2.2 Performances de l'étape hors-ligne d'estimation	106
4.3 Prise en compte de la texture locale	108
4.3.1 Descripteurs de Portilla/Simoncelli	108
4.3.2 Coefficients de scattering de Mallat	111
4.3.3 Mélanges de lois gaussiennes	112
4.3.4 Apprentissage par classe	113
4.4 Performances de l'algorithme général	117
4.4.1 Algorithme antérieur	117
4.4.2 Courbes COR pour un rapport signal sur bruit fixé	118
4.4.3 Courbes COR à probabilité de fausses alarmes fixée	121
4.4.4 Images réelles	126

Introduction

Notre objectif est de prendre en compte la covariance spatiale d'une image I (de taille $M \times M$) dans la mise en place du test de détection. Plusieurs travaux ont été menés en supposant les pixels de l'image I indépendants [MRG85], [Vas11]. Nous avons souhaité, au contraire, bâtir un test qui n'utilise pas cette hypothèse comme l'ont fait Kelly [Kel86] ou Genin [Gen13]. Toutefois, nous ne disposons pas d'un échantillon d'images comme eux mais d'une seule image. Nous allons donc faire des hypothèses supplémentaires sur le modèle suivi par l'image I .

Dans le chapitre 2, nous avons expliqué pourquoi il paraissait naturel de travailler avec des modèles stationnaires par morceaux. En effet, la structure de covariance n'est pas la même dans toutes les zones de l'image. Toutefois, nous pouvons supposer le champ localement stationnaire.

Dans le chapitre 3, nous avons présenté des outils statistiques pour sélectionner, à partir d'une image I , un modèle m dans une collection \mathcal{M} pour représenter le contenu de l'image I . Nous avons aussi vu que cette sélection était, en pratique, chère en temps de calcul.

Nous avons alors inversé le problème : au lieu de sélectionner pour chaque image un nouveau modèle, nous avons mené une étude empirique sur une large base d'images pour sélectionner un modèle que nous appliquerons ensuite à toutes les images. En d'autres termes, au lieu de faire la sélection de modèle dans le cœur de l'algorithme, nous avons fait la sélection de modèles en amont de l'algorithme. Une fois ce modèle présenté, nous en déduisons un algorithme de détection. Ses performances de détection seront ensuite comparées à celles de l'algorithme de détection actuellement utilisé chez TOSA.

4.1 Filtrage proposé

Dans le chapitre 1, nous avons étudié la théorie de Neyman-Pearson. Sans l'appliquer directement, cette théorie nous a toutefois servi d'heuristique pour les choix que nous avons faits. Le détecteur que nous proposons a donc la même structure que le test de Neyman-Pearson. Pour une image I , une statistique de test $\Delta(I)$ est calculée. Elle est ensuite comparée à un seuil s . La figure 4.1 présente sous forme de schéma l'approche retenue.

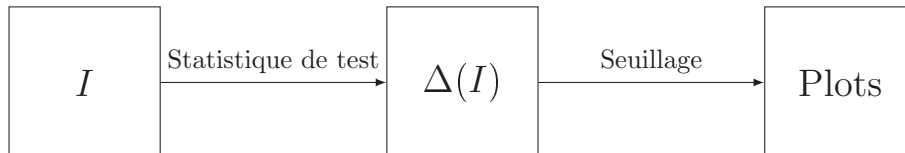


FIGURE 4.1 – Schéma de l'algorithme de détection

Comme dans la théorie de Neyman-Pearson, le choix du seuil s est un point capital pour garantir le taux de fausses alarmes. Nous verrons dans le paragraphe suivant comment s'est fait ce choix.

L'image est toujours supposée être la réalisation d'un champ spatial \mathbf{X} sur la grille $\llbracket 1, M \rrbracket^2$. La statistique de test est définie à partir d'un modèle m supposé pour le champ \mathbf{X} . Nous supposons toujours que ce modèle est gaussien, caractérisé par un vecteur moyenne $\mathbf{m}_0 \in \mathbb{R}^{M^2}$ et une matrice de covariance $\Sigma \in \mathbb{M}_{M^2}(\mathbb{R})$.

4.1.1 Etape 1 : Retrait de la moyenne

Dans le chapitre 2, nous avons présenté différentes approches rencontrées dans la littérature pour estimer le vecteur moyenne \mathbf{m}_0 . Certaines avaient besoin d'hypothèses sur le vecteur \mathbf{m}_0 , comme par exemple les modèles linéaires ou quadratiques proposés par Vasquez [Vas11]. D'autres, en revanche, ne faisaient aucune hypothèse sur le vecteur \mathbf{m}_0 comme l'approche NL-means utilisée par Genin [Gen13].

Dans nos travaux, nous avons décidé de ne pas faire d'hypothèse forte sur la moyenne \mathbf{m}_0 . En revanche, nous avons toutefois supposé qu'elle serait bien estimée par un filtre laplacien.

$$\forall (i, j) \in \llbracket 1, M \rrbracket^2, \widehat{\mathbf{m}}_{i,j} = \frac{X_{i-1,j} + X_{i+1,j} + X_{i,j-1} + X_{i,j+1}}{4}.$$

En soustrayant ce vecteur moyenne estimé à l'image I , nous obtenons une image J qui est la réalisation d'un champ centré \mathbf{Y} défini, à partir du champ \mathbf{X} , par

$$\forall (i, j) \in \llbracket 1, M \rrbracket^2, Y_{i,j} = X_{i,j} - \frac{X_{i-1,j} + X_{i+1,j} + X_{i,j-1} + X_{i,j+1}}{4}. \quad (4.1)$$

Problèmes de bords

Puisque nous travaillons avec des images de taille finie, nous rencontrons dès cette première étape, des problèmes de bords. Dans le chapitre 2, nous avons déjà rencontré ces problèmes de bords et nous avons décidé de remplacer l'ensemble des quatre plus proches voisins par un voisinage appelé *non périodisé*. En effet, pour certains pixels de la grille $\llbracket 1, M \rrbracket^2$, l'ensemble $\{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}$ est mal défini. Pour ces pixels, nous avons restreint le support du filtre laplacien utilisé aux pixels de l'ensemble $\{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}$ qui étaient bien définis. Par exemple, pour estimer la coordonnée (1,1) du vecteur \mathbf{m} , nous n'avons utilisé que deux pixels voisins.

$$\begin{aligned} \widehat{\mathbf{m}}_{1,1} &= \frac{X_{1,2} + X_{2,1}}{2}, \\ Y_{1,1} &= X_{1,1} - \frac{X_{1,2} + X_{2,1}}{2}. \end{aligned}$$

Sur les figure 4.2 et 4.3, nous présentons l'impact de ce filtre sur deux images réelles. Sur la première ligne, nous avons représenté une image réelle et son histogramme. Sur la seconde ligne, nous voyons l'image obtenue à la sortie du filtre laplacien ainsi que son histogramme. Le champ \mathbf{X} a bien été recentré dans les deux exemples.

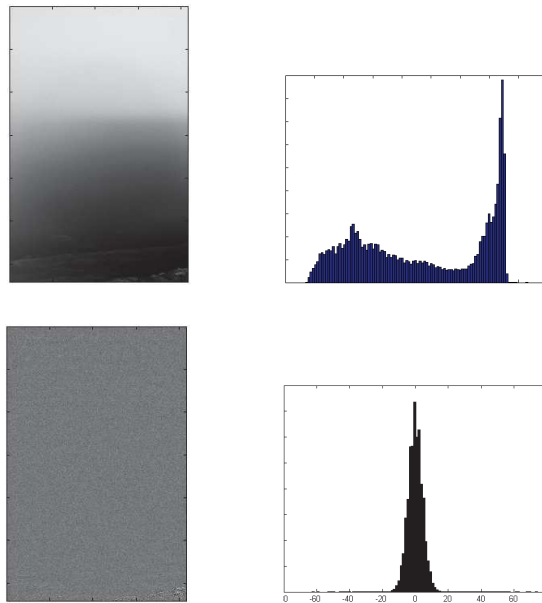


FIGURE 4.2 – Exemple de retrait du vecteur moyenne pour une image de ciel bleu sans nuages.

Puisque nous avons traité les problèmes de bords en introduisant un voisinage non périodisé, le champ \mathbf{Y} obtenu n'est plus un champ stationnaire au sens strict de la définition 2 du chapitre 2. Sa matrice de covariance Σ n'est pas une matrice circulante. Nous allons donc la modéliser autrement.

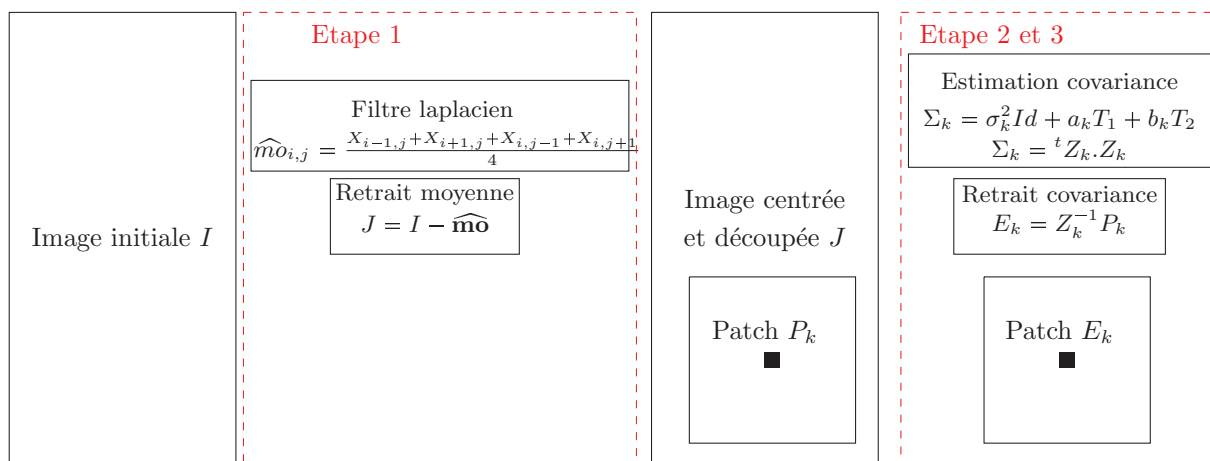


FIGURE 4.4 – Présentation des étapes de filtrage proposé dans le manuscrit

fonction de covariance empirique. Sur la deuxième ligne, sont représentées les images \mathbf{E} obtenues à la sortie des trois étapes de blanchiment ainsi que leur histogramme et leur fonction de covariance empirique. Nous pouvons noter que l'objectif de faire disparaître la structure de covariance a bien été atteint. D'autre part, cet objectif a été atteint pour les deux images alors qu'elles sont de natures différentes. Le filtre proposé semble donc adaptatif, c'est-à-dire qu'il fonctionne sur des images de natures différentes.

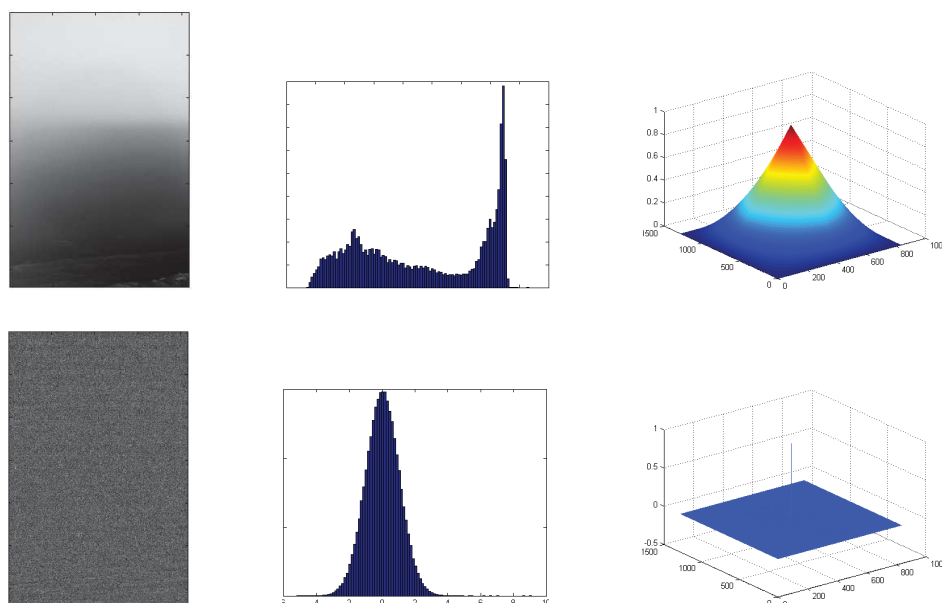


FIGURE 4.5 – Exemple de retrait de la moyenne et de la structure de covariance sur une image de ciel sans nuage.

4.1.4 Définition de la statistique de test

A partir de ce filtre, nous allons maintenant pouvoir proposer une statistique de test pour notre détecteur de cibles sous-résolues.

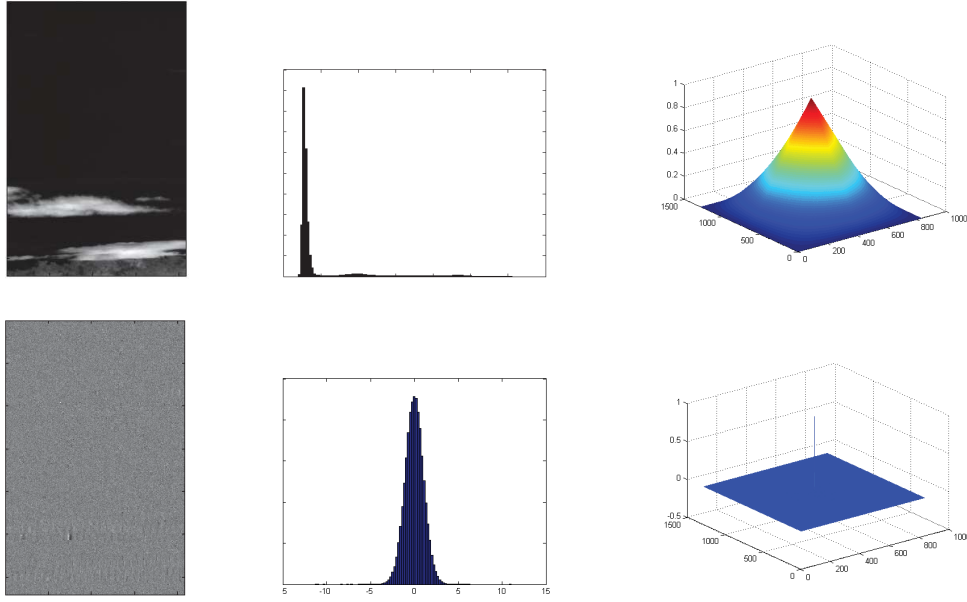


FIGURE 4.6 – Exemple de retrait de la moyenne et de la structure de covariance sur une image de ciel avec nuages.

Pour une image I et un pixel (i, j) , nous souhaitons décider une des deux hypothèses suivantes :

$$H_0^{i,j} : \ll \text{Absence de cible dans le pixel } (i, j) \gg$$

$$H_1^{i,j} : \ll \text{Présence d'une cible d'intensité } c > 0 \text{ dans le pixel } (i, j) \gg.$$

L'image I est d'abord centrée par passage dans le filtre laplacien. Nous notons ensuite $P_{k(i,j)}$ le patch contenant le pixel (i, j) . La statistique de test proposée est la valeur dans le pixel (i, j) à la sortie du filtre. Elle ne dépend que du patch contenant le pixel (i, j) et nous la notons donc $\Delta(P_{k(i,j)}, i, j)$.

$$\Delta(P_{k(i,j)}, i, j) = (\hat{Z}_{k(i,j)}^{-1} \cdot P_{k(i,j)})(i, j).$$

Cette statistique dépend du patch $P_{k(i,j)}$ contenant le pixel (i, j) mais d'aucun autre patch de l'image centrée J . D'autre part, le patch $\hat{Z}_{k(i,j)}^{-1} \cdot P_{k(i,j)}$ ne dépend du pixel (i, j) qu'à travers l'indice $k(i, j)$. Ainsi, pour tester deux pixels appartenant au même patch P_k , nous n'aurons besoin de ne calculer qu'une seule fois le patch $\hat{Z}_k^{-1} \cdot P_k$.

Choix du seuil s

Nous notons T_{TLP} le test qui consiste à comparer cette statistique de test $\Delta(P_{k(i,j)}, i, j)$ à un seuil s . Notre principale contrainte est de proposer un détecteur à taux de fausses alarmes constant (TFAC), c'est-à-dire que pour une image I , nous souhaitons maîtriser le nombre de fausses alarmes. Là encore, nous nous inspirons du test de Neyman-Pearson.

Pour un niveau $\alpha^* \in]0, 1[$ imposé par l'utilisateur, le seuil s doit donc être choisi comme le quantile de niveau α^* de la statistique de test $\Delta(P_{k(i,j)}, i, j)$ sous l'hypothèse $H_0^{i,j}$. Nous avons supposé que chacun des patches P_k suivait une loi gaussienne de moyenne nulle et de matrice de covariance Σ_k . Ainsi, sous l'hypothèse $H_0^{i,j}$,

$$\forall k \in \llbracket 1, N_P \rrbracket, Z_k^{-1} \cdot P_k \sim N(0, Id),$$

$$\forall k \in \llbracket 1, N_P \rrbracket, \forall (i, j) \in P_k, \Delta(P_{k(i,j)}, i, j) \sim N(0, 1).$$

Si nous n'avions pas besoin d'estimer les paramètres du modèle (le vecteur \mathbf{m}_0 ou les différentes matrices Σ_k), nous utiliserions comme seuil le quantile de niveau $1 - \alpha^*$ de la loi gaussienne centrée

et réduite. Plusieurs raisons nous ont poussés à ne pas utiliser ces quantiles.

Tout d'abord, nous ne connaissons pas les matrices Σ_k et nous avons eu besoin de les estimer. Même si l'estimation est bonne, nous ne connaissons pas exactement la loi du patch $\hat{Z}_k^{-1}.P_k$. De plus, nous avons fait des hypothèses assez fortes sur la structure de ces matrices. Même si ces hypothèses ont été justifiées par une étude empirique, elles ne peuvent l'être théoriquement et nous ne connaissons donc plus la loi du patch $\hat{Z}_k^{-1}.P_k$. Enfin, et c'est une raison récurrente lorsque nous travaillons avec des données réelles, il existe un écart entre la réalité (le contenu des images I) et le modèle supposé (gaussien de moyenne nulle et de matrice de covariance Σ). Nous avons donc décidé de ne pas utiliser de quantiles théoriques.

Nous avons plutôt entrepris d'apprendre le comportement de la variable $\Delta(P_{k(i,j)}, i, j)$ sous l'hypothèse $H_0^{i,j}$, plus particulièrement d'estimer ses quantiles pour différents niveaux $1 - \alpha^*$. Ces quantiles empiriques serviront alors de seuils auxquels comparer la statistique de test.

4.2 Apprentissage des seuils

Dans cette partie, nous allons présenter la méthodologie mise en place pour apprendre le comportement de la statistique de test $\Delta(P_{k(i,j)}, i, j)$, plus précisément ces quantiles pour différents niveaux $1 - \alpha^*$. Plusieurs méthodes sont utilisées dans la littérature pour apprendre des lois statistiques. Nous pouvons nous référer à Hastie, Tibshirani et Friedman [HTF01] ou à Vapnik [Vap99]. Dans un cadre de tests statistiques pour la détection de changements dans la valeur des paramètres, Hory et coauteurs [HKC05] ont proposé une méthode d'apprentissage pour le seuil utilisé dans le test. Comme nous, ils souhaitent maîtriser le nombre de fausses alarmes.

Cette étape d'estimation de quantiles va avoir lieu en amont de l'algorithme, dans une phase dite « hors-ligne ». Les quantiles estimés seront ensuite considérés comme des paramètres internes de l'algorithme et ne seront pas réestimés à chaque utilisation de l'algorithme. La figure 4.7 présente le schéma de l'algorithme avec l'étape hors-ligne d'estimation. Pour chaque niveau α^* , nous allons enregistrer un seuil $s(\alpha^*)$ qui sera comparé à la statistique de test lorsque l'utilisateur souhaitera imposer une probabilité de fausses alarmes α^* .

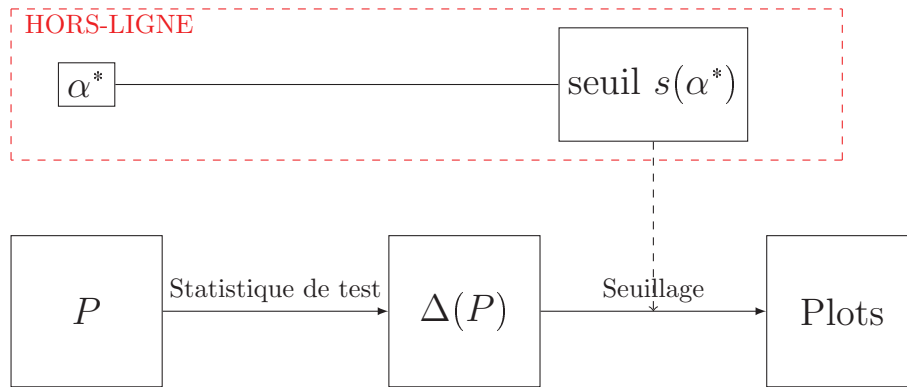


FIGURE 4.7 – Schéma de l'algorithme avec la phase hors-ligne.

4.2.1 Etape hors-ligne d'estimation de quantiles

Notre objectif, lors de cette étape, est idéalement de construire une fonction qui à une probabilité de fausses alarmes α^* associe un seuil q_{α^*} . Ce seuil doit permettre de garantir pour chaque image I son taux de fausses alarmes, c'est-à-dire que q_{α^*} doit être tel que

$$\text{Pour toute image } I, \widehat{\text{pfa}}_n(I, T_{TLP}, q_{\alpha^*}) = \alpha^*.$$

En pratique, nous n'avons pas réussi à construire une telle fonction. Nous nous sommes donnés une famille de probabilité de fausses alarmes

$$\alpha_1^* = 10^{-5}, \alpha_2^* = 2.10^{-5}, \alpha_3^* = 5.10^{-5}, \alpha_4^* = 8.10^{-5}, \alpha_5^* = 10^{-4},$$

et pour chacune de ces probabilités, nous avons estimé un seuil $q_{\alpha_i^*}$.

Cette connaissance probabilité par probabilité au lieu d'une fonction globale $\alpha^* \mapsto q_{\alpha^*}$ a une conséquence importante : si nous souhaitons travailler avec d'autres probabilités de fausses alarmes que les cinq précédentes, nous devons relancer la phase hors-ligne d'estimation.

Estimation de quantiles

Pour estimer les quantiles empiriques de la statistique de test, nous avons besoin d'un échantillon de réalisations de cette statistique. Cet échantillon est dit « d'apprentissage ». Pour cela, nous avons choisi 95 images dans la grande base de données fournie par TOSA. La figure 4.8 en montre quelques unes. Nous avons choisi des images représentatives des images rencontrées chez TOSA. Le nombre de 95 a été un choix arbitraire.

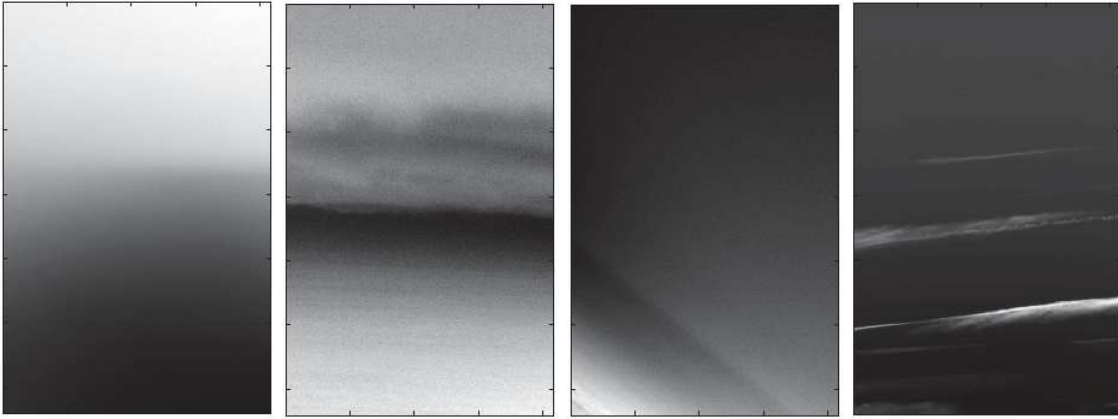


FIGURE 4.8 – Exemples d'images réelles de l'échantillon « d'apprentissage »

Chacune de ces images a été filtrée par le filtre laplacien pour être centrée. Ensuite, nous avons découpé ces images en patches de taille $N = 16$. Nous avons donc obtenu $N_A=98800$ patches que nous avons notés P_1, \dots, P_{N_A} . Puis, pour chacun de ces patches P_k , nous avons estimé une matrice de covariance Σ_k par maximum de vraisemblance sous la contrainte d'appartenir à la structure présentée en (4.2) et nous avons calculé sa décomposition de Cholesky Z_k . Enfin, nous avons fabriqué la statistique de test pour chacun des patches et chacun des pixels.

$$\forall k \in \llbracket 1, N_A \rrbracket, \forall (i, j) \in \llbracket 1, 16 \rrbracket^2, \Delta(P_{k(i,j)}, i, j) = (Z_k^{-1} \cdot P_k)(i, j).$$

Nous disposons finalement d'un échantillon de la statistique de test $\Delta(X)$ de taille $N_A \times N^2 = 25292800$.

Pour chaque probabilité de fausses alarmes α^* , nous définissons le seuil appris q_{α^*} comme le quantile empirique de l'échantillon.

$$q_{\alpha^*} = \min \left\{ t \in \mathbb{R}, \frac{1}{N_A \cdot N^2} \sum_{k=1}^{N_A} \sum_{i,j=1}^N \mathbb{1}_{\Delta(P_{k(i,j)}, i, j) > t} = \alpha^* \right\}.$$

Le tableau de la figure 4.9 donne la valeur de chacun des seuils estimés sur l'échantillon d'apprentissage pour les 5 probabilités de fausses alarmes étudiées.

Ces seuils dépendent entièrement de l'échantillon d'apprentissage. Pour un autre échantillon, nous aurions d'autres valeurs de seuils, mais qui seraient proches de celles-ci.

pfa	0.00001	0.00002	0.00005	0.00008	0.0001
seuils	8.1023	6.3713	5.2127	4.8590	4.7109

FIGURE 4.9 – Seuils estimés à partir de l'échantillon d'apprentissage.

Représentativité de l'échantillon d'apprentissage

Ces seuils estimés vont ensuite être intégrés dans l'algorithme de détection et seront utilisés pour seuiller les statistiques de tests de tous types d'images (ou de patches). Dans la plupart des cas, ils seront utilisés sur des patches n'ayant aucun lien avec les patches (P_1, \dots, P_{N_A}) de l'échantillon d'apprentissage. Nous devons donc nous assurer que l'estimation est robuste et que l'échantillon d'apprentissage est bien représentatif des différents patches rencontrés lors de l'utilisation de l'algorithme. Si, par exemple, l'échantillon d'apprentissage n'est constitué que d'images de ciel bleu, les seuils appris ne garantiront pas le taux de fausses alarmes dans des images de nuages. Pour éviter cela, nous avons porté un soin particulier aux patches que nous avons choisis pour l'échantillon d'apprentissage ainsi qu'à leur nombre qui doit être grand pour assurer la robustesse de l'estimation.

4.2.2 Performances de l'étape hors-ligne d'estimation

Nous souhaitons maintenant vérifier que les seuils appris garantissent bien les taux de fausses alarmes souhaités. Pour cela, nous avons constitué un échantillon de test avec 63 images réelles toutes différentes de celles de l'échantillon d'apprentissage. Une partie de ces images est visible sur la figure 4.10. Ces images sont de la même nature que celles de l'échantillon d'apprentissage.

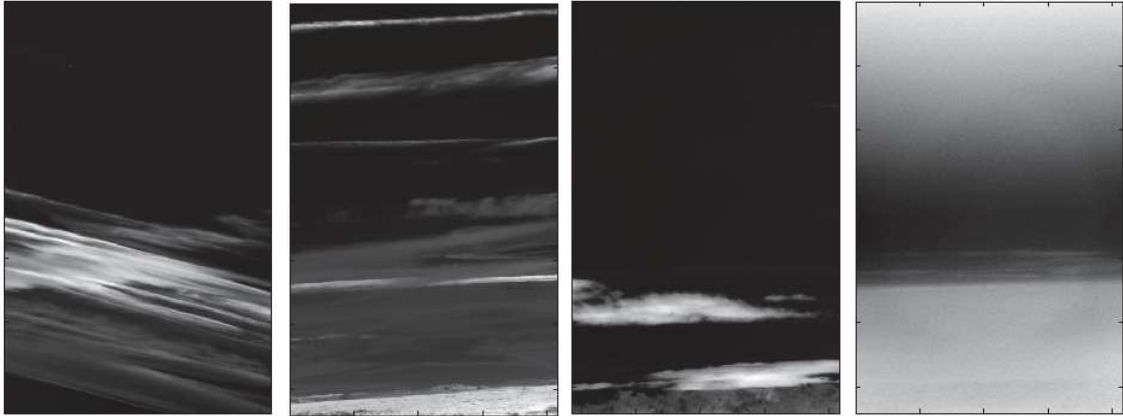


FIGURE 4.10 – Exemples d'images réelles de l'échantillon de test

Nous avons procédé comme pour l'échantillon d'apprentissage. Pour chacun des $N_T=135198$ patches et pour chacun de leurs pixels, nous avons calculé la statistique de test.

$$\forall k \in \llbracket 1, N_T \rrbracket, \forall (i, j) \in \llbracket 1, 16 \rrbracket^2, \Delta(P_{k(i,j)}, i, j) = (Z_k^{-1} \cdot P_k)(i, j).$$

Nous avons ensuite comparé ces statistiques aux seuils appris lors de la phase hors-ligne. Pour chaque seuil $q_{\alpha_i^*}$, nous notons $\hat{\alpha}_i^*$ le taux de fausses alarmes empirique calculé sur l'échantillon test.

$$\hat{\alpha}_i = \frac{1}{N_A \cdot N^2} \sum_{k=1}^{N_T} \sum_{i,j=1}^N \mathbb{1}_{\Delta(P_{k(i,j)}, i, j) > q_{\alpha_i^*}}.$$

Les taux de fausses alarmes empiriques obtenus sont présentés dans le tableau de la figure 4.11 pour les 5 probabilités de fausses alarmes étudiées.

α_i^*	0.00001	0.00002	0.00005	0.00008	0.0001
$\hat{\alpha}_i$	0.000012	0.000022	0.000047	0.000068	0.000082

FIGURE 4.11 – Taux de fausses alarmes empiriques de l'échantillon de test

Les taux de fausses alarmes empiriques sont proches des probabilités de fausses alarmes souhaitées, ce qui était exactement le but recherché. Les seuils appris permettent donc de respecter le taux de fausses alarmes, du moins pour l'échantillon de test. Cela signifie que les patches rencontrés dans l'échantillon d'apprentissage est représentatif de l'échantillon de test. Dans la suite du manuscrit, nous utiliserons les seuils donnés par le tableau de la figure 4.9 lors de l'utilisation de notre algorithme.

Répartition des fausses alarmes

L'expérience précédente démontre que les seuils appris lors de l'étape hors-ligne permettent de garantir les probabilités de fausses alarmes souhaitées sur la totalité d'une image I traitée. Nous pouvons aussi nous intéresser à la répartition des fausses alarmes dans cette image I . La figure 4.12 montre une image réelle de la base de test ainsi que la répartition des fausses alarmes à la sortie du détecteur pour une probabilité de fausses alarmes souhaitée de $\alpha^* = 0.05$.

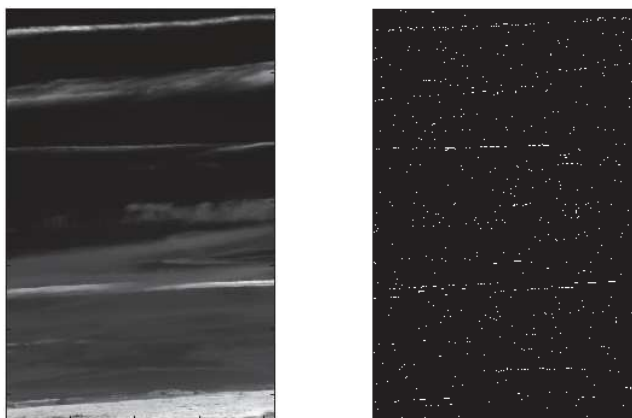
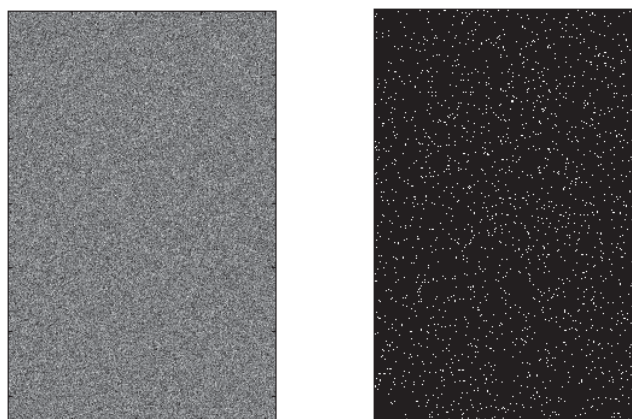


FIGURE 4.12 – Exemple d'une image réelle et la répartition des fausses alarmes

FIGURE 4.13 – Exemple d'une réalisation $N(0, Id)$ et de la répartition des fausses alarmes associée

La figure 4.13 présente, de même, une réalisation d'un champ gaussien $N(0,Id)$ ainsi que la répartition des fausses alarmes associée. Sur cette figure, nous pouvons voir que la répartition des fausses alarmes est uniforme sur toute l'image.

Sur la figure 4.12, en revanche, les fausses alarmes sont plutôt présentes aux bords des nuages. Dans ces zones, les seuils utilisés mériteraient d'être augmentés pour diminuer le nombre de fausses alarmes. Pourtant, ces seuils assurent un taux de fausses alarmes global égal à la probabilité de fausses alarmes souhaitée. Le graphique de gauche de la figure 4.14 permet de schématiser la situation.

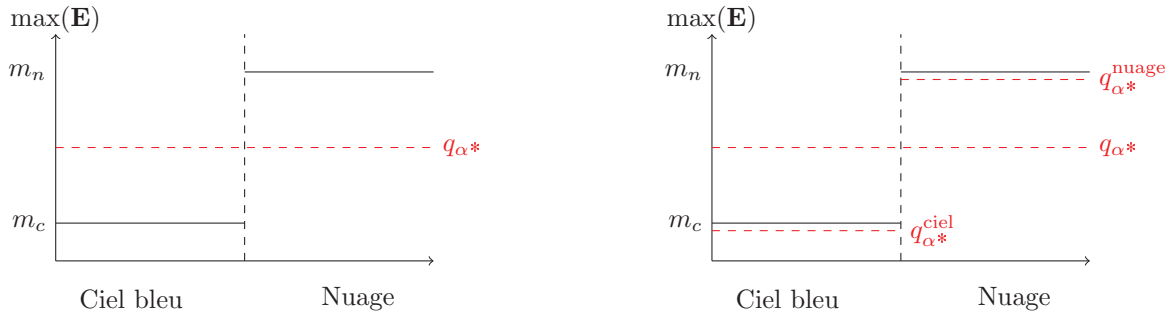


FIGURE 4.14 – Graphiques de la répartition des fausses alarmes pour les deux approches envisagées

Dans le bruit \mathbf{E} récupéré à la sortie du filtre, les grandes valeurs sont atteintes dans des pixels appartenant à des zones de nuages alors que les plus faibles correspondent à des pixels de ciel bleu. Les seuils q_{α^*} correspondent aux quantiles de l'ensemble de l'échantillon et sont donc plus faibles que les plus grandes valeurs prises par le bruit \mathbf{E} . Les fausses alarmes sont donc plus présentes dans les nuages que dans le ciel sans nuages tout en respectant le nombre de fausses alarmes souhaité par l'utilisateur.

Pour remédier à cela, nous pourrions séparer l'échantillon en deux classes : une contenant les patches de ciel sans nuage et l'autre contenant ceux de ciel avec nuages. Nous apprendrions alors deux familles de seuils : une pour les zones de ciel sans nuage $q_{\alpha^*}^{\text{ciel}}$ et une pour les zones avec nuages $q_{\alpha^*}^{\text{nuage}}$. Le taux de fausses alarmes de chacun des sous-échantillon serait respecté, ce qui permettrait de toujours respecter le taux de fausses alarmes global. En revanche, les fausses alarmes seraient mieux réparties. Le graphique de droite de la figure 4.14 permet de schématiser l'approche envisagée.

Dans la suite, nous allons présenter les outils utilisés pour prendre en compte la texture des patches lors de l'apprentissage des seuils. Nous séparerons nos données en plus de deux classes grâce à des outils de classification non supervisée. Cette étape de classification va venir s'ajouter à l'étape hors-ligne déjà présentée sur le schéma de la figure 4.7 pour devenir celui de la figure 4.15.

4.3 Prise en compte de la texture locale

La prise en compte de la texture locale dans un but de segmentation d'images a déjà été envisagée par Unser [Uns95]. Dans cette partie, nous allons classer les patches d'un échantillon d'apprentissage selon leur texture pour ensuite estimer une famille de seuils par texture. Il faut d'abord pouvoir caractériser la texture de chacun des patches. Plusieurs jeux de descripteurs ont été proposés dans la littérature. Nous en avons étudié deux en détails.

4.3.1 Descripteurs de Portilla/Simoncelli

Dans un article datant de 2000, Portilla et Simoncelli [PS00] ont proposé un modèle statistique

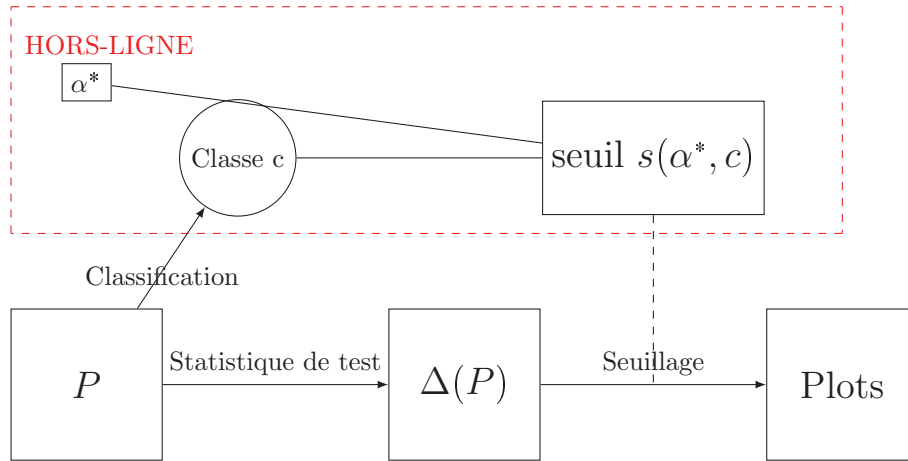


FIGURE 4.15 – Schéma de l'algorithme avec les deux phases hors-ligne

pour la texture d'images. Plusieurs descripteurs statistiques sont calculés sur une image. Leur modèle porte sur cet ensemble de descripteurs.

Décomposition d'une image I

Dans une première étape, une image I est décomposée linéairement à plusieurs échelles et dans plusieurs directions grâce une « pyramide orientable » (*stererable pyramid* en anglais) [SF95]. Comme pour une transformée en ondelettes, l'image est récursivement séparée en un ensemble de sous-images (une pour chacune des K orientations) et une image résiduelle. L'opération est répétée N fois, N étant le nombre d'échelles. Les fonctions de bases d'une telle transformation sont des opérateurs de dérivées directionnelles. La figure 4.16 schématise le diagramme d'une telle transformation.

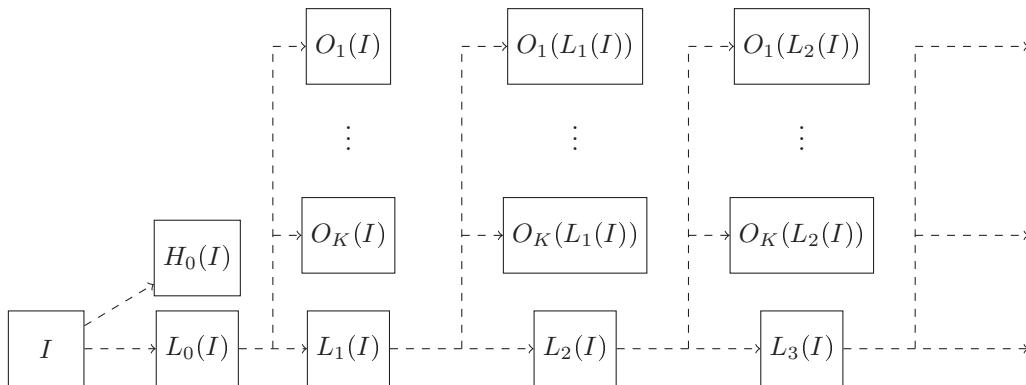


FIGURE 4.16 – Diagramme de la « Pyramide orientable » proposée par Simoncelli et Freeman

Pour chaque échelle i , la somme des sous-images $\sum_{k=1}^K O_k(L_i(I))$ est appelée *image partiellement reconstruite à l'échelle i* .

Cette transformation a l'importante propriété d'être invariante par rotation et par translation, ce qui est très utile en classification. Pour une image I , nous récupérons alors $N \times K + 2$ sous-images : le résidu passe-haut $H_0(I)$, le dernier résidu passe-bas $L_N(I)$ et les images $O_k(L_i(I))$ pour $k \in \llbracket 1, K \rrbracket$ et $i \in \llbracket 1, N \rrbracket$.

Définition des descripteurs

Les descripteurs proposés par Portilla et Simoncelli sont ensuite calculés à partir de l'image I et de ces sous-images obtenues après la décomposition. Ils sont de deux types.

- **Statistiques marginales** : moyenne, variance, valeur minimale, valeur maximale, coefficient d'asymétrie et d'aplatissement de l'image I ; coefficients d'asymétrie et d'aplatissement des images partiellement reconstruites pour chaque échelle ; variance de l'image $H_0(I)$.
- **Coefficients de corrélation** : auto-corrélation locale avec les voisins contenus dans un voisinage de taille $N_c \times N_c$ des images reconstruites à chaque échelle ; corrélation locale avec les voisins contenus dans un voisinage de taille $N_c \times N_c$ de l'amplitude des coefficients de décomposition à chaque échelle et à chaque orientation.

Pour une pyramide à $N = 4$ échelles, $K = 4$ orientations et un voisinage local de taille $N_c = 7$, il y aura 710 coefficients calculés.

Synthèse d'images

Pour justifier la pertinence des descripteurs choisis pour analyser la texture d'une image, Simoncelli [Sim98] a proposé un algorithme de synthèse d'images pour lesquelles la valeur des descripteurs énoncés précédemment était imposée. Par exemple les descripteurs sont calculés sur une image I . Puis, une image ayant la même valeur pour ces descripteurs est synthétisée. Cette synthèse se fait par projection itérative d'un champ gaussien spatial. Le champ gaussien est d'abord projeté sur le sous-ensemble des images ayant la bonne valeur du premier paramètre. L'image obtenue est à son tour projetée sur le sous-ensemble des images ayant la bonne valeur du second paramètre. La projection est répétée pour chacun des paramètres. La comparaison visuelle pour un grand nombre de texture permet alors de justifier les descripteurs choisis. Enfin, ils ont démontré l'utilité de chacun des descripteurs en exhibant pour chacun d'eux une texture mal « reproduite » en l'absence de ce descripteur.

Leur code sous Matlab est accessible depuis le site internet [PS]. La figure 4.17 présente sur la colonne de gauche deux images réelles sur lesquelles nous avons calculé les descripteurs de Portilla/Simoncelli. Dans la colonne de droite sont visibles des images synthétisées avec ces mêmes valeurs de paramètres.

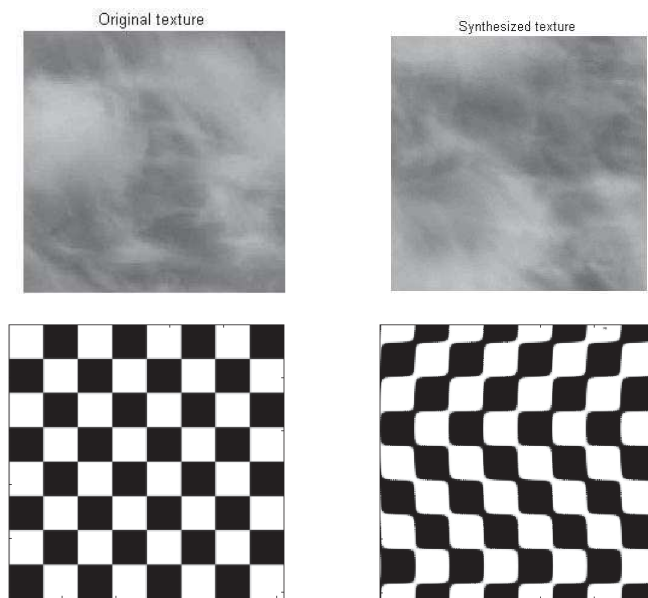


FIGURE 4.17 – Exemples d'images réelles et d'images synthétisées avec les mêmes valeurs pour les descripteurs de Simoncelli/Portilla

Les images synthétisées ne sont pas des copies exactes des images originales. Elles ont simplement les mêmes valeurs des descripteurs. Nous voyons toutefois sur la figure 4.17 qu'elles partagent la même

texture. Nous pouvons utiliser cette méthode de synthèse pour fabriquer des copies indépendantes de certaines textures et ainsi fabriquer un échantillon d'une même texture par exemple. Ces descripteurs ont également été utilisés pour débruiter des images [PSWS03].

4.3.2 Coefficients de scattering de Mallat

MD'autres descripteurs ont été proposés par Bruna et Mallat ([BM11],[BM13]). L'objectif est toujours de trouver des descripteurs stables par translation, rotation et par déformations locales. De plus, ils cherchent à proposer des coefficients qui préservent l'information liée aux hautes fréquences dans un but de classification.

Les coefficients de Fourier n'ont pas cette dernière propriété. Il en résulte que plusieurs textures discernables à l'œil nu ont les mêmes coefficients de Fourier. Les coefficients en ondelettes, eux, sont stables par déformations locales mais pas par translation.

Les *coefficients de scattering* proposés par Bruna et Mallat sont calculés à partir des coefficients en ondelettes pour plusieurs directions et plusieurs échelles auxquels sont appliqués des transformations comme le module complexe ou des opérateurs de moyenne locale.

Pour un signal f (en dimension 1 ou 2) et pour un ensemble de directions $\{\lambda_1, \dots, \lambda_P\}$, nous obtenons pour l'échelle $m = 1$

$$\begin{aligned} \forall i \in \llbracket 1, P \rrbracket, U(\lambda_i)(f) &= |f * \phi_{\lambda_i}| \\ \forall i \in \llbracket 1, P \rrbracket, S(\lambda_i)(f) &= \frac{1}{\mu_i} \int U(\lambda_i)(f) \cdot f \end{aligned}$$

où ϕ est une ondelette choisie par l'utilisateur et $*$ est le symbole de la convolution. A une échelle supérieure m ,

$$\begin{aligned} \forall (i_1, \dots, i_m) \in \llbracket 1, P \rrbracket^m, U(\lambda_1, \dots, \lambda_m)(f) &= ||| |f * \phi_{\lambda_1}| * \phi_{\lambda_1}| * \dots * \phi_{\lambda_m} | \\ \forall (i_1, \dots, i_m) \in \llbracket 1, P \rrbracket^m, S(\lambda_1, \dots, \lambda_m)(f) &= \frac{1}{\mu_{1, \dots, m}} \int U(\lambda_1, \dots, \lambda_m)(f) \cdot f \end{aligned}$$

La figure 4.18 extraite d'un article de Bruna et Mallat présente la décomposition proposée. Cette décomposition est proche dans l'idée et dans la mise en pratique de celle proposée par Simoncelli et Freeman.

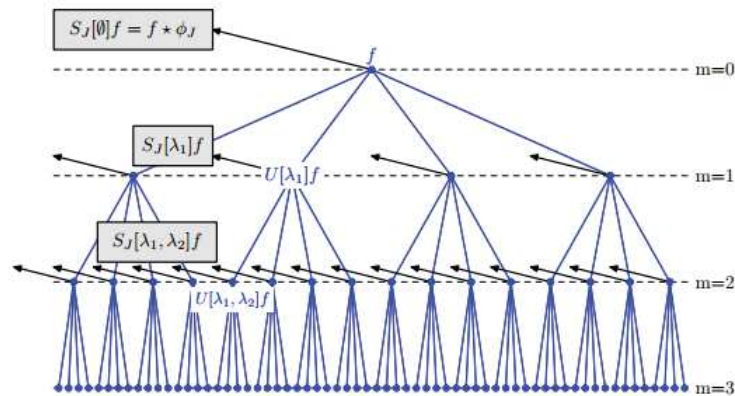


FIGURE 4.18 – Schéma de la décomposition proposée par Mallat

Pour ces descripteurs aussi nous avons pu récupérer le code depuis leur site internet [BM]. Les réglages par défaut fournissent 417 descripteurs pour une image.

Dans nos travaux, nous avons travaillé avec les coefficients de Mallat car les résultats de classification nous ont paru plus pertinents.

4.3.3 Mélanges de lois gaussiennes

Nous souhaitons utiliser coefficients de scattering pour classer les $N_A=98800$ patches de notre échantillon d'apprentissage. Pour cela, nous avons plutôt classé les N_A vecteurs de descripteurs calculés sur les patches de l'échantillon d'apprentissage. En pratique, nous avons réduit la dimension de ces vecteurs en les projetant dans un espace de dimension 10. Nous avons fait cette projection en utilisant une méthode d'analyse en composante principale (ACP). Nous aurions aussi pu le faire en utilisant des projections aléatoires ([Ach03], [BM01]). Ces méthodes ont de bonnes propriétés de conservation de la classification [DG03].

Nous notons $\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_{N_A})$ de tels vecteurs. Nous supposons alors qu'il existe une classification sous-jacente et que ces vecteurs sont des réalisations d'un champ \mathbf{V} sur \mathbb{R}^{10} dont la loi est un mélange de loi gaussienne.

$$f_{\mathbf{V}}(\mathbf{x}|\pi, mo, \Sigma) = \sum_{k=1}^K \frac{\pi_k}{(2\pi)^{\frac{N_A}{2}} \det(\Sigma_k)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}{}^t(\mathbf{x} - mo_k)\Sigma_k^{-1}(\mathbf{x} - mo_k)\right) \quad (4.3)$$

où K et les triplets $\theta = (\pi_k, mo_k, \Sigma_k)_{1 \leq k \leq K}$ sont les paramètres du modèle. Les paramètres (mo_k, Σ_k) sont ceux du modèle gaussien dans la classe k . Les coefficients $(\pi_k, 1 \leq k \leq K)$ représentent la probabilité pour une réalisation \mathbf{v} d'appartenir à la classe k . Enfin, le nombre K correspond au nombre de classes. Dans nos travaux, nous le supposons inconnu et il est donc à estimer également.

Pour estimer ces paramètres et en particulier les proportions $(\pi_k, 1 \leq k \leq K)$, nous souhaitons maximiser la log-vraisemblance de l'échantillon $\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_{N_A})$.

$$\log(f_{\mathbf{V}}(\mathbf{v}|K, \pi, mo, \Sigma)) = \sum_{i=1}^{N_A} \log\left(\sum_{k=1}^K \frac{\pi_k}{(2\pi)^{\frac{N_A}{2}} \det(\Sigma_k)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}{}^t(\mathbf{v}_i - mo_k)\Sigma_k^{-1}(\mathbf{v}_i - mo_k)\right)\right).$$

Nombre de classes connu

Lorsque le nombre de classes K est connu, nous pouvons utiliser un algorithme d'estimation par maximum de vraisemblance proposé par Dempster et coauteurs [DLR77], dénommé *EM* pour Expectation-Maximization. Cet algorithme est basé sur la prise en compte de la structure à données manquantes du problème. Dans nos travaux, en effet, il existe une variable cachée \mathbf{Z} qui associe à chaque réalisation du vecteur \mathbf{V} la classe à laquelle elle appartient. L'algorithme *EM* est un algorithme itératif qui nécessite un critère d'arrêt et une initialisation.

Il a été programmé dans le package de Matlab dédié plus largement aux modèles de mélange MIXMOD (MIXture MODelling). C'est à partir de ce package que nous avons réalisé nos travaux. Nous notons alors θ_{EM}^K l'estimation des paramètres θ obtenue par l'algorithme *EM* sous l'hypothèse d'un modèle à K classes.

Une fois les paramètres estimés, l'affectation de chacun des vecteurs $\mathbf{v}_1, \dots, \mathbf{v}_{N_A}$ à une classe (C_1, \dots, C_K) se fait par maximum à posteriori

$$\mathbf{v}_i \in C_k \text{ si pour tout } k' \neq k, \mathbb{P}(C_k|\mathbf{v}_i, \hat{\theta}_{EMV}) > \mathbb{P}(C_{k'}|\mathbf{v}_i, \hat{\theta}_{EMV}).$$

Nombre de classes inconnu

Dans la pratique, le nombre de classe K est inconnu et nous travaillons alors dans un cadre de classification non supervisée. A chaque valeur de K (1 classe, 2 classes, ...) correspond un modèle de mélange gaussien dont la densité est donnée par (4.3). Nous devons choisir entre ces modèles celui qui est le plus adapté à nos données. Pour cela, nous pouvons utiliser un critère de log-vraisemblance pénalisée comme dans le chapitre 3.

$$\hat{K} = \underset{k \in [K_{\min}, K_{\max}]}{\operatorname{argmin}} \left\{ -\log(f_{\mathbf{V}}(\mathbf{v}_1, \dots, \mathbf{v}_{N_A}|k, \hat{\theta}_{EMV}^k)) + \operatorname{pen}(k) \right\}$$

où $\hat{\theta}_{EMV}^k$ est l'estimateur de θ obtenu par l'algorithme *EM* sous l'hypothèse d'un modèle à k classes. La fonction de pénalité pen ne va dépendre que du nombre de classes k .

Toujours pour tenir compte de la structure à données manquantes de notre problème, un autre critère que celui de log-vraisemblance pénalisée a été proposé. Il s'agit du critère *ICL*, pour *Integrated Completed Likelihood* ([Bau09],[BCG00],[BCSAG08b])

$$\hat{K}_{ICL} = \underset{k \in [K_{\min}, K_{\max}]}{\operatorname{argmin}} \left\{ -2 \log(f_{\mathbf{V}, \mathbf{Z}}(\mathbf{v}_1, \hat{\mathbf{z}}_1, \dots, \mathbf{v}_{N_A}, \hat{\mathbf{z}}_{N_A} | k, \hat{\theta}_{EMV}^k)) + k \log(n) \right\} \quad (4.4)$$

où \mathbf{Z} est la variable aléatoire cachée dont la valeur est la classe à laquelle appartient la variable \mathbf{V} . Cette variable étant inconnue, il est nécessaire d'estimer les réalisations $(\mathbf{z}_1, \dots, \mathbf{z}_{N_A})$ pour pouvoir mettre en place la sélection de modèle (4.4). La méthode est également accessible depuis le package MIXMOD de Matlab. Les observations sont ensuite affectées aux classes du modèle à \hat{K}_{ICL} par la méthode du maximum a posteriori.

Finalement, un algorithme issu de l'EM pour des applications de classifications a été introduit par Celeux et Govaert en 1992 [CG92]. Une étape de classification a été ajoutée entre les deux étapes de l'algorithme *EM*. Cet algorithme est appelé *CEM* pour Classification EM. Il est accessible depuis MIXMOD.

C'est lui que nous allons utiliser pour classer nos vecteurs $\mathbf{v}_1, \dots, \mathbf{v}_{N_A}$.

4.3.4 Apprentissage par classe

Nous allons maintenant appliquer ces outils de classification dans le but d'estimer des seuils pour chacune des classes définies.

Classification obtenue

Nous avons finalement choisi de travailler avec les descripteurs de Stéphane Mallat. Nous avons utilisé l'algorithme CEM avec comme initialisation SMALL-EM. Pour le critère d'arrêt, nous stoppons l'algorithme après 500 itérations ou si le gain en terme de vraisemblance est inférieur à $\epsilon = 10^{-4}$. L'algorithme de classification a formé 8 classes à partir de notre échantillon d'apprentissage. Le tableau de la figure 4.19 récapitule le nombre de patches affectés à chacune de ces 8 classes parmi les $N_A = 98800$ patches.

Classe	1	2	3	4	5	6	7	8
Effectif en patches	20070	35006	5630	10637	14803	7335	1552	3767
% en patches	20.3	35.4	5.7	10.8	15	7.4	1.6	3.8
Effectif en pixels	2428470	4235726	681230	1287077	1791163	887535	187792	455807

FIGURE 4.19 – Tableau récapitulatif des effectifs dans chacune des classes de l'échantillon d'apprentissage

Nous pouvons tout de suite remarquer que cette répartition n'est pas équitable. Les classes 1,2,4 et 5 sont plus représentées que les classes 3,6,7 ou 8. Lorsque nous appliquerons les seuils appris à un échantillon de test, nous reviendrons sur cette répartition non uniforme.

Les figures 4.20 et 4.21 présentent les classifications obtenues pour deux images réelles de la base d'apprentissage. La miniature la plus à gauche correspond à l'image réelle brute. Les 8 miniatures de droite représentent pour chacune des classes (1,2,3 et 4 dans cet ordre sur la première ligne et 5,6,7 et 8 dans cet ordre sur la deuxième ligne) les patches qui y ont été affectés. Par exemple, pour ces deux images, aucun patch n'a été affecté à la classe 2.

En observant ces représentations pour chacune des images de la base d'apprentissage, nous pouvons tenter de caractériser les classes définies par l'outil de classification non supervisée. La classe 6, par exemple, correspond à une classe de bords de nuages tandis que la classe 4 correspond plutôt à une classe de zones stationnaires (nuages ou ciel). Nous n'avons pas mené d'étude assez poussée sur la caractérisation des classes obtenues mais il serait très intéressant de le faire. D'autre part, nous avons vu dans le chapitre 3 sur la sélection de modèles que le choix de la pénalité était

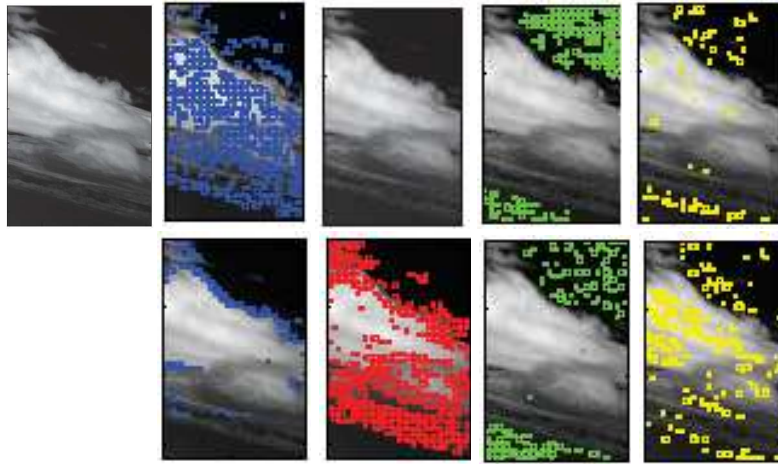


FIGURE 4.20 – Classification obtenue pour les patches issus d’une image de la base d’apprentissage

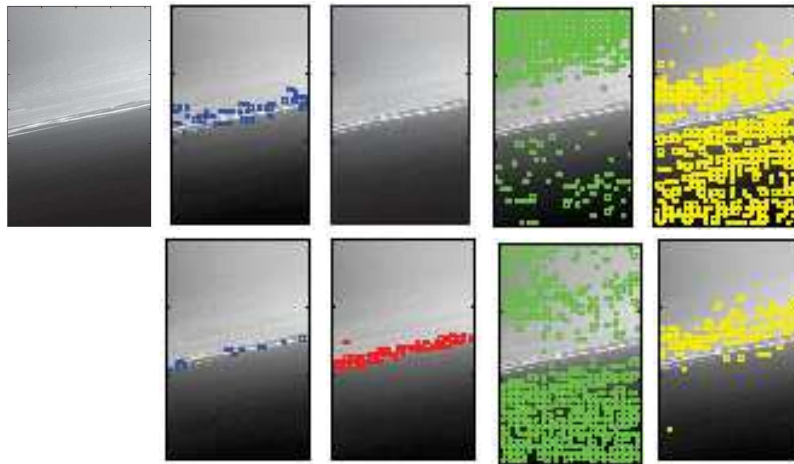


FIGURE 4.21 – Classification obtenue pour les patches issus d’une image de la base d’apprentissage

très important pour favoriser les modèles à petite dimension. Il serait aussi intéressant ici d’étudier l’impact sur la classification obtenue de l’utilisation d’une autre pénalité que celle proposée en (4.4).

Estimation de seuils par classe

Dans chacune des classes C_1, \dots, C_8 , nous avons appliqué la méthodologie présentée au paragraphe 4.2.1. Pour une classe C_l et pour une probabilité de fausses alarmes α^* , nous définissons $q_{\alpha^*}^l$ comme le quantile empirique du sous-échantillon constitué des patches appartenant à la classe l .

$$q_{\alpha^*}^l = \min \left\{ t \in \mathbb{R}, \frac{1}{N_{A'}^l \cdot N^2} \sum_{k=1}^{N_{A'}} \sum_{i,j=1}^N \mathbb{1}_{P_k \in C_l} \mathbb{1}_{\Delta(P_k(i,j), i,j) > t} = \alpha^* \right\}.$$

où $N_{A'}^l$ représente le nombre de patches appartenant à la classe l .

Le tableau de la figure 4.22 donne la valeur des seuils estimés pour chacune des 8 classes et chacune des 5 probabilités de fausses alarmes étudiées : $\alpha_1^* = 10^{-5}$, $\alpha_2^* = 2 \cdot 10^{-5}$, $\alpha_3^* = 5 \cdot 10^{-5}$, $\alpha_4^* = 8 \cdot 10^{-5}$, $\alpha_5^* = 10^{-4}$.

La première remarque à faire est qu’il existe de la variabilité dans les seuils appris. Nous retrouvons bien le comportement schématisé à la figure 4.14. Ensuite, nous pouvons noter que cette variabilité

pfa	0.00001	0.00002	0.00005	0.00008	0.0001
1	8.89	6.91	5.70	5.20	5.00
2	5.95	5.19	4.62	4.38	4.29
3	6.57	5.97	5.29	5.04	4.93
4	9.75	7.30	4.87	4.50	4.35
5	13.41	12.28	9.67	8.68	7.95
6	10.45	8.96	7.43	6.42	6.13
7	7.44	6.09	4.98	4.65	4.50
8	11.88	8.83	5.55	4.91	4.72

FIGURE 4.22 – Seuils estimés dans chacune des classes à partir de l'échantillon d'apprentissage

entre les classes est plus faible pour une probabilité $\alpha^* = 10^{-4}$ que pour une probabilité $\alpha^* = 10^{-5}$. L'apport de l'étape de classification est donc plus important pour les petites probabilités de fausses alarmes.

Cette disparité des seuils va avoir des conséquences sur la probabilité de détection. Sans classification, le seuil estimé pour une probabilité de fausses alarmes de $\alpha^* = 10^{-5}$ vaut 8.1023. Avec classification, ce seuil vaut 5.95 dans la classe 2 et 13.41 dans la classe 5. Ainsi, les cibles contenues dans des pixels de la classe 2 pour lesquelles la statistique de test est comprise entre 5.95 et 8.1023 sont détectées dans l'approche avec classification mais ne sont pas dans l'approche sans classification. La situation inverse se produit dans la classe 7. Les cibles contenues dans des pixels de la classe 5 pour lesquelles la statistique de test est comprise entre 8.1023 et 13.41 sont détectées dans l'approche sans classification mais ne sont pas dans l'approche avec classification. Au global, les performances de détection sont meilleures avec la phase de classification.

Performances de l'étape hors-ligne de classification

Comme dans le paragraphe 4.2.2, nous souhaitons vérifier que les seuils appris garantissent bien le taux de fausses alarmes souhaité. Notre échantillon de test est le même que celui utilisé dans le paragraphe 1.2.2. Il est composé de 63 images, toutes différentes de celles de l'échantillon d'apprentissage. Une partie de ces images est visible sur la figure 4.10.

Nous avons affecté chacun des $N_T=65520$ patches à une des 8 classes (C_1, \dots, C_8) construites lors de l'étape hors-ligne de classification à partir de l'échantillon d'apprentissage. Le tableau de la figure 4.23 récapitule le nombre de patches affectés à chacune des 8 classes.

Classe	1	2	3	4	5	6	7	8
Effectif en patches	2294	31601	7589	6697	807	1429	11866	3237
% en patches	3.5	48.2	11.6	10.2	1.2	2.2	18.1	4.9
Effectif en pixels	587264	8089856	1942784	1714432	206592	365824	3037696	828672

FIGURE 4.23 – Tableau récapitulatif des effectifs dans chacune des classes de l'échantillon de test

Pour chacun des $N_T=65520$ patches et pour chacun de leurs pixels, nous avons calculé la statistique de test

$$\forall k \in \llbracket 1, N_T \rrbracket, \forall (i, j) \in \llbracket 1, 16 \rrbracket^2, \Delta(P_{k(i,j)}, i, j) = (Z_k^{-1} \cdot P_k)(i, j)$$

Nous avons ensuite comparé chaque statistique aux seuils correspondant à sa classe et appris lors de la phase hors-ligne. Pour chaque classe l et pour chaque seuil $q_{\alpha_i}^l$, nous notons $\hat{\alpha}_i^l$ le taux de fausses alarmes empirique calculé sur les patches de l'échantillon test appartenant à la classe l .

$$\hat{\alpha}_i^l = \frac{1}{N_T^l \cdot N^2} \sum_{k=1}^{N_T} \sum_{i,j=1}^N \mathbb{1}_{P_k \in C_l} \mathbb{1}_{\Delta(P_{k(i,j)}, i, j) > q_{\alpha_i}^l}$$

où N_T^l est le nombre de patch affectés à la classe l . Les taux de fausses alarmes empiriques obtenus sont présentés dans le tableau de la figure 4.24 pour les 5 probabilités de fausses alarmes étudiées.

α^*	0.00001	0.00002	0.00005	0.00008	0.0001
$\hat{\alpha}_i^1$	0.000012	0.000018	0.000039	0.000059	0.000081
$\hat{\alpha}_i^2$	0.0000056	0.000012	0.000035	0.000059	0.000076
$\hat{\alpha}_i^3$	0.000011	0.000018	0.000054	0.000083	0.000099
$\hat{\alpha}_i^4$	0.0000029	0.0000035	0.000027	0.000048	0.000064
$\hat{\alpha}_i^5$	0.00020	0.00030	0.00054	0.00062	0.00077
$\hat{\alpha}_i^6$	0	0.0000055	0.000019	0.000027	0.000033
$\hat{\alpha}_i^7$	0.0000066	0.000011	0.000043	0.000071	0.000098
$\hat{\alpha}_i^8$	0	0.0000012	0.00003	0.000048	0.000058

FIGURE 4.24 – Taux de fausses alarmes empiriques par classe pour l'échantillon de test

Plusieurs comportements sont alors visibles. Dans la classe 1 par exemple, les taux de fausses alarmes sont proches des probabilités de fausses alarmes souhaitées. Les seuils appris sont donc pertinents pour le test de détection. Dans la classe 6, les taux de fausses alarmes sont plus faibles que les probabilités de fausses alarmes souhaitées. Les seuils mériteraient, *pour cet échantillon*, d'être plus faibles. A l'inverse, dans la classe 5, les taux de fausses alarmes sont plus élevés que les probabilités de fausses alarmes souhaitées. Les seuils mériteraient, toujours pour cet échantillon, d'être plus élevés.

Nous nous attendions à obtenir des taux de fausses alarmes proches des probabilités de fausses alarmes souhaitées pour chacune des classes (comme pour la classe 1). Nous pensons que ces taux éloignés de ceux attendus peuvent s'expliquer de plusieurs façons. Tout d'abord, l'échantillon d'apprentissage a été divisé en 8 classes et nous disposons donc, dans chacune des classes, de moins de réalisations que lors de l'apprentissage sur la totalité de l'échantillon. L'apprentissage est alors plus difficile. De même, l'échantillon de test est séparé en 8 sous-échantillons. Dans chacun, le taux de fausses alarmes estimé par α_i^* n'est pas toujours un bon estimateur de la probabilité de fausses alarmes dans cette classe.

En pratique, c'est surtout les résultats de la classe 5 qui nous gênent car c'est la seule classe où l'on dépasse le taux de fausses alarmes autorisés.

Nous pouvons également nous intéresser aux taux de fausses alarmes global, c'est-à-dire calculé directement sur l'ensemble de l'échantillon de test.

$$\hat{\alpha}_i = \frac{1}{N_T \cdot N^2} \sum_{k=1}^{N_T} \sum_{i,j=1}^N \mathbb{1}_{\Delta(P_{k(i,j)}, i, j) > q_{\alpha_i}}$$

Les taux de fausses alarmes empiriques obtenus sont présentés dans le tableau de la figure 4.25 pour les 5 probabilités de fausses alarmes étudiées.

α^*	0.00001	0.00002	0.00005	0.00008	0.0001
$\hat{\alpha}_i$	0.000008	0.000015	0.000043	0.000067	0.000086

FIGURE 4.25 – Taux de fausses alarmes empiriques avec l'étape de classification pour l'échantillon de test

Malgré les disparités entre les classes, nous respectons sur l'ensemble de l'échantillon de test les probabilités de fausses alarmes souhaitées.

Nous garderons alors ces seuils appris pour les performances dans le prochain paragraphe.

4.4 Performances de l'algorithme général

Dans cette partie, nous allons présenter les performances de détection de l'algorithme que nous proposons pour l'échantillon de test utilisé dans les parties précédentes. Il est composé de 63 images réelles dont une partie est visible sur la figure 4.10.

Les performances seront présentées sous forme de courbes COR qui montreront le taux de détection d'un échantillon soit en fonction du taux de fausses alarmes pour un rapport signal sur bruit fixé, soit en fonction du rapport signal sur bruit pour une probabilité de fausses alarmes fixée.

Nous souhaitons aussi comparer notre algorithme à celui utilisé actuellement chez TOSA. Nous allons donc commencer par présenter l'algorithme utilisé antérieurement.

4.4.1 Algorithme antérieur

Cet algorithme est également un algorithme de détection de cibles sous résolues. Il a été développé pour des fonds de ciel bleu uniforme et étudié en détails par Vasquez lors de ses travaux de thèse [Vas11].

Modèle utilisé

Comme dans nos travaux, cet algorithme est basé sur une modélisation gaussienne du fond. La covariance spatiale n'est toutefois pas prise en compte de la même façon. Le modèle est le suivant : pour chaque pixel (i, j) , il existe une fenêtre centrée en (i, j) , notée Ω_r , dans laquelle le champ est stationnaire. En l'absence de cibles, la moyenne du fond y est linéaire (ou quadratique) en les coordonnées des pixels et la matrice de covariance est diagonale

$$\forall(i, j), X_{i,j} = \mu_{\Omega_r}(i, j) + \sigma n(i, j)$$

où $n(i, j)$ est une réalisation d'un champ gaussien centré réduit $N(0, 1)$.

Pour le modèle linéaire, nous avons

$$\mu_{\Omega_r}^{(L)}(i, j) = a_{\Omega_r} i + b_{\Omega_r} j + c_{\Omega_r}$$

Pour le modèle quadratique,

$$\mu_{\Omega_r}^{(Q)}(i, j) = a_{\Omega_r} i^2 + b_{\Omega_r} j^2 + c_{\Omega_r} ij + d_{\Omega_r} i + e_{\Omega_r} j + f_{\Omega_r}$$

Les paramètres sont estimés par maximum de vraisemblance dans la zone Ω_r .

Choix de la zone

Le choix de la zone Ω_r est une étape importante. Si la zone est très grande, les estimations sont meilleures mais le modèle est moins en adéquation avec la totalité de la zone. A l'inverse, si la zone est petite, le modèle sera plus vrai mais les estimateurs moins bons. Il faut donc trouver un bon compromis entre la taille de la zone et l'adéquation aux pixels qui s'y trouvent.

Les zones Ω_r envisagées sont des carrés de taille N centrés sur le pixel (i, j) . Pour chaque taille de voisinage N , l'écart-type $\hat{\sigma}_N$ du niveau de gris des pixels appartenant à la zone est estimé. Le voisinage choisi est le plus grand pour lequel l'écart-type estimé ne dépasse pas un seuil s_C .

$$N_{opt} = \max \{N \mid \hat{\sigma}_N < s_C\}.$$

Le seuil s_C dépend des paramètres intrinsèques du capteur utilisé pour acquérir l'image.

Statistique de test

Nous allons maintenant présenter la statistique de test utilisée dans le cas d'une détection dans le pixel (i, j) . Le modèle envisagé est le modèle linéaire. Une zone $\Omega_r(i, j)$ est choisie autour du pixel (i, j) . Les paramètres $a_{\Omega_r(i, j)}$, $b_{\Omega_r(i, j)}$ et $c_{\Omega_r(i, j)}$ et $\sigma_{\Omega_r(i, j)}$ y sont estimés par maximum de vraisemblance. La statistique de test est alors la suivante :

$$\Delta(I, i, j) = \frac{I_{i,j} - \hat{a}_{\Omega_r(i, j)} i - \hat{b}_{\Omega_r(i, j)} j - \hat{c}_{\Omega_r(i, j)}}{\hat{\sigma}_{\Omega_r(i, j)}}.$$

La figure 4.26 présente sous forme de schéma le calcul de cette statistique de test. Nous pouvons comparer ce schéma à celui de la figure 4.4.

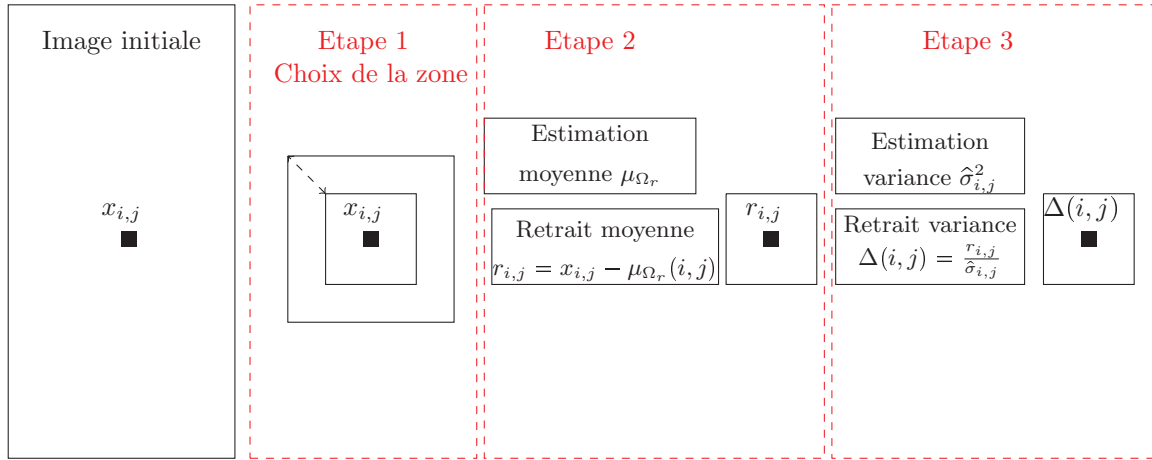


FIGURE 4.26 – Présentation des étapes de l'algorithme antérieur

Cette statistique de test a le net avantage d'être calculable plus rapidement que celle que nous proposons. La moyenne ne dépend des pixels qu'à travers leurs coordonnées tandis que dans notre modèle, elle en dépend via le niveau de gris dans chaque pixel. De plus, nous l'estimons sur toute l'image tandis que dans l'algorithme antérieur, elle est estimée sur la zone $\Omega_{r(i,j)}$. Concernant la matrice de covariance du champ \mathbf{X} , elle est supposée diagonale dans le modèle de l'algorithme antérieur alors que nous lui imposons une structure parcimonieuse.

Le modèle de l'algorithme antérieur est donc plus fin sur la moyenne alors que nous proposons un modèle plus fin sur la matrice de covariance.

Pour effectuer entièrement le test de détection, la statistique $\Delta(I, i, j)$ est comparée à un seuil dont la valeur dépend de la taille de la zone $\Omega_{r(i,j)}$.

4.4.2 Courbes COR pour un rapport signal sur bruit fixé

Nous avons introduit dans le chapitre 1 la notion de rapport signal sur bruit. Nous allons revenir plus en détails sur cette notion.

Rapport Signal sur Bruit (*RSB*)

La statistique de test proposée au début de ce chapitre a été construite à partir du modèle suivi par le champ \mathbf{X} sous l'hypothèse $H_0^{i,j}$. Le but était d'obtenir, sous l'hypothèse que tous les paramètres sont connus, une variable aléatoire qui suit une loi gaussienne centrée réduite. En notant $\Sigma^{-\frac{1}{2}}$ l'inverse de la transformée de Cholesky de la matrice Σ , sous H_0 nous avons

$$\begin{aligned} \mathbf{X} &\sim N(\mathbf{m}\mathbf{o}, \Sigma), \\ (\Sigma^{-\frac{1}{2}}(\mathbf{X} - \mathbf{m}\mathbf{o}))_{i,j} &\sim N(0, 1). \end{aligned}$$

Toujours idéalement, regardons la loi de cette variable sous l'hypothèse $H_1^{i,j}$.

$$\begin{aligned} \mathbf{X} &\sim N(\mathbf{m}\mathbf{o} + c\delta_{i,j}, \Sigma), \\ (\Sigma^{-\frac{1}{2}}(\mathbf{X} - \mathbf{m}\mathbf{o}))_{i,j} &\sim N(c\Sigma^{-\frac{1}{2}}(ij, ij), 1). \end{aligned}$$

où $\delta_{i,j}$ est un vecteur contenant un 1 à la position correspondant au pixel (i, j) et des 0 partout ailleurs. La notation $A(ij, ij)$ correspond au coefficient en position $((i-1)M + j, (i-1)M + j)$ de

la matrice A .

Finalement, l'intensité de la cible c n'intervient qu'à travers la quantité $c\Sigma^{-\frac{1}{2}}(ij, ij)$.

Plaçons nous, cette fois-ci, dans le cas du modèle de l'algorithme antérieur.

$$\mathbf{X}_{\Omega_r} \sim N(\mathbf{m}\mathbf{o}, \sigma^2 Id)$$

où $\mathbf{m}\mathbf{o}$ est linéaire en les coordonnées des pixels. Lorsque tous les paramètres sont connus, la statistique de test est

$$\Delta(\mathbf{X}, i, j) = \frac{\mathbf{X}_{i,j} - a_{\Omega_r(i,j)}i - b_{\Omega_r(i,j)}j - c_{\Omega_r(i,j)}}{\sigma_{\Omega_r(i,j)}}.$$

Sous $H_0^{i,j}$, cette variable suit une loi gaussienne centrée réduite $N(0, 1)$. Sous l'hypothèse $H_1^{i,j}$,

$$\Delta(\mathbf{X}, i, j) \sim N\left(\frac{c}{\sigma}, 1\right).$$

L'intensité de la cible c n'intervient qu'à travers la quantité $\frac{c}{\sigma}$. Nous retrouvons l'espérance de la quantité que nous avons définie comme le rapport signal sur bruit dans le chapitre 1. Il s'agit bien de la définition que Vasquez [Vas11] a utilisée lors de ses travaux. C'est la définition qui est la plus répandue dans la littérature lorsque le fond n'est pas supposé corrélé. Dans le cas d'un fond corrélé, nous n'avons pas trouvé de définition du rapport signal sur bruit dans la littérature. Nous en avons déduit une à partir de celle du modèle de l'algorithme antérieur.

Dans notre modèle, L'intensité de la cible c n'intervient qu'à travers la quantité $c\Sigma^{-\frac{1}{2}}(ij, ij)$. En s'inspirant du modèle de l'algorithme antérieur, c'est cette quantité que nous définissons comme le *rapport signal sur bruit* pour notre modèle.

En conclusion, le rapport signal sur bruit associé à l'algorithme antérieur est le suivant

$$RSB_C(c, i, j) = \frac{c}{\hat{\sigma}_{i,j}},$$

où $\hat{\sigma}_{i,j}$ est la variance empirique du fond estimée sur une fenêtre de taille 11×11 autour du pixel (i, j) . Celui associé à notre modèle est donné par la quantité

$$RSB_{TLP}(c, i, j) = c\hat{\Sigma}^{-\frac{1}{2}}(ij, ij).$$

Dans la suite, pour comparer des algorithmes nous n'utiliserons que la définition du rapport signal sur bruit associé au modèle de l'algorithme antérieur.

Courbes COR globales

Nous allons maintenant tracer les courbes empiriques obtenues sur l'échantillon de test présenté à la figure 4.10. Pour les 63 images réelles (les 98800 patches) qui le composent, nous avons mis en place l'estimation de courbes COR comme décrite au paragraphe 1.3.4 du chapitre 1.

Nous avons travaillé à différentes valeurs du rapport signal sur bruit et différentes probabilités de fausses alarmes de consigne

$$\begin{aligned} r &\in \{1, 2, 3, 4, 5, 6.5, 8, 9, 10\} \\ \alpha^* &\in \{10^{-5}, 2.10^{-5}, 5.10^{-5}, 8.10^{-5}, 10^{-4}\} \end{aligned}$$

Pour les premières courbes nous avons fixé la valeur du rapport signal sur bruit. Soit r la valeur choisie. Nous allons tracer la valeur du taux de détection en fonction de la probabilité de fausses alarmes souhaitée.

Chacune des images de l'échantillon de test contient 1040 patches. Chacun de ces patches est attribué

à une des 8 classes C_1, \dots, C_8 . Dans chacun de ces patches, nous incrustons une cible subpixelique et additive au centre du patch (pixel (8,8) dans le référentiel du patch). Cette cible a un rapport signal sur bruit valant r . Nous appliquons ensuite notre test de détection. Pour chaque probabilité de fausses alarmes α^* , nous utilisons le seuil correspondant à la classe du patch et à la probabilité de fausses alarmes souhaitée. La probabilité de détection est estimée par le nombre moyen de patches dans lequel la cible est détectée.

$$\widehat{pd}(T, r, \alpha^*) = \frac{1}{N_T} \sum_{k=1}^{N_T} \mathbb{1}_{\Delta(P_k, 8, 8) > q_{\alpha^*}^{l_{P_k}}}$$

où l_P est la classe à laquelle appartient le patch P et $q_{\alpha^*}^l$ est le seuil estimé pour la classe l et une probabilité de fausses alarmes souhaitée α^* .

En faisant varier α^* , nous obtenons les courbes de la figure 4.27 pour les rapports signal sur bruit valant $r=4, 8$ ou 10 . Les courbes correspondant aux autres valeurs du rapport signal sur bruit sont données en annexes.

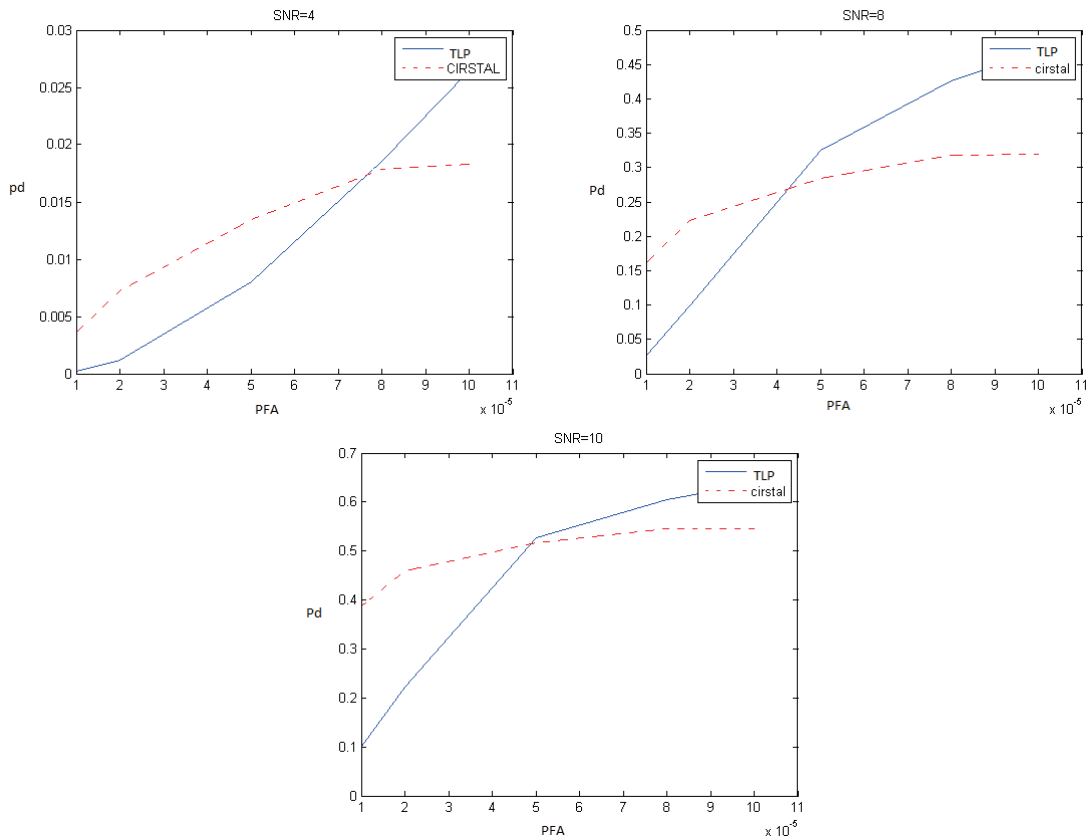


FIGURE 4.27 – Courbes COR pour des valeurs de RSB fixées à 4,8 et 10.

Nous pouvons faire plusieurs remarques sur ces courbes. Tout d'abord, ce sont des courbes COR empiriques, elles dépendent de l'échantillon d'images sur lequel elles sont calculées. D'autre part, nous pouvons voir qu'aucun des 2 algorithmes (TLP ou algorithme antérieur) n'est meilleur que l'autre tout le temps. Pour les petites probabilités de fausses alarmes (inférieure à $5 \cdot 10^{-5}$), c'est l'algorithme antérieur qui a le meilleur taux de détection. En revanche, pour les probabilités de fausses alarmes plus élevées, c'est TLP qui est meilleur.

Il est important ici de rappeler que nous avons appliqué des méthodes d'apprentissage pour régler les seuils de l'algorithme TLP. Or, l'apprentissage de quantiles de niveaux $1-10^{-5}$ nécessite plus de données que l'apprentissage pour des quantiles de niveau $1-10^{-4}$. Notre échantillon d'apprentissage

n'est pas assez important pour permettre une bonne estimation des quantiles par classe pour des niveaux inférieurs à 5.10^{-5} . En revanche, il est assez grand pour estimer correctement les quantiles de niveaux inférieurs. Il serait très intéressant de constituer un échantillon d'apprentissage plus grand et de relancer l'apprentissage des seuils. Malheureusement, le temps de calcul des différentes statistiques de test ne nous a pas permis de le faire. Nous sommes toutefois convaincus de l'amélioration des performances pour de faibles probabilités de fausses alarmes.

Courbes COR par classes

Pour tracer les courbes précédentes, nous avons estimé le taux de détection en faisant la moyenne sur l'ensemble des patchs sans tenir compte de la classe à laquelle ils appartiennent. Nous pouvons aussi nous intéresser au taux de détection par classe. La moyenne n'est alors calculée que sur les patchs appartenant à la classe.

$$\forall l \in \llbracket 1, 8 \rrbracket, \widehat{pd}(T, l, r, \alpha^*) = \frac{1}{N_l} \sum_{k=1}^{N_T} \mathbb{1}_{P_k \in C_l} \cdot \mathbb{1}_{\Delta(P_k, 8, 8) > q_{\alpha^*}^l}$$

où N_l est le nombre de patchs dans la classe l . Nous pouvons alors tracer pour chacune des 8 classes, les courbes COR pour un rapport signal sur bruit fixé. La figure 4.28 présente les courbes COR pour un rapport signal sur bruit de 5 estimée pour les classes 4,5,6,7,et 8.

Là encore, plusieurs remarques sont à faire. Le comportement dans chacune des classes n'est pas le même que pour le comportement global étudié par la figure 4.27. Dans les classes 5 et 6, l'algorithme TLP a de meilleures performances que l'algorithme antérieur. En revanche, dans les classes 4 et 7, c'est l'algorithme antérieur qui est meilleur. En se reportant aux figures 4.20 et 4.21, nous pouvons dire que les classes 5 et 6 sont des classes caractérisées par la présence de frontières de nuages. A l'inverse, les classes 4 et 7 sont plutôt constituées de patchs uniforme (ciel ou nuages). L'algorithme TLP proposé est donc meilleur dans les zones de fonds complexes. La classe 8 est plus difficile à caractériser. Aucun des deux algorithmes ne semble y être meilleur que l'autre pour toute la gamme de probabilité de fausses alarmes visitée.

4.4.3 Courbes COR à probabilité de fausses alarmes fixée

Nous allons maintenant tracer les courbe COR à probabilité de fausses alarmes fixée de notre algorithme pour l'échantillon de test.

Ces courbes sont toujours tracées pour différentes valeurs de probabilité de fausses alarmes fixée par l'utilisateur et pour un rapport signal sur bruit variant entre 1 et 10.

$$\begin{aligned} \alpha^* &\in \{10^{-5}, 2.10^{-5}, 5.10^{-5}, 8.10^{-5}, 10^{-4}\} \\ r &\in \{1, 2, 3, 4, 5, 6.5, 8, 9, 10\} \end{aligned}$$

Courbes COR globales

La probabilité de détection est toujours estimée par le nombre moyen de patchs dans lequel la cible est détectée.

$$\widehat{pd}(T, r, \alpha^*) = \frac{1}{N_T} \sum_{k=1}^{N_T} \mathbb{1}_{\Delta(P_k, 8, 8) > q_{\alpha^*}^{l_{P_k}}}$$

où l_P est la classe à laquelle appartient le patch P et $q_{\alpha^*}^l$ est le seuil estimé pour la classe l et une probabilité de fausses alarmes souhaitée α^* .

En faisant varier r , nous obtenons les courbes de la figure 4.29 pour les cinq probabilités de fausses alarmes souhaitées étudiées.

Plusieurs remarques peuvent être faites à partir de ces courbes. Pour les deux plus faibles probabilités de fausses alarmes, l'algorithme antérieur a de meilleurs performances de détection que

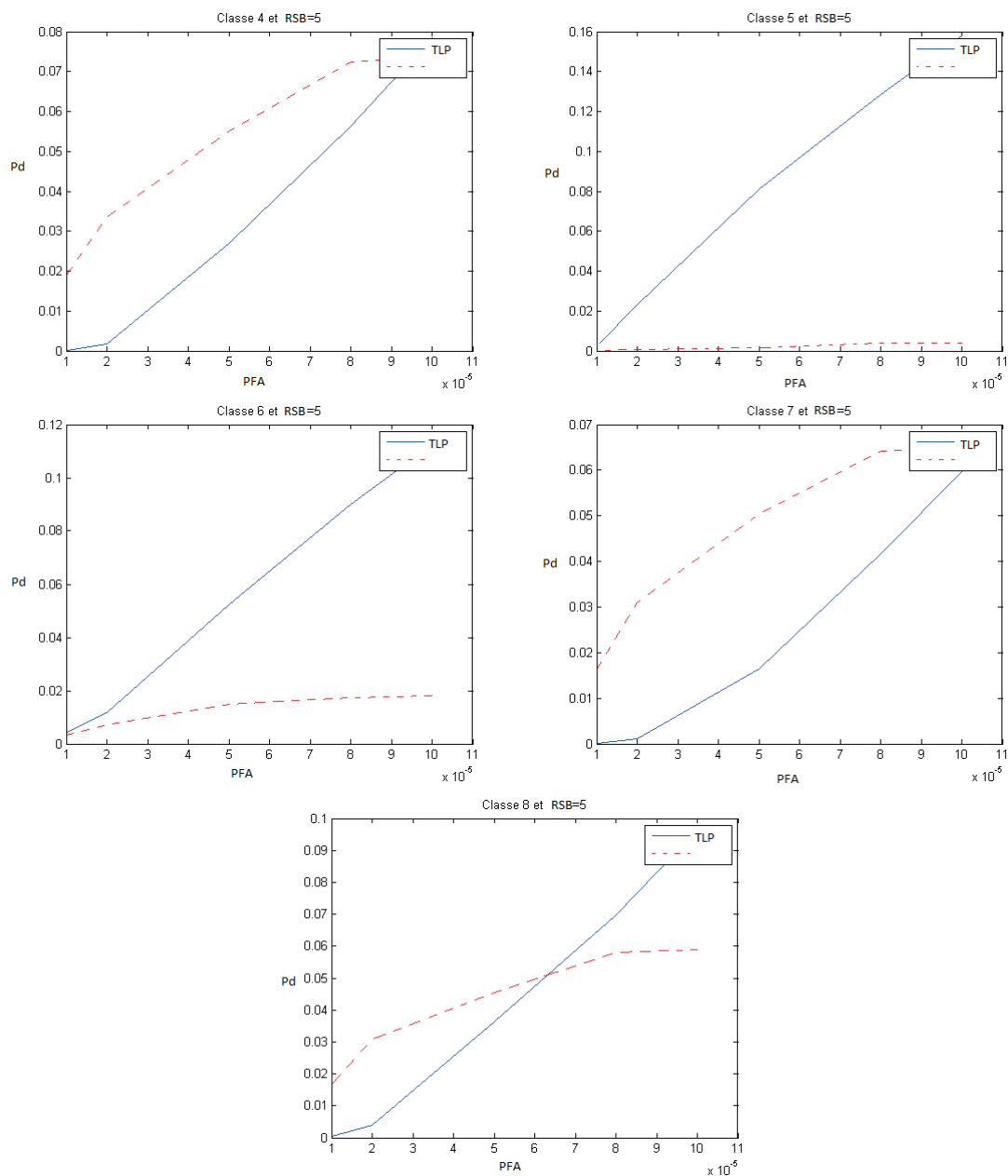


FIGURE 4.28 – Courbes COR pour un RSB de 5 pour les classes 4,5,6,7, et 8.

l'algorithme TLP que nous proposons. Pour les probabilités de fausses alarmes plus élevées, c'est TLP qui est le meilleur. Nous retrouvons bien le même comportement que sur les courbes de la figure 4.27. Ces deux jeux de courbes représentent finalement les mêmes données et il est donc normal d'en déduire les mêmes conclusions.

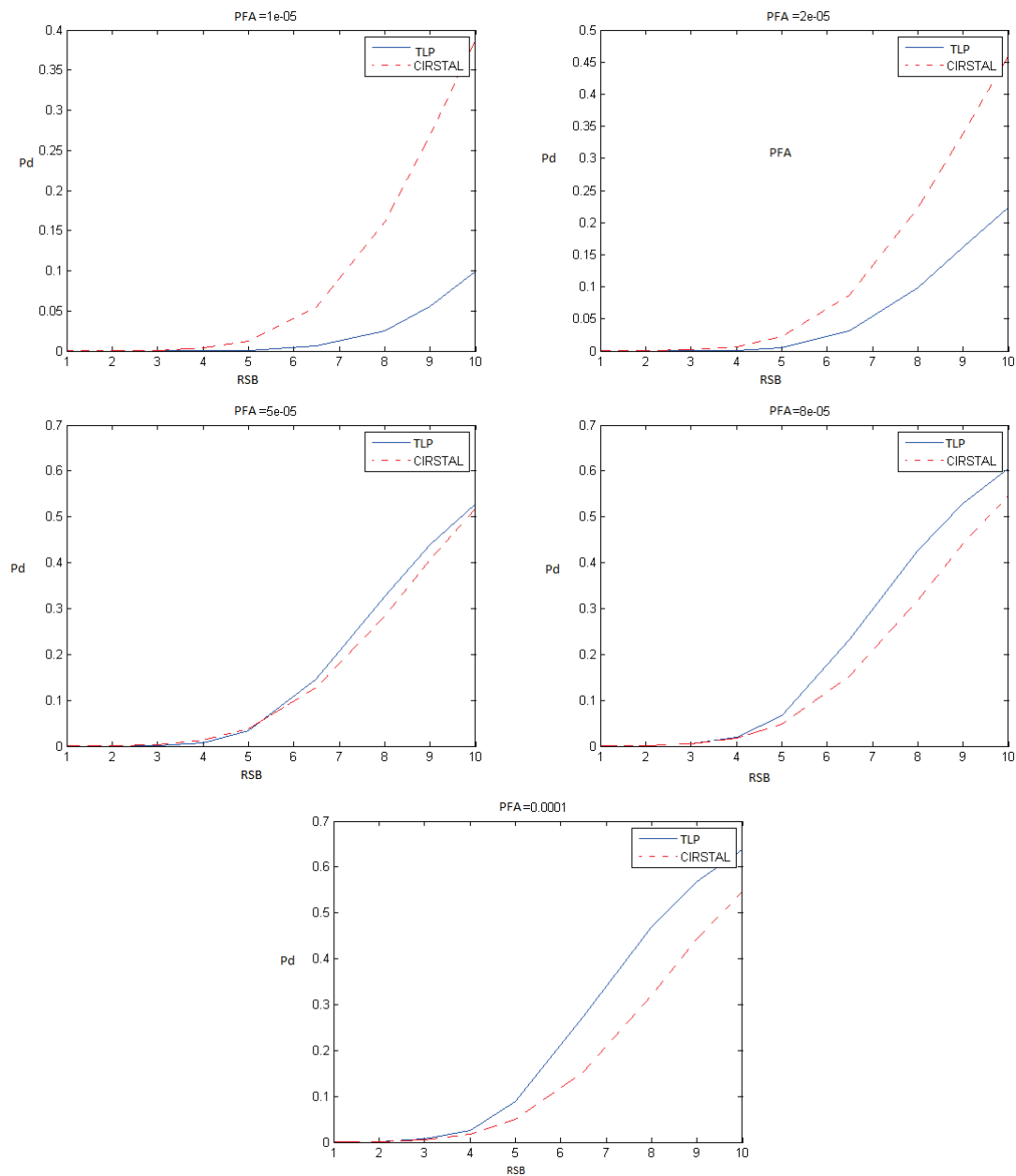


FIGURE 4.29 – Courbes COR pour les cinq probabilités de fausses alarmes étudiées

Courbes COR par classes

Nous nous sommes également intéressés aux courbes COR à probabilité de fausses alarmes fixée pour chacune des classes. Le taux de détection de la classe l est alors estimé en ne tenant compte que des patches affectés à cette classe.

$$\widehat{pd}(T, l, r, \alpha^*) = \frac{1}{N_l} \sum_{k=1}^{N_T} \mathbb{1}_{P_k \in C_l} \mathbb{1}_{\Delta(P_{k(i,j)}, 8, 8) > q_{\alpha^*}}$$

où N_l est le nombre de patches affectés à la classe l .

Les figures 4.30 et 4.31 présentent les courbes COR obtenues pour les classes 4 et 6 pour les cinq probabilités de fausses alarmes étudiées. Les courbes obtenues pour les autres classes sont données en annexe.

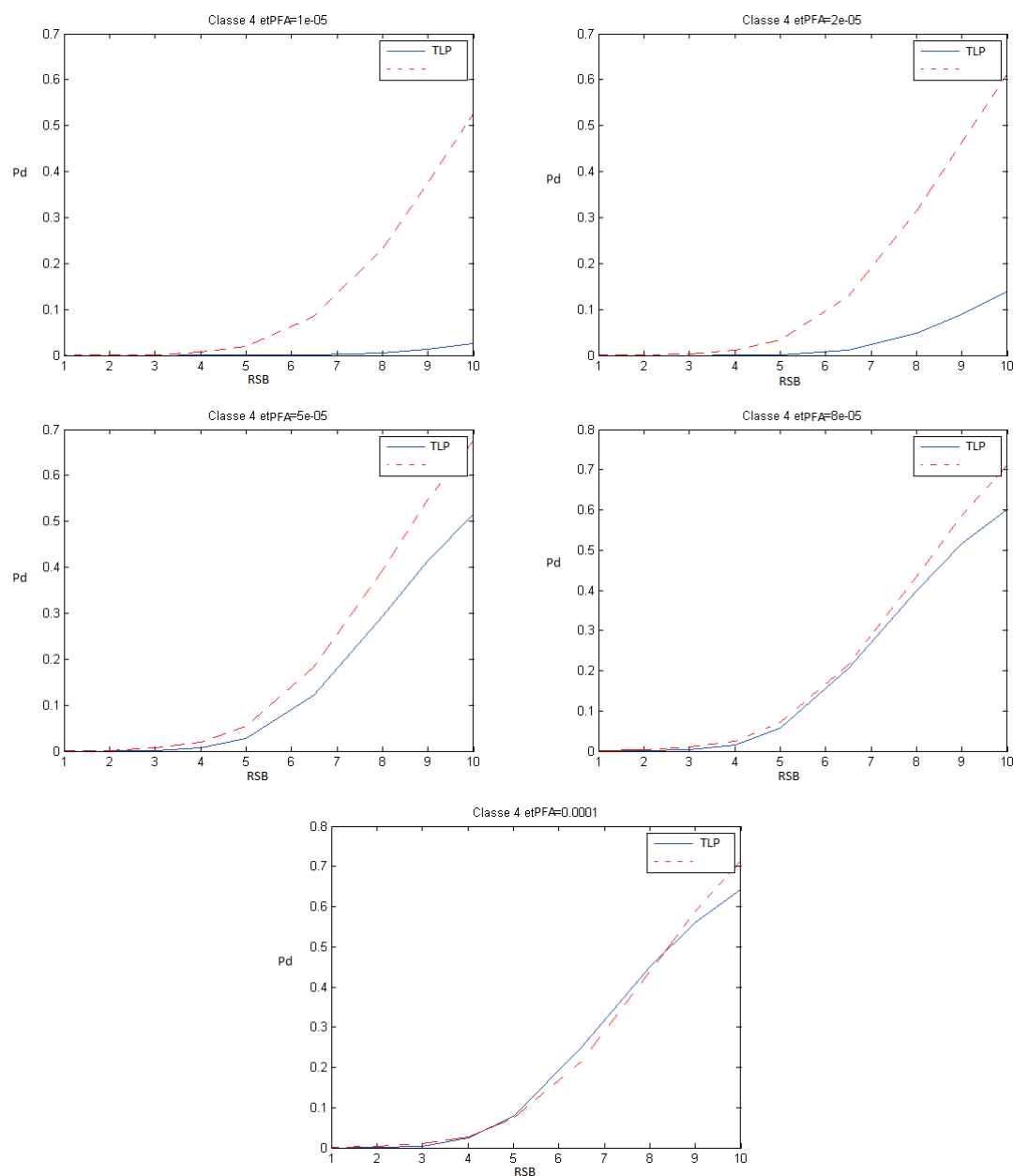


FIGURE 4.30 – Courbes COR pour la classe 4 pour les 5 probabilités de fausses alarmes étudiées.

A partir des courbes pour la classe 4 (figure 4.30), nous pouvons dire que pour des faibles probabilités de fausses alarmes, c'est l'algorithme antérieur qui a meilleures performances de détection. Les différences entre les 2 algorithmes diminuent lorsque la probabilité de fausses alarmes augmente mais qualitativement, c'est l'algorithme antérieur qui est meilleur sur la classe 4. Nous avons déjà fait cette remarque suite à la figure 4.28. Quantitativement, pour atteindre le point de fonctionnement ($p_{fa}=5.10^{-5}$, $p_d=0.5$), il faut un RSB $r = 8.5$ pour l'algorithme antérieur et un RSB $r = 10$ pour TLP soit un gain de rapport 1.2.

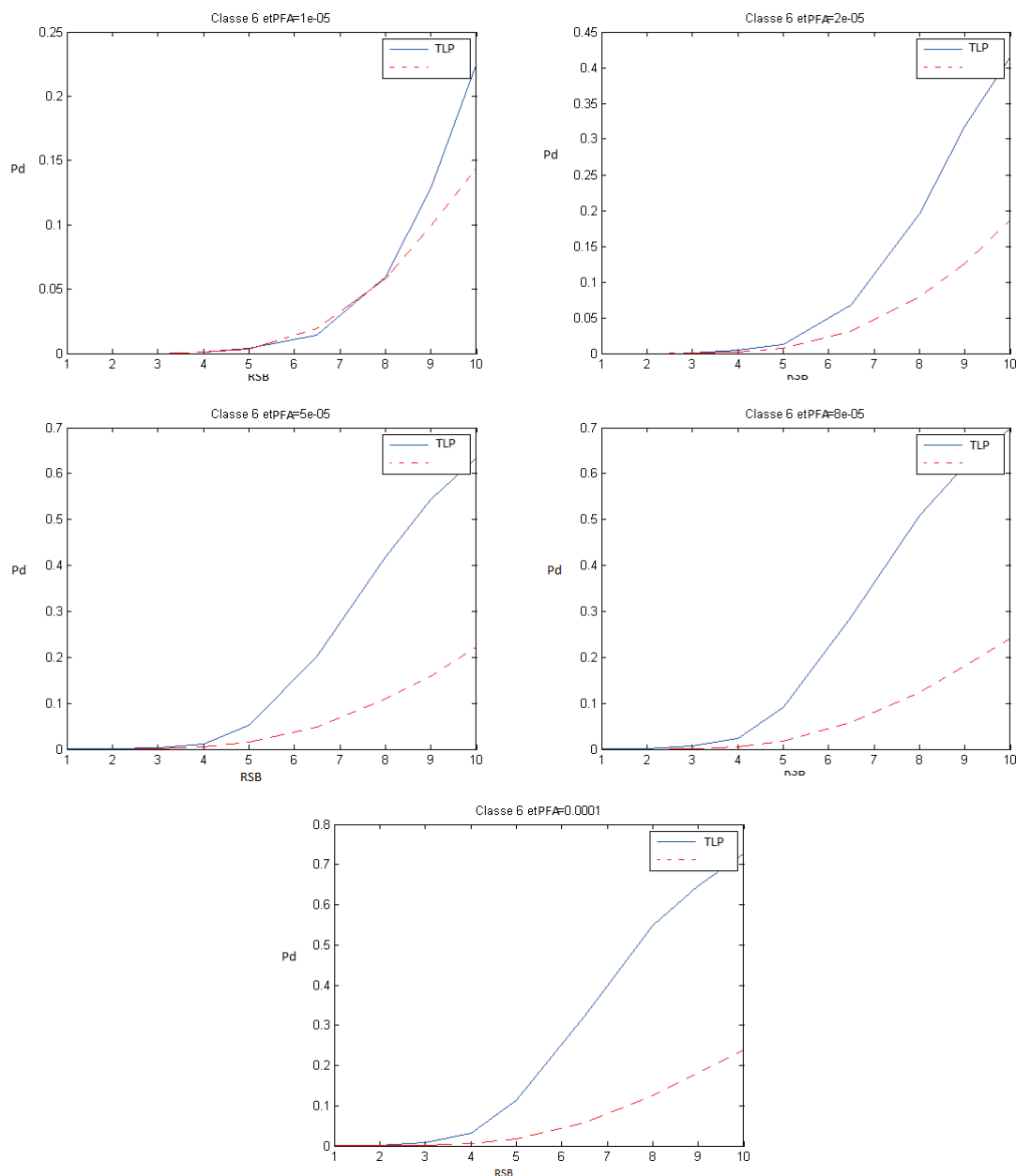


FIGURE 4.31 – Courbes COR pour la classe 6 pour les 5 probabilités de fausses alarmes étudiées.

Dans la classe 6, à l'inverse, c'est TLP qui a de meilleures performances de détection qualitativement. Ces différences augmentent lorsque la probabilité de fausses alarmes augmente. Pour atteindre le point de fonctionnement ($pfa=5.10^{-5}$, $pd=0.5$), il faudra un RSB $r = 8$ pour TLP et un RSB $r > 10$ (non visible sur la courbe) pour l'algorithme antérieur. Le gain aura un rapport bien supérieur à 1.2.

L'ensemble des courbes présentées dans ce paragraphe ou en annexe nous apprend finalement que dans certaines classes, comme la 5 ou la 6, TLP a de meilleures performances de détection. Dans d'autres, comme la 3 ou la 4, c'est l'algorithme antérieur qui a de meilleures performances. Enfin, dans la classe 8, les performances sont équivalentes.

Les deux algorithmes n'ont pas les mêmes performances selon le type de fond. Avec TLP, nous avons d'ailleurs amélioré les performances de l'algorithme antérieur sur fonds nuageux (classe 5 et 6). En

revanche, TLP n'est pas aussi bon que l'algorithme antérieur sur les fonds de ciel bleu uniforme (classe 4). Loin d'être un échec, ce constat permet d'entrevoir une autre amélioration possible : nous pourrions associer les deux algorithmes pour appliquer à chaque patch P l'algorithme qui a les meilleures performances sur la classe à laquelle appartient ce patch. Le schéma de fonctionnement serait alors celui donné par la figure 4.32.

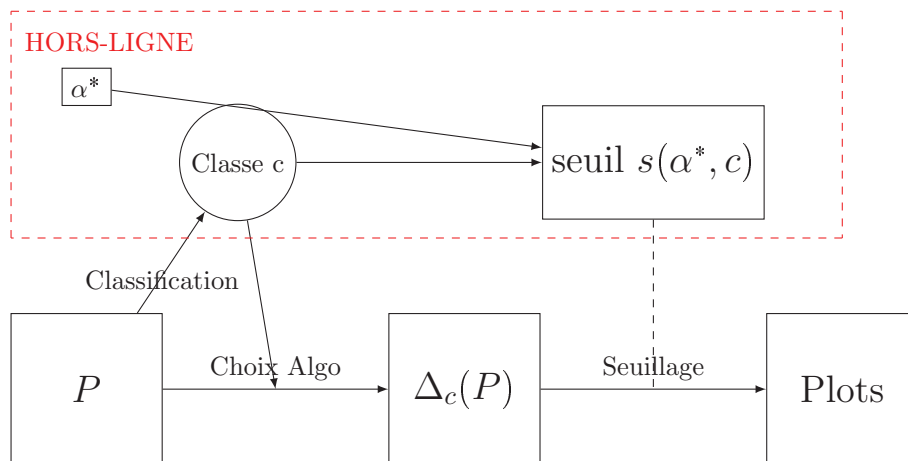


FIGURE 4.32 – Schéma de la chaîne de détection intégrant un choix pour l'algorithme utilisé

Son étude empirique sur données réelles est une des perspectives de nos travaux.

4.4.4 Images réelles

Pour compléter notre étude empirique, nous avons testé notre algorithme sur des images réelles contenant des cibles d'opportunité. Ces cibles étaient présentes dans la scène lors de la capture de l'image par la caméra. Les cibles ont ensuite été labellisées par le service traitement d'images (STI) de TOSA.

La figure 4.33 présente une image réelle contenant une cible d'opportunité au pixel (101,103) ainsi qu'un zoom sur le patch de taille 16×16 contenant la cible.

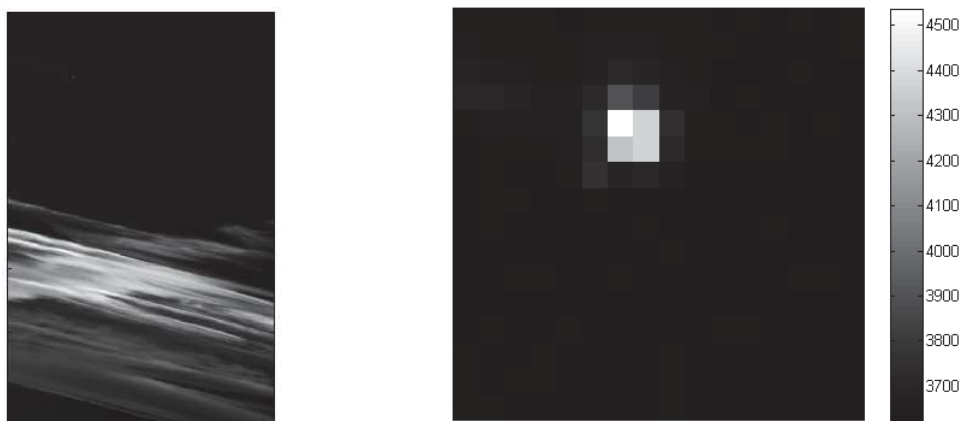


FIGURE 4.33 – Exemple d'image réelle et zoom sur la cible

La première chose que nous pouvons remarquer est que cette cible n'est pas subpixelique. Elle est contenue dans 4 pixels. Comme déjà dit au paragraphe 1.3.3, le modèle que nous avons choisi pour

les cibles n'est pas assez réaliste. Une signature qui s'étend sur plusieurs pixels serait plus réaliste. La valeur dans les pixels contenant la cible est plus élevée que dans les pixels voisins. La cible est donc bien additive et d'intensité positive.

Nous avons appliqué l'algorithme de détection au patch contenant la cible. Ce patch a été affecté à la classe 2. La figure 4.34 présente le patch à la sortie du filtre, avant le seuillage puis après le seuillage par $s=5.9193$ qui correspond à une probabilité de fausses alarmes souhaitée de 10^{-5} . La cible est détectée pour toutes les probabilités de fausses alarmes étudiées.

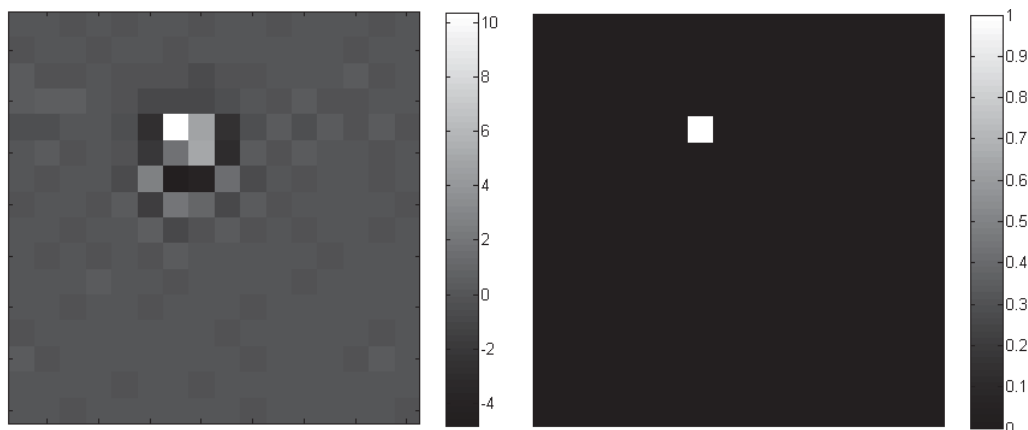


FIGURE 4.34 – Bruit après le filtre puis après le seuillage pour le patch contenant la cible

A la sortie du filtre, image gauche de la figure 4.34, nous pouvons remarquer que le nombre de pixels « occupés » par la cible est plus important qu'avant le filtrage. Cela est dû au filtre laplacien et au produit matriciel lors du retrait de la décorrélation qui ont déplacé une partie de la cible dans les voisins des quatre pixels contenant la cible initialement. Nous aurions pu baser notre test sur la détection de cette structure géométrique présente autour de la position de la cible après le filtrage. Au lieu de seuiller la statistique de test qui blanchit le patch, nous aurions seuillé une fonction de cette statistique de test. C'est d'ailleurs le principe du test du rapport de vraisemblance [SF94].

Performances

Nous avons utilisé notre algorithme de détection sur un échantillon de 70 cibles d'opportunité. Elles ont été labélisées « à la main » par le service du traitement d'images de TOSA. La majorité de ces cibles est présente dans une partie de l'image contenant du ciel bleu.

Nous avons imposé une probabilité de fausses alarmes valant 10^{-4} . Sur ces 70 cibles, 48 sont détectées par l'algorithme antérieur et 34 par notre algorithme TLP. Ces cibles étant présentes dans un fond uniforme, il est naturel (vu nos résultats empiriques) que l'algorithme antérieur les détecte mieux.

Le point positif de cette étude est que sur les 34 cibles détectées par TLP, 5 ne l'étaient pas par l'algorithme antérieur. Cela vient confirmer notre intention d'associer les deux algorithmes pour améliorer leurs performances sur tous les types de fonds.

Conclusion

Dans ce chapitre, nous avons défini un modèle gaussien pour les images de nuages. Nous avons supposé qu'une image I pouvait être segmentée en patches carrés de taille 16×16 avec des patches indépendants. Nous avons ensuite supposé que la matrice de covariance de chaque patch était structurée et parcimonieuse

A partir de ce modèle, nous avons proposé un filtre de blanchiment. La détection s'effectue par seuillage de la sortie du filtre. Notre objectif principal était de respecter le nombre de fausses alarmes par image. Le choix du seuil a donc été un point crucial de nos travaux.

Si le filtre de blanchiment était parfait, nous pourrions appliquer un quantile théorique. Toutefois, il existe un écart entre le modèle supposé et les images réelles. Nous avons donc estimé le seuil par le quantile empirique d'un échantillon d'apprentissage. Avec ces seuils, les fausses alarmes ne sont pas uniformément réparties dans l'image. Nous avons donc décidé de prendre en compte la texture du patch pour définir les seuils. Pour cela, nous avons utilisé des outils de classification non supervisée afin de classer les patches de l'échantillon d'apprentissage. Nous avons ensuite estimé les seuils indépendamment dans chacune des classes.

Finalement, la chaîne de détection TLP est composée de deux étapes :

Étape hors ligne : Construction des classes et apprentissage des seuils.

Étape en ligne : Application du filtre de blanchiment puis du seuil issu de l'étape hors ligne.

Nous avons ensuite comparé les performances de détection de notre algorithme à l'algorithme actuellement utilisé au sein de TOSA. Nous avons alors vu que leurs performances étaient complémentaires. Une perspective à court terme serait de les associer et d'appliquer l'un ou l'autre selon la texture de l'image traitée.

Conclusion

Durant ces travaux de thèse, nous avons rempli notre objectif principal qui était de proposer et d'implémenter une chaîne de détection de cibles sous-résolues sur fonds complexes. La construction d'une telle chaîne s'est faite en réponse à plusieurs questions.

Tout d'abord, quels modèles peut-on utiliser pour représenter le contenu des images réelles sur lesquelles nous avons travaillé ?

Nous avons étudié ce point dans le chapitre 2 en proposant des modèles gaussiens stationnaires (directs ou indirects). Nous en avons ensuite déduit des modèles gaussiens finis stationnaires par morceaux qui modélisaient mieux le contenu des fonds complexes. Les choix faits pour définir ces modèles l'ont été dans un objectif de prise en compte de la corrélation spatiale à faible portée du contenu des images. Nous disposions d'une grande base de données et nous avons pu très vite tester nos modèles sur des données réelles, ce qui nous a permis de les améliorer sans perdre de vue la recherche d'un modèle générique.

Nous avons également programmé plusieurs algorithmes d'estimation par maximum de vraisemblance des paramètres sous-jacents aux modèles.

Très naturellement, une nouvelle question est apparue : dans une collection de modèles \mathcal{M} construite avec peu d'a priori, comment choisir le modèle le plus adapté pour représenter le contenu d'une image particulière I ?

La théorie de la sélection de modèles introduite par Birgé et Massart nous a permis de répondre à cette question dans le chapitre 3. Nous avons sélectionné le modèle qui minimisait un critère pénalisé calculé directement à partir de l'image I . Nous avons obtenu une inégalité oracle pour cette procédure de sélection. Ce résultat théorique a permis de conforter certains choix faits lors de la construction de la chaîne de détection comme le découpage de l'image en patchs supposés indépendants par exemple. Nous avons implémenté cette procédure de sélection de modèles. Les temps de calcul étant trop longs, nous avons modifié le critère proposé par Massart. Même sans résultat théorique, ce nouveau critère a toutefois permis de réduire un peu le temps de calcul pour des résultats proches de ceux attendus. Les temps atteints ne permettaient malheureusement pas une utilisation dans la chaîne de détection.

Répondre aux questions précédentes nous a également permis de prendre conscience des difficultés qu'il existe lorsque l'on passe de la théorie à la pratique, lorsque les données ne veulent pas se comporter comme prévu !

Nous avons donc dû inverser notre approche et finalement, en nous basant sur les études menées sur notre base de données, nous avons choisi d'utiliser pour toutes les images un des modèles gaussiens finis stationnaires par morceaux proposés pour définir la statistique de test dans la chaîne de détection.

Concernant la chaîne en elle-même, les questions ont été plus pratiques. La plus cruciale a été : Quel seuil doit-on appliquer pour respecter le taux de fausses alarmes souhaité ?

En théorie, bien sûr, il suffit d'appliquer le quantile de la loi de la statistique de test pour le niveau souhaité. Encore une fois, il existe un écart entre le contenu des images et le modèle supposé. Pour tenir compte de cet écart, nous avons proposé une méthode d'apprentissage des seuils. Cette approche, présentée dans le chapitre 4, consiste à classer les patchs d'une base d'apprentissage selon leur texture. Pour cela, nous avons utilisé les descripteurs de scattering de Mallat ainsi que

l'algorithme CEM de classification non supervisée. Nous avons alors pu estimer des quantiles de différents niveaux dans chacune des classes. L'implémentation de cette méthode nous a de nouveau mis face à des difficultés pratiques comme la constitution de la base d'apprentissage par exemple.

Une fois la chaîne entière construite, une dernière question s'est posée : comment mesurer les performances d'un algorithme de détection ?

Nous nous sommes alors replongés dans la théorie de la détection à la fin du chapitre 1 pour pouvoir tracer les courbes COR présentées dans le chapitre 4. Certaines définitions ont eu besoin d'être adaptées pour des modèles prenant en compte la covariance spatiale du fond.

En conclusion, ces travaux de thèse nous auront amenés à réaliser plusieurs aller-retours entre la théorie et les données. L'accès à des images réelles a permis une meilleure compréhension des modèles même si sous certains aspects, elles les ont également mis en défaut.

La prise en compte de la corrélation spatiale du contenu des images nous semble indispensable pour réaliser une meilleure détection. Elle entraîne toutefois une augmentation non négligeable du temps nécessaire au traitement.

De même, l'apprentissage des seuils par classe a nettement amélioré les performances de détection sur les images réelles.

Perspectives

Chacune des réponses apportées aux questions précédentes a entraîné de nouvelles questions auxquelles nous n'avons pas répondu dans ces travaux. Les perspectives sont aussi bien d'aspect théorique que pratique.

Amélioration du temps de calcul

D'un point de vue pratique, toutes les implémentations ont été faites sous Matlab sans optimisation spécifique. Une implémentation sur architecture dédiée (FPGA, GPU) pourrait être envisagée.

Nous pourrions également étudier le gain en temps de calcul obtenu si l'on remplaçait l'estimation des paramètres par maximum de vraisemblance dans la chaîne de détection (paragraphe 4.1.2) par une estimation par maximum de pseudo vraisemblance (paragraphe 3.4).

Modélisation des fonds et des cibles

Modélisation du fond : Pour le fond, nous avons supposé la matrice de covariance Σ du champ \mathbf{B} diagonale par bloc avec la même structure parcimonieuse dans chaque bloc. Il pourrait être intéressant de ne pas imposer la même structure dans chacun des blocs mais de plutôt sélectionner parmi une collection la structure la plus adaptée dans chaque bloc. Verzelen [Ver09] a, par exemple, proposé une procédure pour sélectionner le voisinage du modèle CAR le plus adapté à un champ gaussien particulier \mathbf{Y} .

Modélisation des cibles : Nous avons présenté au paragraphe 1.3.4 différentes signatures pour des cibles additives subpixeliques. L'étude menée au chapitre 4 pour mesurer les performances de détection pourrait être relancée avec ces autres signatures pour la cible. De même, nous pourrions utiliser des signatures correspondant à des cibles dont la taille serait de plusieurs pixels (2-5). L'hypothèse d'additivité est une hypothèse forte et la totalité de la chaîne de détection est basée sur cette hypothèse. Un point intéressant serait de ne pas supposer les cibles additives mais de supposer que la cible vient remplacer le fond dans le pixel qui la contient. L'algorithme de détection devra alors être adapté à cette hypothèse.

Nous n'avons pas étudié le cas où une deuxième cible serait présente dans le patch. Cette cible va sûrement perturber l'estimation des paramètres. La robustesse de l'algorithme proposé est un axe de travail intéressant.

Choix des descripteurs

Nous avons décidé de travailler avec les coefficients de scattering de Mallat suite aux études empiriques de classification menées sur nos images. Une étude comparative sur les performances de la chaîne entière avec les descripteurs de Mallat (paragraphe 4.3.2) ou avec des descripteurs de Simoncelli/Portilla (4.3.1) permettrait de mieux justifier ce choix.

D'autre part, nous avons réduit la dimension de ces descripteurs de 417 à 10 par analyse en composantes principales. Nous aurions également pu sélectionner, dès la définition des descripteurs, les dix les plus adaptés à notre problème. Par exemple, quels sont les descripteurs qui sont les plus discriminants en présence d'une cible ? Quels sont ceux qui expliquent le mieux nos données ? Nous

pourrions, pour cela, appliquer la théorie de la sélection de modèles pour ne conserver que les variables pertinentes ([Mey12],[Dev15]).

Classification

Nous avons obtenu 8 classes en appliquant l'algorithme *CEM* sur les vecteurs contenant les descripteurs de Mallat pour notre base d'apprentissage. Nous avons vu que certaines classes étaient assez bien caractérisées : la classe 5 correspond aux bords de nuage par exemple. Un travail intéressant consisterait à caractériser chacune des 8 classes obtenues.

Il semble également que certaines pourraient être regroupées. La pénalité appliquée est peut être encore trop faible. Comme pour le critère pénalisé du chapitre 3, celui du chapitre 4 pour la classification mériterait une analyse plus approfondie.

Détection

Détection spatio-temporelle : Notre chaîne de détection ne prend pas en compte l'aspect temporel du problème. Par exemple, nous pourrions effectuer notre détection sur plusieurs images de la même séquence puis utiliser l'aspect temporel pour faciliter la détection. Nous utiliserions alors des techniques de *Track Before Detect* (TBD) ou de *Track After Detect* (TAD).

Performances : Nous avons comparé notre algorithme à l'état de l'art chez TOSA puisque notre objectif principal était d'améliorer ses performances sur fonds complexes. Il serait toutefois très intéressant de comparer notre algorithme à d'autres algorithmes prenant en compte la corrélation spatiale comme celui proposé par Genin lors de sa thèse par exemple [Gen13].

Association d'algorithmes

Nous avons vu dans le chapitre 4 que les deux algorithmes comparés avaient des performances complémentaires. L'extension la plus naturelle consiste à associer ces deux algorithmes. En pratique, l'algorithme appliqué dépendrait de la classe à laquelle appartient le patch traité. Cette association peut se faire très facilement puisque nous disposons déjà du critère de choix entre les deux algorithmes grâce au classifieur que nous avons fabriqué au chapitre 4.

Le schéma suivant présente la chaîne de détection prenant en compte cette association d'algorithmes. Nous pourrions bien sûr le généraliser à plus de deux algorithmes, idéalement un par classe.

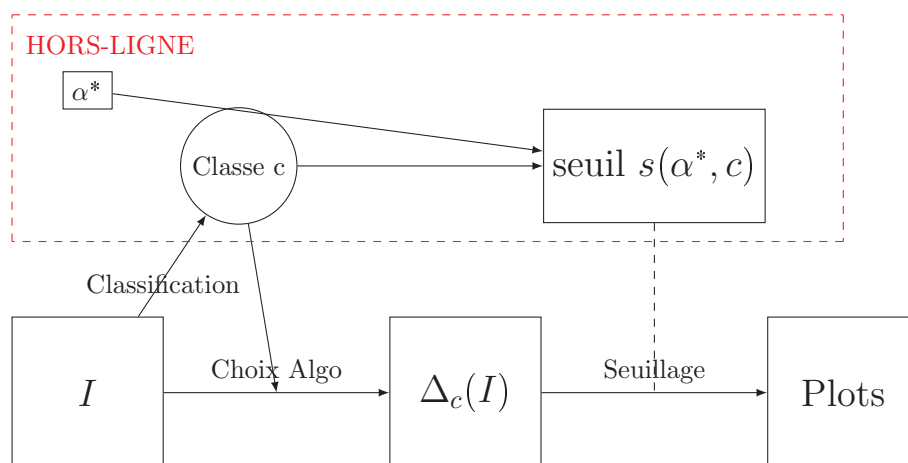


Schéma de la chaîne intégrant un choix pour l'algorithme utilisé

Annexe A

Idée de preuve pour l'algorithme d'estimation sous contraintes

Pour un échantillon $\mathbf{X}_1, \dots, \mathbf{X}_n$, nous notons la matrice de covariance empirique $S_e = \frac{1}{n} \sum_{i=1}^n {}^t \mathbf{X}_i \mathbf{X}_i$. La log-vraisemblance s'exprime à partir de cette matrice S_e

$$g(S, \Sigma) = -\log(\det(\Sigma)) - \text{tr}(\Sigma^{-1} S_e),$$

sous la contrainte que Σ soit de la forme $\sum_{i=1}^d c_i T_i$ (structure \mathcal{R}).

Nous allons d'abord calculer la différentielle de la fonction g en tout point de la structure.

A.1 Calcul de la différentielle

Pour $R \in \text{GL}_n(\mathbb{R})$, $D_R \det(H) = \text{tr}[{}^t \text{com}(R).H]$.

Or, pour une matrice A quelconque, $A.{}^t \text{com}(A) = \det(A).I_n$ donc $D_R \det(H) = \det(R) \text{tr}[R^{-1}.H]$.

Par la formule des différentielles composées,

$$D_R(\log(\det))(H) = D_{\det(R)} \log(D_R \det(H)) = \frac{\det(R) \text{tr}[R^{-1}.H]}{\det(R)},$$

donc $D_R(\log(\det))(H) = \text{tr}[R^{-1}.H]$.

De plus, pour $R \in \text{GL}_n(\mathbb{R})$, $D_R \text{inv}(H) = -R^{-1} H R^{-1}$.

La fonction trace étant linéaire, nous obtenons

$$\begin{aligned} D_R g(H) &= -\text{tr}(R^{-1} H) + \text{tr}(R^{-1} H R^{-1} S_e) \\ &= \text{tr}[(R^{-1} S_e R^{-1} - R^{-1}) H]. \end{aligned}$$

A.2 Recherche du maximum de vraisemblance

Nous ne pouvons cependant pas résoudre directement $D_R g = 0$ et nous allons le faire de façon itérative. Supposons que l'on connaisse une matrice R proche de la matrice cherchée. Nous avons donc

$$\forall h \in \mathcal{R}, \text{tr}[(R^{-1} S_e R^{-1} - R^{-1}) H] = o(1).$$

Nous essayons de trouver une nouvelle solution sous la forme $R' = R + qD$ où q et D seront à préciser. Nous avons alors

$$\begin{aligned} R' &= R + qD = R(Id + qR^{-1}D) \\ R'^{-1} &= (Id + qR^{-1}D)^{-1} R^{-1} \\ &= (Id - qR^{-1}D + o(q)) R^{-1} \\ &= R^{-1} - qR^{-1} D R^{-1} + o(q) \end{aligned}$$

Nous souhaitons avoir $\text{tr}[(R'^{-1}S_eR'^{-1} - R'^{-1})H] = o(1)$. Calculons le membre de gauche.

$$\begin{aligned}
 R'^{-1}S_eR'^{-1} - R'^{-1} &= (R^{-1} - qR^{-1}DR^{-1} + o(q))S_e(R^{-1} - qR^{-1}DR^{-1} + o(q)) \\
 &\quad - R^{-1} - qR^{-1}DR^{-1} + o(q) \\
 &= R^{-1}S_eR^{-1} - qR^{-1}S_eR^{-1}DR^{-1} - qR^{-1}DR^{-1}S_eR^{-1} \\
 &\quad - R^{-1} + qR^{-1}DR^{-1} + o(q) \\
 &= R^{-1}S_eR^{-1} - R^{-1} - q \underbrace{R^{-1}S_eR^{-1}}_{R^{-1}+o(1)} DR^{-1} + qR^{-1}D \underbrace{R^{-1}S_eR^{-1}}_{R^{-1}+o(1)} \\
 &\quad - qR^{-1}DR^{-1} + o(q) \\
 &= R^{-1}S_eR^{-1} - R^{-1} - q[R^{-1}DR^{-1} + R^{-1}DR^{-1} - R^{-1}DR^{-1}] + o(q) \\
 &= R^{-1}S_eR^{-1} - R^{-1} - qR^{-1}DR^{-1} + o(q) \\
 \text{tr}[(R'^{-1}S_eR'^{-1} - R'^{-1})H] &= \text{tr}[(R^{-1}S_eR^{-1} - R^{-1})H] - q\text{tr}[(R^{-1}DR^{-1})H] + o(q) \\
 &= \text{tr}[(R^{-1}S_eR^{-1} - R^{-1} - q(R^{-1}DR^{-1}))H] + o(q) \\
 &= \text{tr}[(R^{-1}S_eR^{-1} - R^{-1}RR^{-1} - q(R^{-1}DR^{-1}))H] + o(q) \\
 &= \text{tr}[(R^{-1}S_eR^{-1} - R^{-1}(R + qD)R^{-1})H] + o(q) \\
 &= \text{tr}[(R^{-1}S_eR^{-1} - R^{-1}R'R^{-1})H] + o(q)
 \end{aligned}$$

Finalement, nous pouvons fabriquer une nouvelle matrice proche de la matrice cherchée en résolvant

$$\boxed{\forall H \in \mathcal{R} \quad \text{tr}[(R^{-1}S_eR^{-1} - R^{-1}R'R^{-1})H] = 0.}$$

Annexe B

Démonstration du théorème 2

Soit $\mathcal{M}(\mathcal{P}, \mathcal{S})$ une collection de modèles pour lesquels l'hypothèse $(H_{\mathcal{P}})$ et les hypothèses (G_m) sont vérifiées. Pour obtenir la conclusion de théorème 2, nous souhaitons appliquer le théorème de Massart énoncé au début du chapitre 3. Nous devons donc vérifier que nous sommes bien dans les conditions de ce théorème.

Nous devons donc montrer que le jeu d'hypothèses $(H_{\mathcal{P}})$ et (G_m) pour $m \in \mathcal{M}(\mathcal{P}, \mathcal{S})$ entraîne le jeu d'hypothèses $H_{\mathcal{M}(\mathcal{P}, \mathcal{S})}$ et (H_m) pour $m \in \mathcal{M}(\mathcal{P}, \mathcal{S})$.

B.1 Hypothèse $(H_{\mathcal{M}(\mathcal{P}, \mathcal{S})})$

Pour un modèle $m = (R_1, \dots, R_P, S_{i_1}, \dots, S_{i_P})$, posons $\rho_m = \rho_R + |R| \log(|\mathcal{S}|)$ où la famille $(\rho_R, R \in \mathcal{P})$ est celle définie par l'hypothèse $(H_{\mathcal{P}})$. Alors,

$$\sum_{m \in \mathcal{M}(\mathcal{P}, \mathcal{S})} e^{-\rho_m} = \sum_{m \in \mathcal{M}(\mathcal{P}, \mathcal{S})} e^{-\rho_R - |R| \log(|\mathcal{S}|)} = \sum_{R \in \mathcal{P}} e^{-\rho_R} \sum_{i_1, \dots, i_P \in \mathcal{S}^{|R|}} e^{-|R| \log(|\mathcal{S}|)} = \sum_{R \in \mathcal{P}} e^{-\rho_R}$$

L'hypothèse $H_{\mathcal{M}(\mathcal{P}, \mathcal{S})}$ est donc bien vérifiée.

Pour démontrer que l'hypothèse (H_m) est elle aussi vérifiée pour tout modèle $m \in \mathcal{M}(\mathcal{P}, \mathcal{S})$, nous allons avoir besoin de la proposition 2 (pour montrer que l'hypothèse (G_m) entraîne l'hypothèse (H'_m)) puis du lemme 1 (pour montrer que l'hypothèse (H'_m) entraîne l'hypothèse (H_m)).

B.2 Démonstration de la proposition 2

Pour majorer l'entropie à crochets $H_{\square}(\varepsilon, m)$, nous allons chercher à majorer la quantité $N_{\square}(\varepsilon, m)$. Le modèle m est un modèle paramétrique donc il est de la forme $m = \{f_{\theta} = N_p(0, \Sigma(\theta)), \theta \in \Theta\}$.

L'ensemble Θ est un ensemble de dimension d que l'on suppose borné : $\Theta \subset \prod_{i=1}^d [m_i, M_i]$.

Nous cherchons à construire un crochet $[t_{\theta}^-, t_{\theta}^+]$ de taille ε pour chaque fonction f_{θ} du modèle m . Très souvent dans la littérature, ces fonctions sont cherchées sous la forme :

$$\begin{aligned} t_{\theta}^- (x) &= (1 + \delta)^{-p} \Phi_p(x|0, (1 + \delta)^{-1} \Sigma(\hat{\theta})) \\ t_{\theta}^+ (x) &= (1 + \delta)^p \Phi_p(x|0, (1 + \delta) \Sigma(\hat{\theta})) \end{aligned}$$

où $\Phi_p(\cdot|0, \Sigma)$ est la densité d'un modèle gaussien $N(0, \Sigma)$.

δ et $\hat{\theta}$ sont des paramètres à choisir pour garantir que $h^2(t_{\theta}^-, t_{\theta}^+) \leq \varepsilon^2$ et que $t_{\theta}^- \leq f_{\theta} \leq t_{\theta}^+$.

B.2.1 $[t_{\theta}^-, t_{\theta}^+]$ crochet de taille ε

Nous allons calculer la taille du crochet $[t_{\theta}^-, t_{\theta}^+]$.

$$\begin{aligned}
h^2(t_\theta^-, t_\theta^+) &= (1 + \delta)^p + (1 + \delta)^{-p} - 2 + h^2(\Phi_p(\cdot|0, (1 + \delta)\Sigma(\hat{\theta})), \Phi_p(\cdot|0, (1 + \delta)^{-1}\Sigma(\hat{\theta}))) \\
&= (1 + \delta)^p + (1 + \delta)^{-p} - 2 \times 2^{\frac{p}{2}} \left((1 + \delta) + (1 + \delta)^{-1} \right)^{-\frac{p}{2}} \\
&= 2\cosh(p \log(1 + \delta)) - 2 + 2 - 2\cosh(\log(1 + \delta))^{-\frac{p}{2}}.
\end{aligned}$$

où la deuxième égalité est obtenue grâce à la formule suivante

$$h^2(\Phi_p(\cdot|0, (1 + \delta)\Sigma), \Phi_p(\cdot|0, (1 + \delta)^{-1}\Sigma)) = 2 - 2 \times 2^{\frac{p}{2}} \left((1 + \delta) + (1 + \delta)^{-1} \right)^{-\frac{p}{2}}.$$

Posons $f(x) = 1 - \cosh(x)^{-\frac{p}{2}}$. Par une étude de la fonction f , on obtient

$$f''(x) = \frac{p}{2} \cosh(x)^{-\frac{p}{2}} \left(1 - \frac{p}{2} \left(\frac{p}{2} + 1 \right) \frac{\sinh(x)^2}{\cosh(x)} \right) \leq \frac{p}{2}.$$

Puisque $f'(0) = 0$ et $f(0) = 0$, on en déduit, par intégration successive, que $f(x) \leq \frac{p}{2} \frac{x^2}{2} \leq \frac{p^2}{2} \frac{x^2}{2}$. Finalement $2 - 2\cosh(\log(1 + \delta))^{-\frac{p}{2}} \leq \frac{p^2}{2} \log(1 + \delta)^2$ et avec $\log(1 + \delta) \leq \delta$, on obtient

$$2 - 2\cosh(\log(1 + \delta))^{-\frac{p}{2}} \leq \frac{p^2 \delta^2}{2}$$

Posons de même $g(x) = \cosh(x) - 1$. Comme $g''(x) = \cosh(x) \leq \cosh(c)$ pour $x \in [0, c]$, par intégration successive, nous obtenons

$$\forall \delta \text{ tel que } p \log(1 + \delta) \leq c, \quad 2\cosh(p \log(1 + \delta)) - 2 \leq \cosh(c) p^2 \delta^2.$$

Revenons alors à la distance d'Hellinger entre les deux fonction t_θ^- et t_θ^+ .

$$h^2(t_\theta^-, t_\theta^+) \leq \cosh(c) p^2 \delta^2 + \frac{p^2 \delta^2}{2} \leq \frac{p^2 \delta^2}{2} \left(1 + 2\cosh(c) \right) \leq 4p^2 \delta^2.$$

dès que $|c| \leq \frac{3}{2}$. Ainsi, quelque soit la valeur de $\hat{\theta}$ si on pose $\delta = \frac{\varepsilon}{2p}$ nous obtenons alors $h^2(t_\theta^-, t_\theta^+) \leq \varepsilon^2$.

Remarque. Nous avons bien $p \log(1 + \frac{\varepsilon}{2p}) \leq \frac{3}{2}$ d'où cette valeur pour c .

B.2.2 $[t_\theta^-, t_\theta^+]$ contient f_θ .

Cette partie est plus compliquée que la précédente. Nous devons montrer qu'il existe un choix judicieux de $\hat{\theta}$ garantissant que $t_\theta^- \leq f_\theta \leq t_\theta^+$. Nous allons commencer par calculer les deux fractions $\frac{t_\theta^-}{f_\theta}$ et $\frac{f_\theta}{t_\theta^+}$.

$$\begin{aligned}
\frac{f_\theta}{t_\theta^+} &= \frac{\Phi_p(x|0, \Sigma(\theta))}{(1 + \delta)^p \Phi_p(x|0, (1 + \delta)\Sigma(\hat{\theta}))} = \frac{1}{(1 + \delta)^p} \frac{\Phi_p(x|0, \Sigma(\theta))}{\Phi_p(x|0, (1 + \delta)\Sigma(\hat{\theta}))} \\
&\leq \frac{1}{(1 + \delta)^p} \sqrt{\frac{|(1 + \delta)\Sigma(\hat{\theta})|}{|\Sigma(\theta)|}} \\
&\leq (1 + \delta)^{-\frac{p}{2}} \sqrt{\frac{|\Sigma(\hat{\theta})|}{|\Sigma(\theta)|}}
\end{aligned}$$

où la première inégalité est obtenue en appliquant la proposition 6 de Maugis et Michel [MM11]. Cette proposition nécessite que la matrice $(1 + \delta)\Sigma(\hat{\theta}) - \Sigma(\theta)$ soit définie positive.

De façon similaire, nous pouvons montrer que

$$\frac{t_{\hat{\theta}}^-}{f_{\theta}} \leq (1 + \delta)^{-\frac{p}{2}} \sqrt{\frac{|\Sigma(\theta)|}{|\Sigma(\hat{\theta})|}}.$$

où cette fois c'est la matrice $\Sigma(\theta) - (1 + \delta)^{-1}\Sigma(\hat{\theta})$ qui doit être définie positive.

Pour tout paramètre $\theta \in \Theta$, nous devons donc construire un paramètre $\hat{\theta}$ tel que

- la matrice $(1 + \delta)\Sigma(\hat{\theta}) - \Sigma(\theta)$ est définie positive ;
- la matrice $\Sigma(\theta) - (1 + \delta)^{-1}\Sigma(\hat{\theta})$ est définie positive et
- la majoration suivante est vérifiée :

$$(1 + \delta)^{-\frac{p}{2}} \leq \sqrt{\frac{|\Sigma(\hat{\theta})|}{|\Sigma(\theta)|}} \leq (1 + \delta)^{\frac{p}{2}}$$

Nous allons traduire ces trois conditions directement sur la distance entre θ et $\hat{\theta}$.

Nous nous intéressons tout d'abord à la dernière condition que l'on peut réécrire

$$\left| \log \left(\frac{|\Sigma(\hat{\theta})|}{|\Sigma(\theta)|} \right) \right| \leq p \log(1 + \delta)$$

Or, la fonction $\theta \mapsto \log(|\Sigma(\theta)|)$ est k - p -lipschitz et on peut donc remplacer la dernière condition par

$$\|\theta - \hat{\theta}\| \leq \frac{1}{k} \log(1 + \delta).$$

C'est la deuxième hypothèse faite sur le modèle m qui va nous permettre de traduire les deux premières conditions. Soit x un vecteur de \mathbb{R}^p .

$${}^t x [(1 + \delta)\Sigma(\hat{\theta}) - \Sigma(\theta)]x = {}^t x (1 + \delta)[\Sigma(\hat{\theta}) - \Sigma(\theta)]x + {}^t x \delta \Sigma(\theta)x.$$

Nous allons minimiser les deux termes séparément.

$$\begin{aligned} {}^t x [\Sigma(\hat{\theta}) - \Sigma(\theta)]x &> -\|\Sigma(\hat{\theta}) - \Sigma(\theta)\| \cdot \|x\|^2 \\ {}^t x \Sigma(\theta)x &> \lambda_1(\theta) \cdot \|x\|^2 \end{aligned}$$

où $\lambda_1(\theta)$ est la plus petite valeur propre de $\Sigma(\theta)$. Alors,

$${}^t x (1 + \delta)[\Sigma(\hat{\theta}) - \Sigma(\theta)]x + {}^t x \delta \Sigma(\theta)x > [-(1 + \delta)\|\Sigma(\hat{\theta}) - \Sigma(\theta)\| + \delta \lambda_1(\theta)] \cdot \|x\|^2.$$

Ainsi, si $\hat{\theta}$ est tel que

$$\|\Sigma(\hat{\theta}) - \Sigma(\theta)\| \leq \frac{\delta \lambda_1(\theta)}{1 + \delta},$$

la matrice $(1 + \delta)\Sigma(\hat{\theta}) - \Sigma(\theta)$ sera bien définie positive. Or, la fonction $\theta \mapsto \Sigma(\theta)$ est k -lipschitz. Il est donc suffisant que $\hat{\theta}$ vérifie la condition suivante

$$\|\theta - \hat{\theta}\| \leq \frac{\delta \lambda_1(\theta)}{k(1 + \delta)}.$$

On raisonne de même pour la matrice $\Sigma(\theta) - (1 + \delta)^{-1}\Sigma(\hat{\theta})$.

$${}^t x[\Sigma(\theta) - (1 + \delta)^{-1}\Sigma(\hat{\theta})]x \geq \frac{\delta\lambda_1(\theta) - \|\Sigma(\hat{\theta}) - \Sigma(\theta)\|}{1 + \delta} \|x\|^2$$

La deuxième condition peut elle être remplacée par

$$\|\theta - \hat{\theta}\| \leq \frac{\delta\lambda_1(\theta)}{k}.$$

Construction de la grille pour $\hat{\theta}$

Remarquons tout d'abord que $\forall \delta \leq \sqrt{2}$, $\frac{\delta\lambda_1(\theta)}{1+\delta} \geq \frac{\lambda_1(\theta)}{3}\delta$ et $\log(1 + \delta) \geq \frac{\delta}{3}$.

Les trois conditions définissant $\hat{\theta}$ peuvent être remplacées par la condition suivante

$$\|\theta - \hat{\theta}\| \leq \frac{1}{6pk} \min(1, \lambda)\varepsilon.$$

où $\lambda = \min_{\theta} \lambda_1(\theta)$.

Nous souhaitons donc construire une grille $\hat{\theta}_1, \dots, \hat{\theta}_{N(\varepsilon)}$ telle que

$$\forall \theta \in \Theta, \exists i \in [1, N(\varepsilon)] \text{ tel que } \|\theta - \hat{\theta}_i\| \leq \frac{1}{6pk} \min(1, \lambda)\varepsilon.$$

Cette grille permettra de fabriquer des crochets $[t_{\hat{\theta}_i}^-, t_{\hat{\theta}_i}^+]$ de taille ε qui recouvriront tout le modèle m .

Comme nous souhaitons majorer $N_{\square}(\varepsilon, m)$, ce n'est pas la grille en elle-même qui nous intéresse mais plutôt sa taille $N(\varepsilon)$ puisqu'en effet $N_{\square}(\varepsilon, m) \leq N(\varepsilon)$. Pour avoir la majoration la plus fine possible de $N_{\square}(\varepsilon, m)$, nous voulons que cette grille soit la plus grossière possible, c'est-à-dire avec le moins de points possibles.

La famille $(\hat{\theta}_1, \dots, \hat{\theta}_{N(\varepsilon)})$ est en fait un recouvrement de taille $\frac{1}{6pk} \min(1, \lambda)\varepsilon$ de l'ensemble Θ . Le résultat suivant va permettre de majorer la taille de la grille $N(\varepsilon)$.

Lemme 2 (Recouvrement et paquets de taille δ) Soit $\delta \leq \sqrt{2}$. On définit $N_{cov}(\delta, \Theta)$, le nombre de recouvrement de taille δ et $N_{pack}(\delta, \Theta)$, le nombre de paquets de taille δ par

$$\begin{aligned} N_{cov}(\delta, \Theta) &= \min \{n, \exists (x_1, \dots, x_n) \in \Theta, \forall x \in \Theta, \exists i \in [1, n], \|x - x_i\| \leq \delta\} \\ N_{pack}(\delta, \Theta) &= \max \{n, \exists (y_1, \dots, y_n) \in \Theta, \forall i, j \in [1, n], \|y_i - y_j\| \geq \delta\} \end{aligned}$$

Alors,

$$N_{cov}(\delta, \Theta) \leq N_{pack}(\delta, \Theta).$$

Démontrons ce résultat. Soit $(y_1, \dots, y_{N_{pack}(\delta, \Theta)})$ un paquet de Θ de taille maximale. Puisqu'il est de taille maximale, aucun point ne peut être ajouté à cet ensemble afin d'y former un nouveau paquet. Donc, pour tous les points x de Θ , il existe un point y_i tel que $\|x - y_i\| \leq \delta$. L'ensemble $(y_1, \dots, y_{N_{pack}(\delta, \Theta)})$ est donc un recouvrement de Θ . D'où le résultat.

Ainsi, en notant $c = \frac{1}{6k} \min(1, \lambda)$, nous obtenons $N_{\square}(\varepsilon, m) \leq N_{cov}\left(\frac{c\varepsilon}{p}\right) \leq N_{pack}\left(\frac{c\varepsilon}{p}\right)$.

Majoration de $N_{\text{pack}}\left(\frac{c\varepsilon}{p}\right)$

On note $y_1, \dots, y_{N_{\text{pack}}}$ les points de Θ apparaissant dans la définition de $N_{\text{pack}}\left(\frac{c\varepsilon}{p}\right)$. La réunion de toutes les boules de centre y_i et de rayon $\frac{c\varepsilon}{2p}$, notée $B\left(y_i, \frac{c\varepsilon}{2p}\right)$, est contenue dans l'ensemble $\Theta + \frac{c\varepsilon}{2p}$. De plus, ces boules sont disjointes. On a donc

$$\begin{aligned} \text{Vol}\left(\bigcup_{i=1}^{N_{\text{pack}}} B\left(y_i, \frac{c\varepsilon}{2p}\right)\right) &\leq \text{Vol}\left(\Theta + \frac{c\varepsilon}{2p}\right) \\ N_{\text{pack}}\left(\frac{c\varepsilon}{p}\right) \times \left(\frac{c\varepsilon}{2p}\right)^d &\leq \text{Vol}\left(\Theta + \frac{c\varepsilon}{2p}\right) \\ &\leq \prod_{i=1}^d [M_i - m_i + c\varepsilon] \\ &\leq [\max_i (M_i - m_i) + c\sqrt{2}]^d \\ N_{\text{pack}}\left(\frac{c\varepsilon}{p}\right) &\leq C(\Theta)^{\dim(m)} \times \left(\frac{2p}{c\varepsilon}\right)^{\dim(m)} \end{aligned}$$

On conclut alors que

$$H_{\square}(\varepsilon, m) = \log(N_{\square}(\varepsilon, m)) \leq \log\left(N_{\text{pack}}\left(\frac{c\varepsilon}{p}\right)\right) \leq \dim(m) \left(C + \log\left(\frac{p}{\varepsilon}\right)\right).$$

B.2.3 Démonstration Lemme 1

Soit m un modèle de la collection $\mathcal{M}(\mathcal{P}, \mathcal{S})$. L'hypothèse (H_m) porte sur l'entropie à crochet $H_{\square}(\varepsilon, m)$. Nous devons majorer cette entropie. Plus particulièrement, nous allons majorer $N_{\square}(\varepsilon, m)$.

Soit f une densité du modèle m . Nous souhaitons construire un crochet de taille ε contenant f .

Comme le modèle est un modèle par morceaux, nous savons que f est de la forme $f(x) = \prod_{j=1}^P f_j(x_{|j})$

où chaque $f_{|j}$ est une densité du modèle S_{i_j} . Nous allons construire un crochet de taille ε à partir de crochet $[t_j^-, t_j^+]$ de taille η_j pour chaque $f_{|j}$.

Posons alors

$$t^-(x) = \prod_{j=1}^{|R|} t_j^-(x_{|j}) \text{ et } t^+(x) = \prod_{j=1}^{|R|} t_j^+(x_{|j}).$$

Nous allons montrer que, pour un choix particulier des coefficients η_j , le couple de fonctions (t^-, t^+) est bien un crochet de taille ε et qu'il contient f . Tout d'abord, puisque chaque couple $[t_i^-, t_i^+]$ est un crochet pour la fonction $f_{|i}$,

$$\begin{aligned} \forall j \in [1, P], t_i^-(x_i) &\leq f_j(x_{|j}) \leq t_i^+(x_i) \\ \text{donc } t^-(x) &\leq f(x) \leq t^+(x). \end{aligned}$$

et $[t^-, t^+]$ est donc bien un crochet contenant f .

Calculons maintenant sa taille, c'est à dire $h^2(t^-, t^+)$. Pour cela, nous allons utiliser l'affinité d'Hellinger définie pour deux fonction f et g par $A(f, g) = \int_{\mathcal{R}} \sqrt{f(x)g(x)} dx$. On a donc $h^2(f, g) = 2(1 - A(f, g))$. Alors,

$$h^2(t^-, t^+) = 2[1 - A(t^-, t^+)] = 2\left[1 - \prod_{i=1}^{|R|} A(t_i^-, t_i^+)\right] = 2\left[1 - \prod_{i=1}^{|R|} \left(1 - \frac{1}{2}h^2(t_i^-, t_i^+)\right)\right]$$

En utilisant l'inégalité suivante $\prod_{i=1}^n (1 - x_i) \geq 1 - \sum_{i=1}^n x_i$ alors

$$h^2(t^-, t^+) \leq 2 \left[1 - 1 + \frac{1}{2} \sum_{i=1}^{|R|} h^2(t_i^-, t_i^+) \right] \leq \sum_{i=1}^{|R|} h^2(t_i^-, t_i^+) \leq \sum_{i=1}^{|R|} \eta_i^2.$$

Ainsi, si on choisit $\eta_i^2 = \frac{\varepsilon^2 \#(R_i)^2}{\sum_{i=1}^{|R|} \#(R_i)^2}$, le crochet $[t^-, t^+]$ est bien de taille ε et contient f .

Cette construction de crochets pour f , à partir de crochets pour chaque fonction $f|_j$ permet aussi de majorer le nombre de crochets de taille ε du modèle m .

$$N_{\square}(\varepsilon, m) \leq \prod_{j=1}^{|R|} N_{\square} \left(\frac{\varepsilon \#(R_j)}{\sqrt{\sum_{i=1}^{|R|} \#(R_i)^2}}, S_{i_j} \right)$$

et donc $H_{\square}(\varepsilon, m) \leq \sum_{j=1}^{|R|} H_{\square} \left(\frac{\varepsilon \#(R_j)}{\sqrt{\sum_{i=1}^{|R|} \#(R_i)^2}}, S_{i_j} \right).$

On applique l'hypothèse du lemme 1 pour conclure que

$$\begin{aligned} H_{\square}(\varepsilon, m) &\leq \sum_{i=1}^{|R|} \dim(S_{i_j}) \left(C + \log \left(\frac{\sqrt{\sum_{i=1}^{|R|} \#(R_i)^2}}{\varepsilon} \right) \right) \\ &\leq \dim(m) \left(C + \log \left(\frac{M^2}{\varepsilon} \right) \right) \end{aligned}$$

où la dernière inégalité est obtenue car $\sum_{i=1}^{|R|} \#(R_i)^2 \leq M^2$. On est alors dans les conditions du lemme 2.

Nous pouvons maintenant appliquer un résultat démontré par Le Pennec et Cohen dans [LPC13].

Lemme 3 *Soit m un modèle tel que*

$$\forall \varepsilon > 0, H_{\square}(\varepsilon, m) \leq \dim(m) \left(D + \log \frac{1}{\varepsilon} \right)$$

alors la fonction ϕ_m définie par

$$\phi_m(\sigma) = \sigma \sqrt{\dim(m)} \left(\sqrt{D} + \sqrt{\pi} + \sqrt{\log \frac{1}{\sigma \wedge e^{-1/2}}} \right)$$

vérifie l'hypothèse (H_m) .

En appliquant ce lemme avec $D = C + \log(M^2)$, nous obtenons la conclusion du lemme 1.

Conclusion

Nous avons démontré que les hypothèses $(H_{\mathcal{M}})$ et toutes les (H_m) étaient donc vérifiées pour notre collection $\mathcal{M}(\mathcal{P}, \mathcal{S})$. Nous pouvons donc appliquer le théorème 1. Pour trouver la minorité de la pénalité annoncée, nous avons majoré la quantité $\sigma_m^2 + \frac{\rho_m}{n}$.

— Le réel σ_m est solution de $\phi_m(\sigma_m) = \sqrt{n} \sigma_m$ c'est à dire

$$\sqrt{\frac{\dim(m)}{n}} \left(\sqrt{C + \log(M^2)} + \sqrt{\pi} + \sqrt{\log \frac{1}{\sigma_m \wedge e^{-1/2}}} \right) = \sigma_m.$$

Si on pose $\sigma^* = \sqrt{\frac{\dim(m)}{n}} \left(\sqrt{C + \log(M^2)} + \sqrt{\pi} \right)$ on a $\sigma^* \leq \sigma_m$.

Or, la fonction $x \mapsto \frac{\phi_m(x)}{x}$ est décroissante et donc

$$\begin{aligned} \frac{\phi_m(\sigma_m)}{\sigma_m} &\leq \frac{\phi_m(\sigma^*)}{\sigma^*} \\ \sigma_m &\leq \sqrt{\frac{\dim(m)}{n}} \left(\sqrt{C + \log(M^2)} + \sqrt{\pi} + \sqrt{\log \frac{1}{\sigma^* \wedge e^{-1/2}}} \right) \end{aligned}$$

— pour un modèle $m = (R_1, \dots, R_P, S_{i_1}, \dots, S_{i_P})$, $\rho_m = \rho_R + |R| \log(|S|) \sim |R|$ puisque ρ_R est proportionnelle à la dimension de R pour les deux collections de partitions utilisées. Ainsi, $\rho_m \leq \text{Cte} \cdot \dim(m)$.

Nous pouvons donc majorer $\sigma_m^2 + \frac{\rho_m}{n}$ par

$$K \frac{\dim(m)}{n} \left[\sqrt{C + \log(M^2)} + \sqrt{\pi} + \sqrt{\log \frac{1}{\sqrt{\frac{\dim(m)}{n}} \left(\sqrt{C + \log(M^2)} + \sqrt{\pi} \right) \wedge e^{-1/2}}} \right]^2.$$

Annexe C

Autres démonstrations du chapitre 3

C.1 Démonstration de la proposition 4

Notons $q = \#(P)$, le nombre de pixels présents dans la zone P . Nous notons W au lieu de W_P pour désigner la matrice des quatre plus proches voisins adaptée à la taille de P .

Hypothèse 1. Soit $f_1 : (a, \sigma) \mapsto \log(\det(\Sigma(a, \sigma)))$. Pour le calcul du déterminant, nous allons utiliser les valeurs propres de la matrice Σ qui s'écrivent à partir de celles de la matrice W . Notons e_1, \dots, e_q les valeurs propres de W et $\lambda_1, \dots, \lambda_q$ celles de Σ . Alors,

$$\forall i \in \llbracket 1, q \rrbracket \quad \lambda_i = 1 - ae_i,$$
$$\det(\Sigma(a, \sigma)) = \sigma^{2q} \prod_{i=1}^q (1 - ae_i)^{-2}.$$

On a donc une nouvelle expression pour $f_1 : f_1(a, \sigma) = q \log(\sigma^2) - 2 \sum_{i=1}^q \log(1 - ae_i)$.

Soient $(a, a') \in [0, \frac{1}{4} - \eta]$ et $(\sigma^2, \sigma'^2) \in [v, V]$.

$$\begin{aligned} |f_1(a, \sigma) - f_1(a', \sigma')| &= \left| q \log\left(\frac{\sigma^2}{\sigma'^2}\right) - 2 \sum_{i=1}^q \log\left(\frac{1 - ae_i}{1 - a'e_i}\right) \right| \\ &\leq \left| q \log\left(\frac{\sigma^2}{\sigma'^2}\right) \right| + 2 \sum_{i=1}^q \left| \log\left(\frac{1 - ae_i}{1 - a'e_i}\right) \right| \\ &\leq \frac{q}{v} |\sigma - \sigma'| + \frac{2q}{\eta} |a - a'| \end{aligned}$$

où on a utilisé pour la dernière inégalité que la fonction $x \mapsto \log(x)$ est $\frac{1}{v}$ -Lipschitz sur l'intervalle $[v, V]$ et que pour toute valeur de e , la fonction $x \mapsto \log(1 - xe)$ est elle $\frac{1}{\eta}$ -Lispchitz sur l'intervalle $[0, \frac{1}{4} - \eta]$.

Finalement, en utilisant la norme infinie $\|(a, \sigma) - (a', \sigma')\|_\infty = \max(|\sigma - \sigma'|, |a - a'|)$, on obtient

$$|f_1(a, \sigma) - f_1(a', \sigma')| \leq q \left(\frac{1}{v} + \frac{1}{\eta} \right) \|(a, \sigma) - (a', \sigma')\|_\infty$$

et la fonction $(a, \sigma) \mapsto \log(\det(\Sigma(a, \sigma)))$ est bien qk -lispchitz.

Hypothèse 2. Soit, maintenant, $f_2 : (a, \sigma) \mapsto \Sigma(a, \sigma)$.
Soient $(a, a') \in [0, \frac{1}{4} - \eta]$ and $(\sigma^2, \sigma'^2) \in [v, V]$.

$$\begin{aligned}
 |||f_2(a, \sigma) - f_2(a', \sigma')||| &= \max_{i=1, \dots, q} \left| \frac{\sigma^2}{(1 - a\varepsilon_i)^2} - \frac{\sigma'^2}{(1 - a'\varepsilon_i)^2} \right| \\
 &= \max_{i=1, \dots, q} \left| \frac{\sigma^2}{(1 - a\varepsilon_i)^2} - \frac{\sigma'^2}{(1 - a\varepsilon_i)^2} + \frac{\sigma'^2}{(1 - a\varepsilon_i)^2} - \frac{\sigma'^2}{(1 - a'\varepsilon_i)^2} \right| \\
 &\leq \max_{i=1, \dots, q} \left[\frac{1}{(1 - a\varepsilon_i)^2} |\sigma^2 - \sigma'^2| + \sigma'^2 \left| \frac{1}{(1 - a\varepsilon_i)^2} - \frac{1}{(1 - a'\varepsilon_i)^2} \right| \right] \\
 &\leq |\sigma^2 - \sigma'^2| \max_{i=1, \dots, q} \frac{1}{(1 - a\varepsilon_i)^2} + \sigma'^2 \max_{i=1, \dots, q} \left| \frac{1}{(1 - a\varepsilon_i)^2} - \frac{1}{(1 - a'\varepsilon_i)^2} \right|
 \end{aligned}$$

Pour toutes les valeurs de e , la fonction $x \mapsto \frac{1}{(1-xe)^2}$ est k -lipschitz. Avec la norme infinie, on obtient donc

$$|||f_2(a, \sigma) - f_2(a', \sigma')||| \leq k \|(a, \sigma) - (a', \sigma')\|_{\infty}.$$

La fonction $(a, \sigma) \mapsto \Sigma(a, \sigma)$ est bien k -lipschitz.

C.2 Expression de la pseudo-vraisemblance pour un modèle SAR aux 4 ppv

Soit \mathbf{Y} un champ gaussien centré sur une zone P de cardinal q suivant un modèle SAR aux 4 plus proches voisins de paramètres a et σ . \mathbf{Y} est donc entièrement déterminée par sa matrice de covariance $\Sigma = \sigma^2(Id - aW_4)^{-2}$.

On définit la carte des 4, 8 ou 12 plus proches voisins notées s_4 , s_8 ou s_{12} de la façon suivante :

$$\begin{aligned}
 \forall (i, j) \in \llbracket 1, q \rrbracket^2, \quad s_4(i, j) &= Y_{i-1, j} + Y_{i+1, j} + Y_{i, j-1} + Y_{i, j+1} \\
 s_8(i, j) &= Y_{i-1, j-1} + Y_{i+1, j-1} + Y_{i-1, j+1} + Y_{i+1, j+1} \\
 s_{12}(i, j) &= Y_{i-2, j} + Y_{i+2, j} + Y_{i, j-2} + Y_{i, j+2}
 \end{aligned}$$

On peut alors montrer que pour tout point (i, j) , la densité conditionnelle de la variable $Y_{i, j}$ sachant tous les autres pixels (variable notée $Y^{i, j}$) est donnée par

$$(Y_{i, j} | Y^{i, j}) = \sqrt{\frac{1 + 4a^2}{2\pi\sigma^2}} \exp \left(- \frac{1 + 4a^2}{2\sigma^2} \left(Y_{i, j} - \frac{2a}{1 + 4a^2} s_1(i, j) + \frac{a^2}{1 + 4a^2} s_2(i, j) + \frac{2a^2}{1 + 4a^2} s_3(i, j) \right)^2 \right).$$

On en déduit alors la valeur de la pseudo log-vraisemblance conditionnelle pour une réalisation \mathbf{y} du champ centré \mathbf{Y} .

$$PLV(\mathbf{y}) = \frac{q}{2} \log \left(\frac{1 + 4a^2}{2\pi\sigma^2} \right) - \frac{1 + 4a^2}{2\sigma^2} \sum_{(i, j) \in \llbracket 1, q \rrbracket^2} \left(y_{i, j} - \frac{2a}{1 + 4a^2} s_1(i, j) + \frac{a^2}{1 + 4a^2} s_2(i, j) + \frac{2a^2}{1 + 4a^2} s_3(i, j) \right)^2$$

On cherche $(\hat{a}, \hat{\sigma})$ les estimateurs par maximum de pseudo log-vraisemblance des paramètres a et σ . En fixant a et en dérivant par rapport à σ , on montre que

$$\hat{\sigma}^2(a) = \frac{1 + 4\hat{a}^2}{M^2} \sum_{(i, j) \in \llbracket 1, q \rrbracket^2} \left(Y_{i, j} - \frac{2\hat{a}}{1 + 4\hat{a}^2} s_1(i, j) + \frac{\hat{a}^2}{1 + 4\hat{a}^2} s_2(i, j) + \frac{2\hat{a}^2}{1 + 4\hat{a}^2} s_3(i, j) \right)^2$$

Pour obtenir \hat{a} , il nous reste alors à maximiser la fonction

$$a \mapsto \frac{q}{2} \log \left(\frac{1 + 4a^2}{2\pi\hat{\sigma}(a)^2} \right) - \frac{1 + 4a^2}{2\hat{\sigma}(a)^2} \sum_{(i,j) \in \llbracket 1, q \rrbracket^2} \left(Y_{i,j} - \frac{2a}{1 + 4a^2} s_1(i, j) + \frac{a^2}{1 + 4a^2} s_2(i, j) + \frac{2a^2}{1 + 4a^2} s_3(i, j) \right)^2.$$

Grâce à la définition de $\hat{\sigma}$, on en déduit que la valeur maximum de la pseudo log-vraisemblance est

$$\log(PV(Y, \hat{a}, \hat{\sigma})) = \frac{q}{2} \left[\log \left(\frac{1 + 4\hat{a}^2}{2\pi\hat{\sigma}(a)^2} \right) - 1 \right].$$

Annexe D

Courbes COR pour un rapport signal sur bruit fixé

D.1 Classe 1

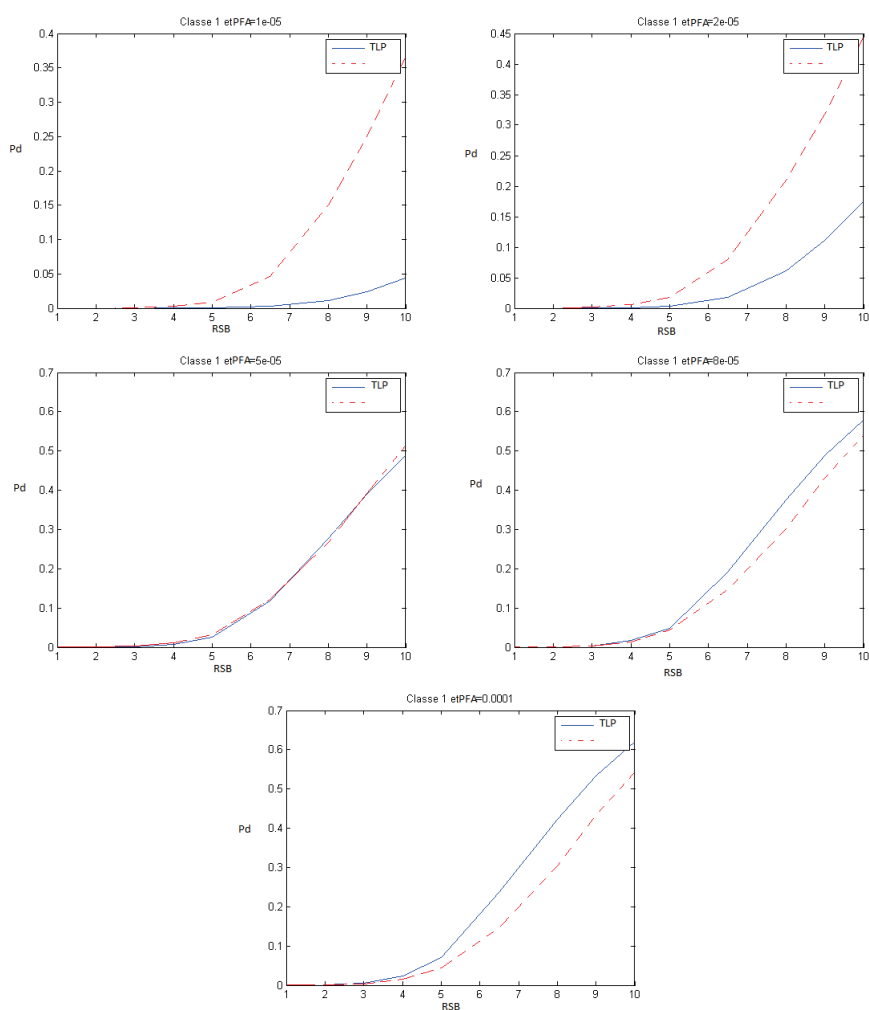


FIGURE D.1 – Courbes COR pour la classe 1 pour les 5 probabilités de fausses alarmes étudiées.

D.2 Classe 2

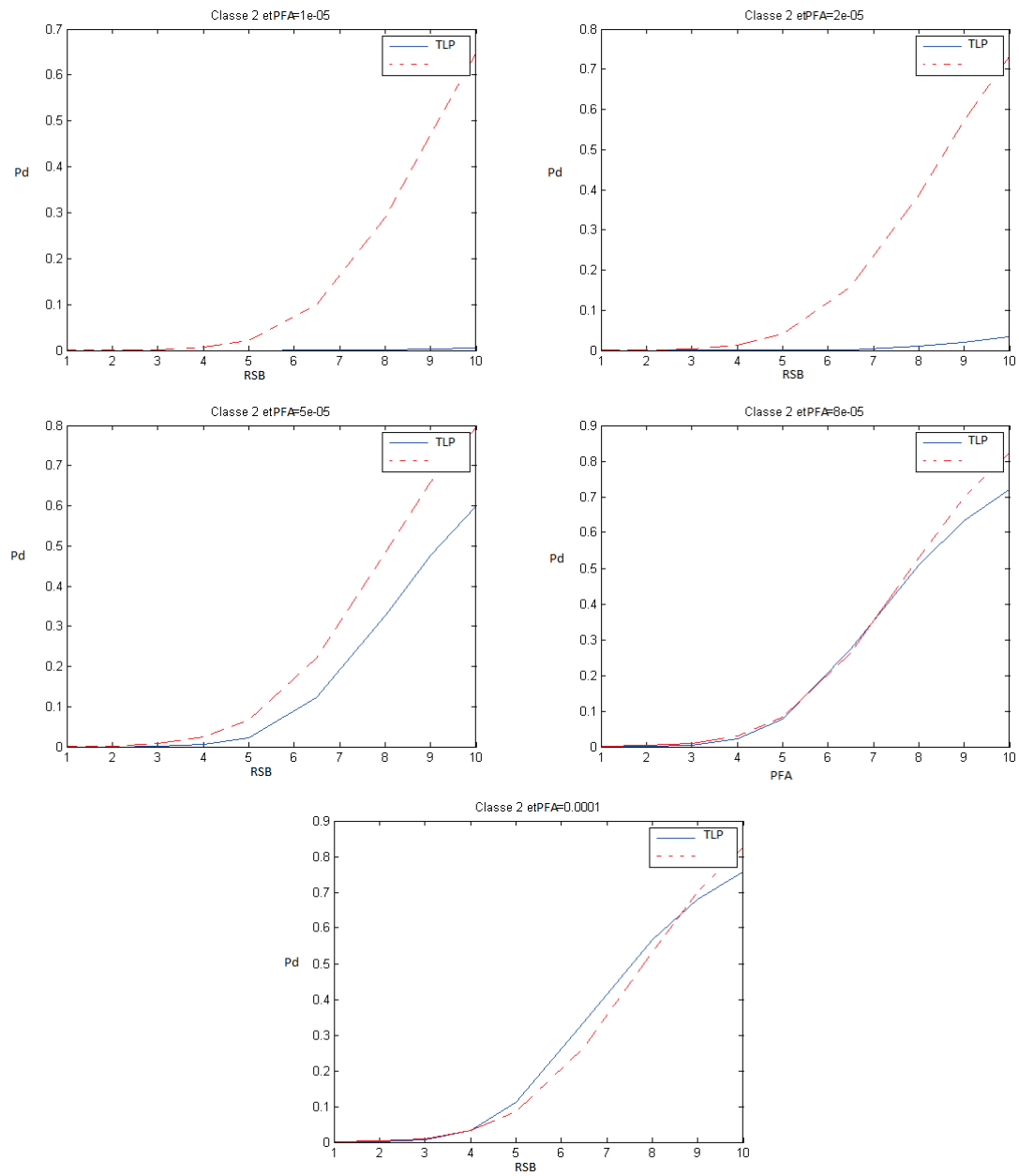


FIGURE D.2 – Courbes COR pour la classe 2 pour les 5 probabilités de fausses alarmes étudiées.

D.3 Classe 3

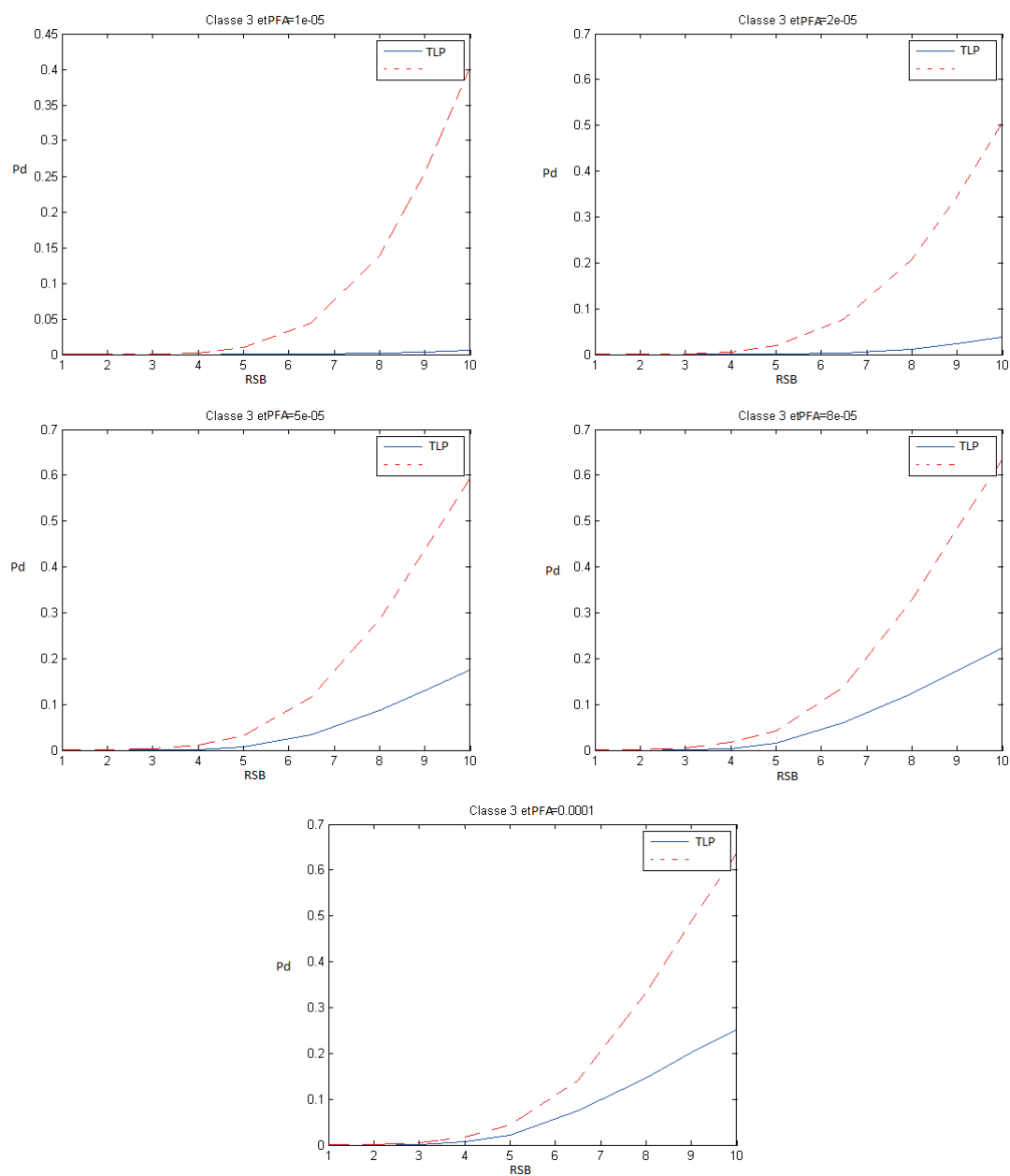


FIGURE D.3 – Courbes COR pour la classe 3 pour les 5 probabilités de fausses alarmes étudiées.

D.4 Classe 5

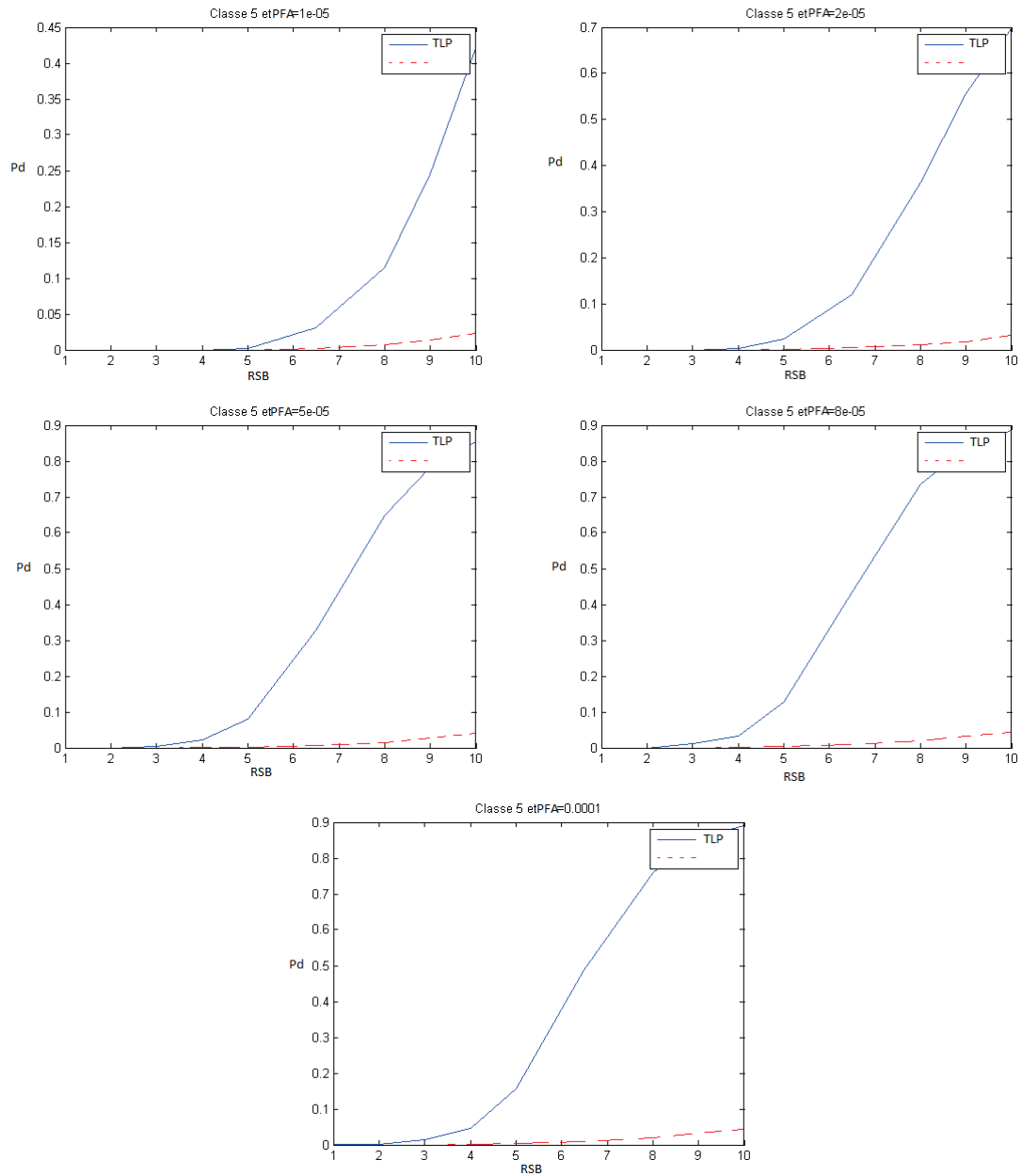


FIGURE D.4 – Courbes COR pour la classe 5 pour les 5 probabilités de fausses alarmes étudiées.

D.5 Classe 7

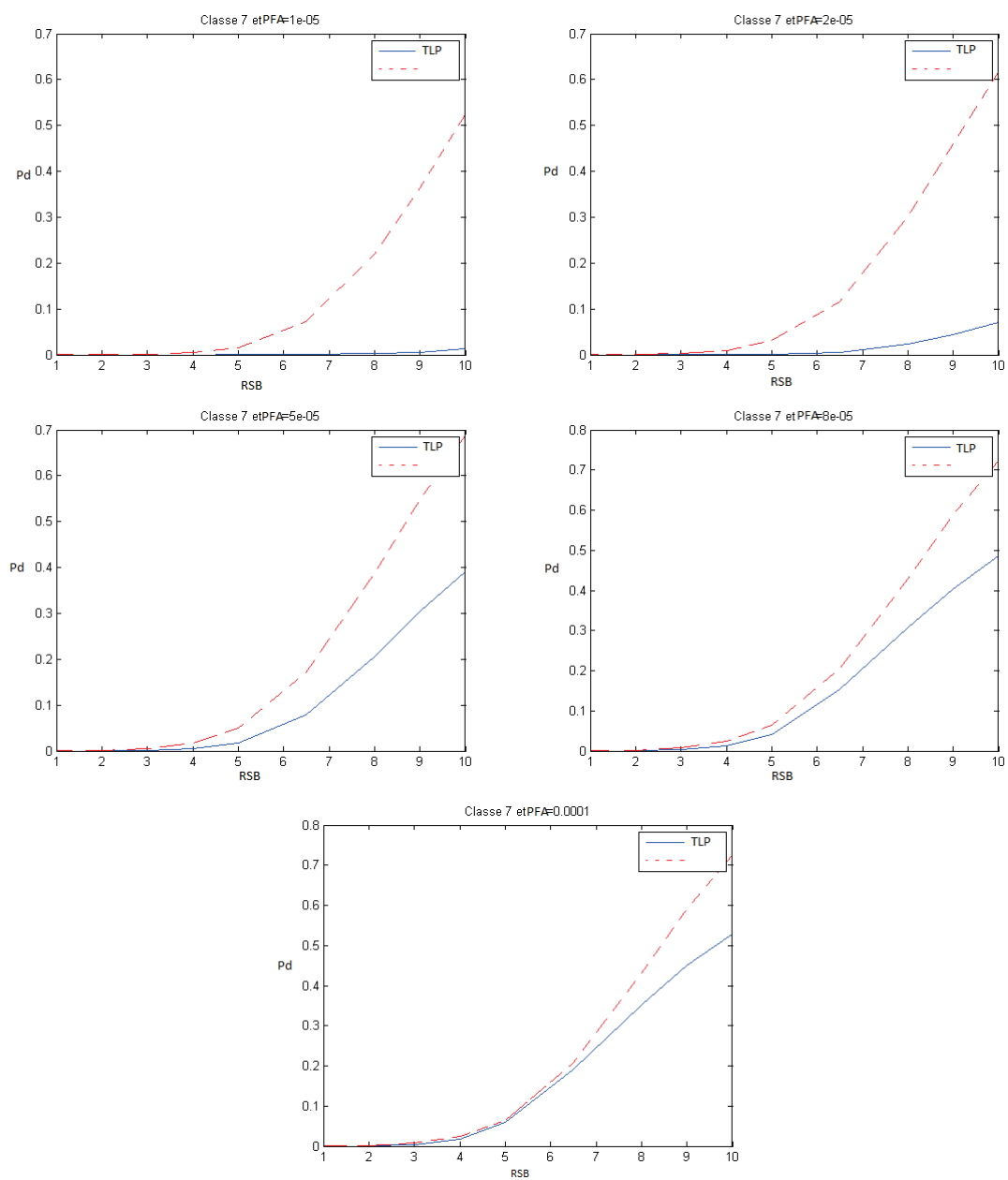


FIGURE D.5 – Courbes COR pour la classe 7 pour les 5 probabilités de fausses alarmes étudiées.

D.6 Classe 8

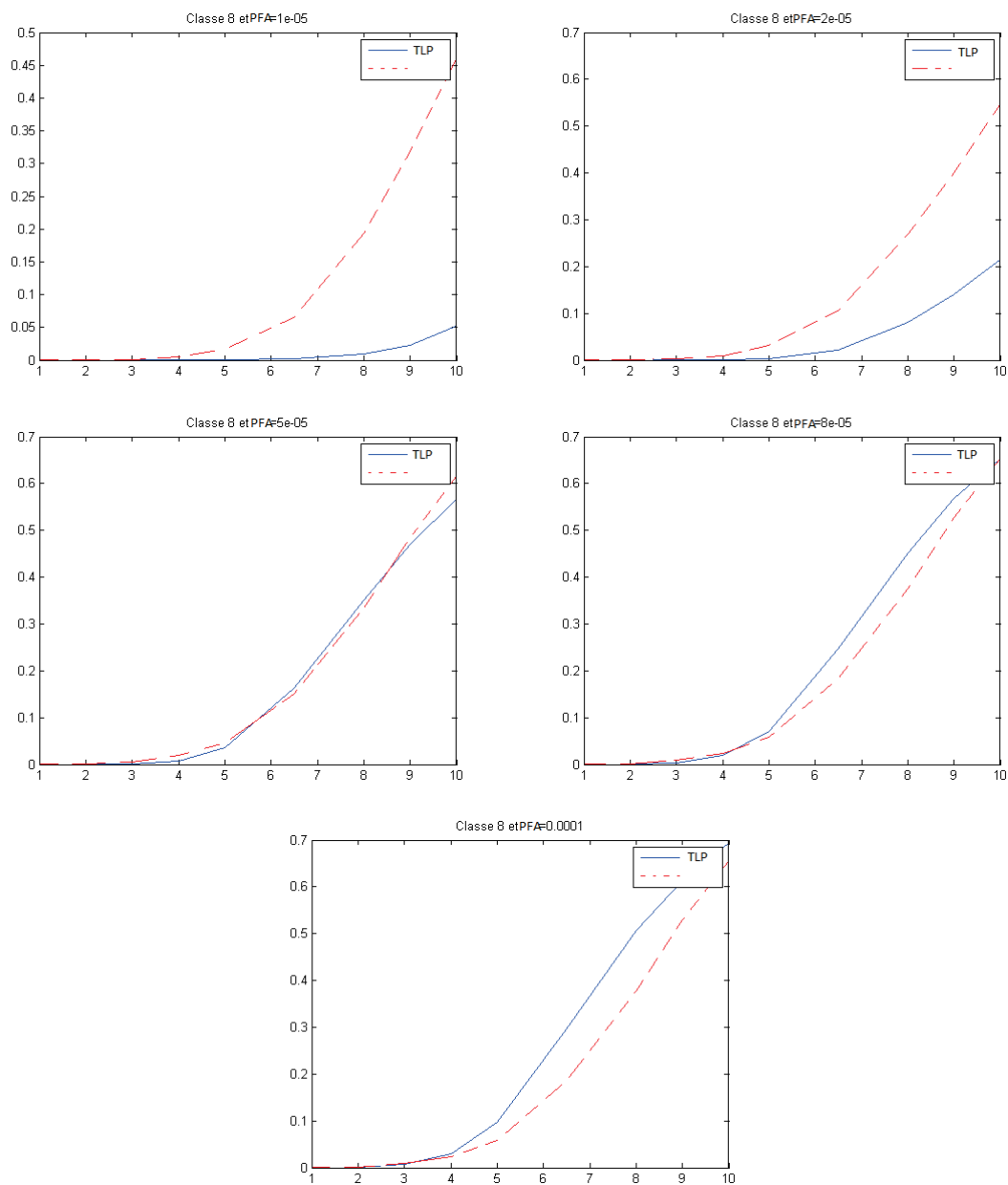


FIGURE D.6 – Courbes COR pour la classe 8 pour les 5 probabilités de fausses alarmes étudiées.

Bibliographie

- [Ach03] D. Achlioptas. Database-friendly random projections. *Journal of Computers and System Sciences*, 66(4) :671–687, 2003.
- [AM09] S. Arlot and P. Massart. Data-driven calibration of penalties for least squares regression. *Journal of Machine Learning Research*, 10 :245–279, 2009.
- [And73] T.W. Anderson. Asymptotically efficient estimation of covariance matrices with linear structure. *The Annals of Statistics*, 1(1) :135–141, 1973.
- [Arl07] S. Arlot. *Rééchantillonnage et Sélection de modèles*. PhD thesis, Université Paris-Sud 11, 2007.
- [Bau09] J-P. Baudry. *Sélection de modèle pour la classification non supervisée, Choix du nombre de classes*. PhD thesis, Université Paris-Sud XI, 2009.
- [BCG00] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7) :719–725, 2000.
- [BCM05] A. Buades, B. Coll, and J-M. Morel. A non-local algorithm for image denoising. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 60–65, 2005.
- [BCSAG08a] C. Biernacki, G. Celeux, J-F. Si Abdellah, and G. Govaert. *High Performance Model-Based Cluster and Discrimination Analysis*, 2008.
- [BCSAG08b] C. Biernacki, G. Celeux, J-F. Si Abdellah, and G. Govaert. *MIXMOD Statistical Documentation*, 2008.
- [Bes74] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, 36(2) :192–236, 1974.
- [Bes77] J. Besag. Efficiency of pseudolikelihood estimation for simple gaussian fields. *Biometrika*, 64(3) :616–618, 1977.
- [BF92] J.R. Bunch and R.D. Fierro. A constant-false-alarm-rate algorithm. *Linear Algebra and Its Applications*, 172 :231–241, 1992.
- [BFSO84] L. Breiman, J. Friedman, C. Stone, and R. Olshen. *Classification and Regression Trees*. Taylor and Francis, 1984.
- [BHL03] Y. Baraud, S. Huet, and B. Laurent. Adaptive tests of linear hypothesis by model selection. *The Annals of Statistics*, 31(1) :225–251, 2003.
- [BK99] J. Bennett and A. Khotanzad. Maximum likelihood estimation methods for multispectral random field image models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6) :537–543, 1999.
- [BLW82] J.P. Burg, D.G. Luenberger, and D.L. Wenger. Estimation of structured covariance matrices. *Proceedings of the IEEE*, 70(9) :963–974, 1982.
- [BM] J. Bruna and S. Mallat. Codes pour le calcul les coefficients de scattering. <http://www.di.ens.fr/data/scattering/>.
- [BM01] E. Bingham and H. Mannila. Random projection in dimensionality reduction : Applications to image and text data. In *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery*, 2001.

- [BM07] L. Birge and P. Massart. Minimal penalties for gaussian model selection. *Probab. Theory Related Fields*, 138(1-2) :33–73, 2007.
- [BM11] J. Bruna and S. Mallat. Classification with scattering operators. In *IEEE Conference on*, 2011.
- [BM13] J. Bruna and S. Mallat. Invariant scattering convolution network. *IEEE Trans.*, 54(5), 2013.
- [BMM11] J-P. Baudry, C. Maugis, and B. Michel. Slope heuristics : Overview and implementation. *Statistics and Computing*, 22(2) :455–470, 2011.
- [BN93] M. Basseville and I.V. Nikiforov. *Detection of Abrupt Changes : Theory and Applications*. Prentice-Hall, Inc., 1993.
- [BT11] J. Bien and R. Tibshirani. Sparse estimation of a covariance matrix. *Biometrika*, 98(4) :807–820, 2011.
- [CC09] D. Chafaï and D. Concordet. A new method for the estimation of variance matrix with prescribed zeros in nonlinear mixed effects models. *Statistics and Computing*, 19(2) :129–138, 2009.
- [CDR07] S. Chaudhuri, M. Drton, and T.S. Richardson. Estimation of a covariance matrix with zeros. *Biometrika*, 94(1) :199–216, 2007.
- [CG92] G. Celeux and G. Govaert. A classification e.m. algorithm for clustering and two staochastic versions. *Computational Statistics and Data Analysis*, 14 :315–332, 1992.
- [Cha02] A. Chambaz. Detecting abrupt changes in random fields. *ESAIM Probability and Statistics*, 6 :189–209, 2002.
- [CJ83] G.R. Cross and A.K. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(1) :25–39, 1983.
- [Cre93] N.A.C. Cressie. *Statistics for Spatial Data*. John Wiley and Sons, 1993.
- [Dah12] R. Dahlhaus. Locally stationary processes. *Handbook of Statistics*, 30, 2012.
- [Dev15] E. Devijver. *Modèle de mélange pour la régression en grande dimension*. PhD thesis, Université Paris Sud XI, 2015.
- [DF00] B.S. Denney and R.J.P. Figueiredo. Optimal point target detection using adaptive auto regressive background prediction. In *Proc. SPIE 4048*, 2000.
- [DG03] S. Dasgupta and A. Gupta. A elementary proof of a theorem of johnson and lindenstrauss. *Random Struictures and Algorithm*, 22(1) :60–65, 2003.
- [DK87] R. Dahlhaus and H. Künsch. Edge effects and efficient parameter estimation for stationary random fields. *Biometrika*, 74(4) :877–882, 1987.
- [DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal statistical Society*, 39(1) :1–38, 1977.
- [GBR03] F. Galland, N. Bertaux, and P. Réfrégier. Minimum description length synthetic aperture radar image segmentation. *IEEE Transactions on Image Processings*, 12(9), 2003.
- [GC05] A. Goldman and I. Cohen. Anomaly subspace detection based on a multi-scale markov random field model. *Signal Processing*, 85(3) :463–479, 2005.
- [Gen09] X. Gendre. *Estimation par sélection de modèles en régression hétéroscédastique*. PhD thesis, Université de Sophia Antipolis, 2009.
- [Gen13] L. Genin. *Détection d’objets de petite taille sur des séquences aériennes ou satellitaires*. PhD thesis, Université Paris 13 Nord, 2013.
- [Gen14] X. Gendre. Model selection and estimation of a component in additive regression. *ESAIM : Probability and Statistics*, 18 :77–116, 2014.
- [GG84] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6) :721–741, 1984.

BIBLIOGRAPHIE

- [Guy82] X. Guyon. Parameter estimation for a stationary process on a d-dimensional lattice. *Biometrika*, 69(1) :95–105, 1982.
- [Guy07] X. Guyon. Statistiques spatiales. In *Conférence S.A.D.A.*, 2007.
- [HKC05] C. Hory, A. Kokaram, and W.J. Christmas. threshold learning from samples drawn from the null hypothesis for the generalized likelihood ratio cusum test. In *IEEE international Workshop on Machine Learning for Signal Processing*, 2005.
- [HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
- [KC83] R.L. Kashyap and R. Chellappa. Estimation and choice of neighbors in spatial-interaction models of images. *IEEE Transactions on Information Theory*, 29(1) :60–72, 1983.
- [Kel86] E.J. Kelly. An adaptive detection algorithm. *IEEE Transactions on Aerospace and Electronic Systems*, AES-22(1), 1986.
- [KMH05] H. Kim, B.K. Mallick, and C.C. Holmes. Analyzing nonstationary spatial data using piecewise gaussian processes. *Journal of the American Statistical Association*, 100, 2005.
- [KN12] W. Kleiber and D. Nychka. Nonstationary multivariate spatial covariance modeling. *Journal of Multivariate Analysis*, 112, 2012.
- [Lé73] Lévine. *Fondements théoriques de la radiotechnique statistique*. MIR, 1973.
- [Lav98] M. Lavielle. Optimal segmentation of random processes. *IEEE Transactions on Signal Processings*, 46(5) :1365–1373, 1998.
- [Lee98] T.C.M. Lee. Segmenting images corrupted by correlated noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5) :481–492, 1998.
- [LL00] C. Ludena and M. Lavielle. The multiple change-points problem for the spectral distribution. *Bernoulli*, 6(5) :845–869, 2000.
- [LPC13] E. Le Pennec and S. Cohen. Partition-based conditional density estimation. *ESAIM, Probability and Statistics*, 17, 2013.
- [Mas03] P. Massart. *Concentration inequalities and model selection*. Springer, 2003.
- [Mas09] P. Massart. Sélection de modèle : de la théorie à la pratique. *Journal de la SFDS*, 149(4) :5–28, 2009.
- [MB92] J.M.F. Moura and N. Balram. Recursive structure of noncausal gauss-markov random fields. *IEEE Transactions on Information Theory*, 38(2) :334–354, 1992.
- [McM56] B. McMillan. Two inequalities implied by unique decipherability. *IEEE Trans. Information Theory*, 2(4) :115–116, 1956.
- [MDC10] S. Matteoli, M. Diani, and G. Corsini. A tutorial overview of anomaly detection in hyperspectral images. *IEEE Transactions on Aerospace and Electronic Systems Magazine*, 25(7), 2010.
- [Mey12] C. Meynet. *Sélection de variables pour la classification non supervisée en grande dimension*. PhD thesis, Université Paris Sud XI, 2012.
- [Mic06] O.J.J. Michel. Cours d’introduction à la théorie de la détection. Laboratoire LUAN CNRS, 2006.
- [MM11] C. Maugis and B. Michel. A non asymptotic penalized criterion for gaussian mixture model selection. *ESAIM : Probability and Statistics*, 15 :41–68, 2011.
- [MRG85] A. Margalit, I.S. Reed, and R.M. Gagliardi. Adaptive optical target detection using correlated images. *IEEE Transactions on Aerospace and Electronic Systems*, AES-21(3) :394–405, 1985.
- [Pas06] F. Pascal. *Détection et Estimation en Environnement Non Gaussien*. PhD thesis, Université Paris X-Nanterre, 2006.
- [PS] J. Portilla and E. Simonelli. Codes pour la synthèse d’images avec les descripteurs de portilla/simoncelli. <http://www.cns.nyu.edu/lcv/texture/>.

- [PS00] J. Portilla and E.P. Simoncelli. A parametric model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1) :49–71, 2000.
- [PSWS03] J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli. Image denoising using scale mixture of gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11) :1338–1351, 2003.
- [RH05] H. Rue and L. Held. *Gaussian Markov Random Field : Theory and Applications*. Chapman and Hall, 2005.
- [Ris86] J. Rissanen. Stochastic complexity and modeling. *The Annals of Statistics*, 14(3) :1080–1100, 1986.
- [Sar14] M. Sart. Estimation of the transition density of a markov chain. *Annales de l’Institut Henry Poincaré, Probabilités et Statistiques*, 50(3), 2014.
- [Sch78] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6 :461–464, 1978.
- [SF94] L. Scharf and B. Freidlander. Matched subspace detector. *IEEE Transaction On signal processing*, 42(8), 1994.
- [SF95] E.P. Simoncelli and W.T. Freeman. The steerable pyramid : A flexible architecture for multi-scale derivative computation. In *Proc 2nd IEEE Int’l Conf on Image Proc*, pages 444–447, 1995.
- [Sim98] E.P. Simoncelli. Statistical models for images : Compression restoration and synthesis. *IEEE Computer Society*, 1 :673–678, 1998.
- [The14] J. Theiler. Transductive and matched-pair machine learning for difficult target detection problems. *Proc. SPIE*, 2014.
- [Uns95] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, 4(11) :1549–1560, 1995.
- [Vap99] V.N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5) :988–999, 1999.
- [Vas11] E. Vasquez. *Techniques statistiques de détection de cibles dans des images infrarouges inhomogènes en milieu maritime*. PhD thesis, Aix Marseille Université, 2011.
- [Ver09] N. Verzelen. Data-driven neighborhood selection of a gaussian field. *Comput. Statist. Data Anal*, 54(5), 2009.
- [Ver10] N. Verzelen. Adaptive estimation of stationary gaussian fields. *Annals of Statistics*, 38(3) :1363–1402, 2010.
- [Vez78] G. Vezzosi. Détection d’un signal dans un bruit auto-régressif gaussien. *Ann. Télécommunications*, 33(7/8), 1978.
- [Whi54] P. Whittle. On stationary processes in the plane. *Biometrika*, 41(3/4) :434–449, 1954.

BIBLIOGRAPHIE

Index

Champ de Markov, 63
Courbes COR, 42
Critère ICL, 114

Distance d'Hellinger, 78
Divergence de Kullback-Liebler, 77

Entropie à crochet, 78
Erreur de première espèce, 32
Erreur de seconde espèce, 32

Heuristique de pente, 88

Intensité d'une cible, 36

Matrice de précision, 56
Modèle oracle, 77

Niveau d'un test, 34

Partitions récursives dyadiques, 82
Partitions récursives split, 84
Pixel, 34
Probabilité de détection, 35, 41
Probabilité de fausses alarmes, 35, 40
Puissance d'un test, 34

Rapport signal sur bruit, 46, 119

Stationnarité, 56

Taux de détection, 45
Taux de fausses alarmes, 44
Test d'hypothèses, 31
Test de Neyman-Pearson, 34

Voisinage non périodisé, 58
Voisinage périodisé, 58