Modélisation multivariée de champs texturaux : application à la classification d'images.

Aurélien Schutz

université de BORDEAUX

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DES SCIENCES PHYSIQUES ET DE L'INGÉNIEUR

SPÉCIALITÉ : AUTOMATIQUE, PRODUCTIQUE, SIGNAL ET IMAGE, INGÉNIERIE COGNITIQUE

Par Aurélien Schutz

Modélisation multivariée de champs texturaux

Application à la classification d'images

Sous la direction de : Yannick BERTHOUMIEU
(co-directeur : Lionel BOMBRUN)

Soutenue le 15 décembre 2014

Membres du jury :

M. ARNAUDON, Marc	Professeur Université de Bordeaux	Président
M. BERTHIER, Michel	Professeur Université La Rochelle	Rapporteur
M. CARRÉ, Philippe	Professeur Université de Poitier	Rapporteur
M. OVARLEZ, Jean-Philippe	Maître de Recherche, Docteur, HDR ONERA	Examinateur
M. BERTHOUMIEU, Yannick	Professeur Bordeaux Institut National Polytechnique	Directeur
M. BOMBRUN, Lionel	Maître de conférences, Bordeaux Sciences Agro	Co-Directeur

Titre : Modélisation multivariée de champs texturaux – Application à la classification d'images

Résumé : Le travail présenté dans cette thèse a pour objectif de proposer un algorithme de classification supervisée d'images texturées basée sur la modélisation multivariée de champs texturaux. Inspiré des algorithmes de classification dits à « Sac de Mots Visuels » (SMV), nous proposons une extension originale au cas des descripteurs paramétriques issus de la modélisation multivariée des coefficients des sous-bandes d'une décomposition en ondelettes. Différentes contributions majeures de cette thèse peuvent être mises en avant. La première concerne l'introduction d'une loi *a priori* intrinsèque à l'espace des descripteurs par la définition d'une loi gaussienne concentrée. Cette dernière étant caractérisée par un barycentre ¯_ et une varianceσ^2, nous proposons un algorithme d'estimation de ces deux quantités. Nous proposons notamment une application au cas des modèles multivariés SIRV (*Spherically Invariant Random Vector*), en séparant le problème complexe d'estimation du barycentre comme la résolution de deux problèmes d'estimation plus simples (un sur la partie gaussienne et un sur
le multiplieur). Afin de prendre en compte la diversité naturelle des images texturées (contraste, orientation, . . .), nous
proposons une extension au cas des modèles de mélanges permettant ainsi de construire le dictionnaire d'apprentissage.
Enfin, nous validons cet algorithme de classification sur diverses bases de données d'images texturées et montrons de bonnes performances de classification vis-à-vis d'autres algorithmes de la littérature.

Mots clés : classification, diversité intra-classe, loi a priori intrinsèque, modèles multivariés, texture.

Title : Multivariate modeling of texture space – Image classification application

Abstract : The prime objective of this thesis is to propose an unsupervised classification algorithm of textured images based on multivariate stochastic models. Inspired from classification algorithm named "Bag of Words" (BoW), we propose an original extension to parametric descriptors issued from the multivariate modeling of wavelet subband coefficients. Some major contributions of this thesis can be outlined. The first one concerns the introduction of an intrinsic prior on the parameter space by defining a Gaussian concentrated distribution. This latter being characterized by a centroid ¯_ and a variance _2,
we propose an estimation algorithm for those two quantities. Next, we propose an application to the multivariate SIRV (*Spherically Invariant Random Vector*) model, by resolving the difficult centroid estimation problem as the solution of two simpler ones (one for the Gaussian part and one for the multiplier part). To handle with the intra-class diversity of texture images (scene enlightenment, orientation . . .), we propose an extension to mixture models allowing the construction of the training dictionary. Finally, we validate this classification algorithm on various texture image databases and show interesting classification performances compared to other state-of-the-art algorithms.

Keywords : classification, intra-class diversity, intrinsic prior, multivariate models, texture.

Unité de recherche

IMS, UMR 5218

Remerciements

En préambule de ce mémoire je souhaite remercier plusieurs personnes qui ont permis que ce document soit publié.

Je souhaite remercier très chaleureusement le professeur Marc Arnaudon d'avoir accepté de siéger dans le jury en qualité de président. Le professeur Arnaudon a participé au contenu de la thèse sur la démonstration théorique de l'existence et l'unicité du barycentre. Ce qui me donne l'opprtunité de remercier de nouveau le professeur. Je souhaite remercier les personnes qui ont relues le document. Je remercie professeur Michel Berthier et professeur Philippe Carré pour leurs nombreux retours sur le manuscrit. Je voudrait également remercier Jean-Philippe Ovarlez pour ses retours complémentaires.

Je remercie le groupe Total pour avoir financé mes travaux et de proposer un sujet aussi novateur. Je remercie tout particulièrement le professeur Yannick Berthoumieu pour m'avoir accordé sa confiance pour ce projet ; pour m'avoir accompagné et poussé pour que j'innove ; pour me rappeler de constamment prendre du recul sur mon travail et l'apprécier dans sa globalité. Merci au professeur Jean-François Giovanelli, M. Charles Dossal et M. Laurent Duval pour m'avoir donné le socle nécessaire de connaissances mathématiques pour ce travail. Je souhaite remercier également les membres du groupe signal du laboratoire IMS dont M. Lionel Bombrun et M. Atto Abdourrahmane, pour leur pédagogie et leurs idées pour enrichir mes travaux de thèse. Je remercie également M. Olivier Scwhander, le professeur Franck Nielsen et M. Nour-Eddine Lasmar pour leurs apports sur ce mémoire.

Merci aussi à Emile, Marc, Emilie, Alexandre, Eric, Nicolas pour leur amitié et les moments partagés sur cette période de recherche. Cela m'a permis de conserver la même motivation durant l'ensemble de la thèse. Je souhaite également remercier ma famille, comme ma mère et mon père, pour leur soutien et leur amour depuis plus de vingt cinq ans. Alexis et Arthur, je souhaite vous montrer ce qu'il est possible de faire et le potentiel que l'on peut atteindre si nous restons soudés.

Table des matières

Table des figures

Liste des tableaux

Liste des Algorithmes

Chapitre I

Introduction générale

1 Contexte de l'étude

La classification d'images texturées est un problème posé par de nombreuses applications relevant de domaines très variés tels que les sciences des matériaux, les géosciences, les sciences du vivant ou la production de documents multimédias. Aujourd'hui avec l'augmentation des puissances de calcul, des capacités de stockage et la disponibilité des données à travers le réseau, le déploiement de méthodes de classification fondées sur l'apprentissage statistique n'est plus un problème insurmontable et permet d'envisager le recours à des méthodes offrant des performances de classification souvent bien supérieures aux méthodes sans apprentissage. L'apprentissage statistique désigne ici la classe de méthodes supervisées visant la résolution du problème de classification à partir d'exemples déjà observés. L'apprentissage statistique permet notamment de tirer simultanément avantage de deux familles d'approches bien connues en classification que sont les méthodes génératives et discriminatives. Les méthodes génératives impliquent une bonne connaissance de la modélisation du processus de synthèse de la donnée. Les méthodes discriminatives vont, quant à elles, s'intéresser uniquement à l'existence d'une relation souvent implicite entre les données d'entrées et la décision en ne se focalisant que sur certains aspects de la donnée pour la décrire. Ce type de méthodes ne peut être que supervisé.

Implémenter un système de classification de textures, c'est faire face à un niveau de difficulté qui dépend d'une part de la nature même des contenus texturaux adressés et d'autre part de la connaissance *a priori* de l'application finale. Concernant le premier point, il est question de la diversité de contenu. Il peut s'agir du caractère stochastique et/ou périodique des images texturées étudiées, du degré de non homogénéité des échantillons ou des possibles variations d'éclairage et/ou de couleur, des déformations spatiales linéaires ou non-linéaires etc. De même, selon les informations *a priori* du problème à disposition

Figure I.1 – Exemple d'images texturées issues de la base de données UIUC `http://www-cvr.ai.uiuc.edu/ponce_grp/data/`

(nombre de classes, base de données d'apprentissage etc.) et des ressources de calcul, des choix d'ordre méthodologiques et algorithmiques doivent être fait. Dans ce mémoire, nous nous intéresserons plus particulièrement aux méthodes de classification utilisant une base de données d'apprentissage en cherchant des approches relativement performantes face à la diversité de contenu, par rapport à la littérature et celles favorisant le compromis entre performances de classification et performances algorithmiques (temps de calcul et ressources mémoires).

Concernant la « diversité », il n'existe pas, en effet, un modèle mathématique génératif capable de synthétiser de manière fiable l'ensemble des transformations applicables aux textures naturelles. Cette diversité représente le nombre important de classes possibles et les variations observables des échantillons à l'intérieur d'une même classe. Pour ces 2 cas, nous parlerons respectivement de diversité de contenu et de diversité intra-classe. De fait, pour le problème de la classification il s'agit de trouver le descripteur ou la famille de descripteurs présentant les performances moyennes les plus élevées au regard de la tâche de classification tout en garantissant un niveau de généricité suffisant pour ne pas être trop dépendant du contenu et ainsi garantir un bon niveau d'invariance aux aléas de l'acquisition notamment. En pratique, pour mesurer la faible sensibilité à la diversité intra-classe la solution utilisée (même si elle n'est pas complètement satisfaisante) consiste à déployer des séries de tests de performances sur plusieurs bases de données en libres accès et largement utilisées par la communauté. Comme le montre le tableau I.1, les bases couvrent différents aspects du problème en termes de diversité de contenu et de diversité intra-classe.

Nom	Classes	Dimension	Couleur	Orien.	Pts vue	Illum.
Vistex	40	512x512	oui	1	1	1
Vistex Complete	167	512x512	oui	1	1	1
Alot	250	1536x1024	oui	4	4	5
AlotGrey	250	1536x1024	non	4	4	5
BrodatzChoy	20	512x512	non	1	1	1
BrodatzR	16	512x512	non	7	1	1
Brodatz Complete	111	640x640	non	1	1	1
OuTex_TC_00	24	128x128	non	1	1	1
OuTex_TC_01	24	64x64	non	1	1	1
OuTex_TC_02	24	32x32	non	1	1	1
OuTex_TC_10	24	128x128	non	9	1	1
OuTex_TC_11	24	128x128	non	1	2	1
OuTex_TC_12	24	128x128	non	9	1	3
OuTex_TC_13	68	128x128	oui	1	1	1
OuTex_TC_14	68	128x128	oui	1	1	3
OuTex_TC_15	68	128x128	oui	1	1	3
OuTex_TC_16	319	128x128	non	1	1	1
CUReT	61	200x200	oui	1	7	205
CUReT gray	61	200x200	non	1	7	205
UIUC Texture	25	640x480	non	1	40	1
STex 1024	476	1024x1024	oui	1	1	1
STex 512	476	512x512	oui	1	1	1
MeasTex		512x512	non	1	2	3
KTH-TIPS	10	200x200	oui	3	9	3
KTH-TIPS grayscale	10	200x200	non	3	9	3
KTH-TIPS2	44	200x200	oui	3	9	4
PhoTex	63	1280x1024	non	1	48	1
MondialMarmi	12	136x136	oui	9	1	1
Jerry Wu	39	256x256	oui			
PerTex	20	1024x1024	non	1	1	1
UBO2003	6	256x256	oui	1	81	81
ATRIUM Bonn BTF	4	800x800	oui	1	81	81
UTIA BTF	6	varying	oui	1	81	81
Bonn BTF	10	200x200	oui			

Tableau I.1 – Liste de base de données disponibles gratuitement. Pour chacune nous précisons (de gauche à droite) le nombre de classes texturées contenues, la dimension des images, si les images sont en couleur, le nombre d'orientations différentes du contenu avant acquisition, le nombre de points de vue différents pour l'acquisition et le nombre de conditions d'illumination différentes.

A titre d'illustration, la figure I.1 présente des échantillons tirés de la base de textures UIUC et qui sont sujets à des transformations spatiales rigides ou non-rigides. Face à cette diversité, il est donc nécessaire de développer des méthodes de classification invariantes à ces transformations spatiales.

Sur le plan des choix méthodologiques et algorithmiques pour la mise en œuvre d'un système de classification d'images texturées, il existe un large panel d'approches. En effet, le problème n'est pas nouveau et depuis une trentaine d'années de nombreux travaux lui ont été consacrés. A l'instar des méthodes de classification de documents, d'images naturelles ou de vidéo [5], les travaux récents en classification se concentrent sur le développement de méthodes dites à dictionnaire qui permettent d'« hybrider » les méthodes descriptives et génératives. Regroupées sous le sigle « sac de mots virtuels » (SMV), il s'agit de méthodes fondées sur un encodage - partie descriptive - des images à partir d'un dictionnaire de mots visuels. Le dictionnaire est soit connu *a priori* soit appris - partie générative - à partir d'une collection de descripteurs locaux évalués spatialement de manière dense. Le caractère dense correspond au calcul des descripteurs sur une grille spatiale régulière. Sur le plan méthodologique, les approches de type SMV s'organisent selon deux phases distinctes correspondant, pour la première, à la phase d'apprentissage et pour la seconde à la phase de test.

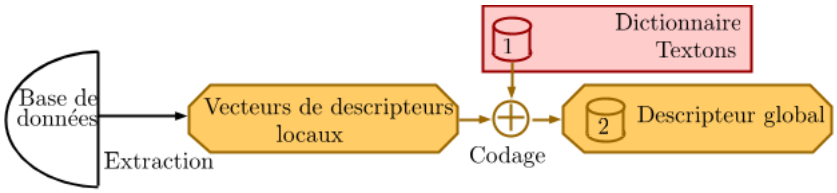


Figure I.2 – Schéma décrivant la phase d'apprentissage d'un algorithme de classification basé sur l'utilisation des sacs de mots visuels

Chacune de ces phases est construite sur la même architecture. Elles visent à fournir un codage compact de l'image à partir d'un dictionnaire fini et d'une collection d'échantillons d'un descripteur évalué localement. Pour la phase d'apprentissage, il s'agit de coder les images de la base d'apprentissage et de disposer également, pour chacune des classes, d'un code de référence (voir figure I.2). Pour la phase de test, la décision d'affectation à une classe se fera à partir de la comparaison du codage de l'image de test avec le codage des images de référence de chacune des classes (voir figure I.3). Cette comparaison peut être implémentée par différentes techniques allant du plus proche voisin à l'utilisation de méthodes évoluées telles que les machines à vecteurs de support [6].

La classification SMV, munie de ses deux phases, est soumise à trois degrés de liberté. Le descripteur choisi peut influer sur la méthode de codage. Si le descripteur fait partie d'un ensemble infini, le diction-

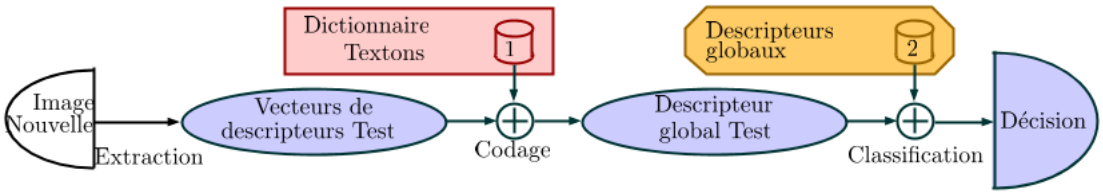


Figure I.3 – Schéma décrivant la phase de test d'un algorithme de classification basé sur l'utilisation des sacs de mots visuels

naire n'est alors plus exhaustif. Le choix de la constitution du dictionnaire réduit le nombre de méthodes de codage utilisable. Enfin, la nature du codage doit être discutée pour améliorer les performances de classification.

2 Objectifs du travail et organisation des chapitres

Le travail présenté dans ce document se focalise sur l'objectif central d'étudier l'extension de l'approche SMV au contexte de la modélisation stochastique paramétrique. En effet, plusieurs travaux récents sur l'analyse de textures ont montré la pertinence d'associer les espaces échelles et la modélisation stochastique. Il était donc naturel d'étudier la possibilité de transposer l'approche SMV au contexte de la modélisation stochastique et d'en estimer les performances. L'objectif central est de décliner l'approche SMV avec, comme descripteur, le jeu de paramètres caractérisant une densité de probabilité paramétrique approximant l'histogramme des coefficients locaux d'une décomposition par trames de l'image. Cela conduit à la formalisation de la notion de classe dans un espace géométrique contraint, appelé communément « variété » en mathématiques, et dont les dimensions correspondent respectivement au nombre de paramètres caractérisant la famille de densités de probabilités utilisée pour caractériser les textures. La notion d'« espace contraint » est relative au fait que le domaine de définition de chacun de paramètres n'est pas libre et cela doit être pris en compte dans les différentes étapes algorithmiques du système de classification (apprentissage, modèle génératif pour le dictionnaire etc.). En effet, si un modèle gaussien est utilisé pour décrire la texture, la notion de groupement, au sens par exemple de la construction d'un dictionnaire par algorithme des K-moyennes, doit se faire en tenant compte du fait que la moyenne est définie sur $\mathbb{R}$ et que la variance est positive ou nulle. Cela implique de faire appel à des fonctions coût adaptées à la géométrie des variétés considérées.

Cette étude soulève des questions d'ordre théorique et méthodologique qui dépassent largement le cadre de la classification des textures et qui concernent les notions d'optimisation, d'*a priori*, d'extensions

des concepts bayésiens au cadre des variétés. Ces questions ont motivé le travail que nous allons développer dans ce mémoire.

Après un chapitre de présentation de l'état de l'art, le chapitre 2 présente le contexte de l'analyse de textures par trames, couplée à une modélisation stochastique. Nous rappellerons, dans un premier temps, les différentes familles de lois paramétriques proposées. Nous montrerons notamment qu'il s'agit de lois non-gaussiennes dédiées aux processus :

— univariés si nous supposons l'indépendance des réalisations ;

— multivariés si nous prenons en compte les dépendances entre coefficients.

Nous aborderons par la suite la présentation de modèles adaptés aux variétés stochastiques que nous proposons pour la caractérisation du dictionnaire. Sous l'hypothèse d'un modèle hiérarchique bayésien, nous nous focaliserons sur un *a priori* intrinsèque, au sens de la contrainte géométrique, étendant le modèle gaussien et le modèle de mélange de gaussiennes, couramment utilisés, dans le contexte des variétés riemanniennes. Nous proposerons notamment un algorithme d'estimation des hyperparamètres du modèle *a priori* par maximum de vraisemblance marginalisé.

Le chapitre 3 est entièrement dédié au problème de l'estimation des mots du dictionnaire dans l'espace des descripteurs paramétriques. L'estimation des mots du dictionnaire est obtenue grâce à un algorithme d'optimisation. Nous allons présenter la fonction qui est minimisée ainsi que le processus de mise à jour utilisé dans l'algorithme.

Le chapitre 4 représente la partie pratique de ce mémoire. Nous présentons les bases de données d'images texturées qui serons utilisées dans ce mémoire. Il est alors question de valider le modèle paramétrique par rapport aux données, puis de représenter les descripteurs paramétriques dans un but illustratif. Deux implémentations de classification d'images texturées sont mises en place à partir des résultats théoriques des chapitres précédents : la classification basée sur des barycentres et des variances puis la classification SMV. Les deux implémentations sont réalisées pour des descripteurs paramétriques et sont comparées à la littérature. Le modèle stochastique reste très proche de la distribution empirique et valide l'utilisation de descripteurs paramétriques. Notre approche approfondit la théorie développée par la communauté en présentant des gains en termes de performances de classification.

Chapitre II

État de l'art sur la classification d'images texturées de type sac de mots

Contenu du chapitre

1 Introduction

Comme nous l'avons indiqué dans l'introduction générale, les méthodes dites « à sac de mots » ont connu un fort intérêt de la part la communauté et cela quel que soit le média (texte, image, vidéo, cross-média etc.). Cela s'explique par le fait qu'elles offrent un bon compromis entre facilité d'utilisation et performances de classification. Ces méthodes partent d'un constat relativement intuitif. Une source d'informations peut être discriminée de manière efficace s'il est possible de trouver un dictionnaire de taille finie sur la base duquel l'information peut être projetée ou codée. Par exemple, un texte administratif ne présentera pas la même récurrence de certains mots qu'un texte issu de la littérature enfantine.

À partir d'un jeu fini de mots, l'étude des récurrences de ces mots peut donc permettre de discriminer la nature de textes traitant de thèmes divers tel que le sport, la politique, la littérature jeunesse etc. Dans le contexte des images, le concept de « mots » ou de dictionnaire est évidemment plus problématique. Un des enjeux pour ce type d'approche est donc de trouver une définition adaptée du concept de « mots » notamment dans le cas des images texturées, objet de notre étude.

Il s'agit donc de trouver un nombre fini d'instances numériques appelé « dictionnaire » et permettant de coder l'information globale, l'image, à partir d'un ensemble de mesures locales, le descripteur. Dans ce chapitre nous allons présenter un état de l'art des méthodes SMV appliquées aux images texturées. La mise en place d'un dictionnaire est donc une des clés des méthodes de type SMV. En pratique, les travaux antérieurs se sont concentrés sur deux types de dictionnaires selon qu'ils sont définis *a priori* ou *a posteriori*.

2 Définition d'un dictionnaire *a priori* de « mots ou motifs visuels »

Concernant les textures, l'article de Fernandez et al de 2013 [1] fait une synthèse très complète des approches prenant en compte un dictionnaire *a priori*. Les auteurs font un bilan des différents descripteurs proposés dans la « littérature » depuis une trentaine d'années. À titre illustratif, nous reprenons le tableau II.1 issu de l'article qui reprend l'ensemble des descripteurs testés. En règle générale, le descripteur est extrait à partir d'un voisinage de taille finie du pixel courant et fournit une valeur entière codant le voisinage. Le nombre d'états possibles relatifs au voisinage choisi étant fini, les P états (ou

mots) constituent le dictionnaire. La taille du dictionnaire peut fortement varier en fonction de la nature du descripteur, de la taille du voisinage, du fait que le descripteur intègre des aspects multi échelles et prenne en compte l'invariance par rotation etc ... Une grande majorité de ces approches exploite ce qui est communément appelé l'« histogramme des motifs équivalents » (HEP en anglais pour histograms of equivalent patterns). Il s'agit en fait du système de codage qui est ici l'histogramme où les probabilités d'apparition des mots. La formulation générique est donnée par :

$$h_k = \frac{1}{Z} \sum_{x=1}^{L} \sum_{y=1}^{H} \delta \left[f(V^{\Omega}_{I(x,y)}) - k \right], \qquad k = 1, \ldots, P$$

avec I une images de dimensions $H \times L$ pixels, $V^{\Omega}_{I(x,y)}$ l'ensemble des niveaux de gris des pixels voisins du pixel $I(x, y)$ de l'image I à la position (x, y), f une fonction projetant le niveau de gris dans le dictionnaire de P valeurs k, δ le symbole de Kronecker renvoyant 1 si son contenu est nul et 0 sinon et Z une valeur de normalisation.

Sans être exhaustif, nous citerons les descripteurs fondés sur les matrices de co-occurrences de niveau de gris (GLCM [7]) ou les motifs binaires locaux (LBP [8]) et toutes leurs variantes. Les travaux précurseurs d'Haralick et al. [7] avec les GLCM ont en effet ouvert la voie à de nombreuses propositions visant à extraire une signature à partir du voisinage du pixel courant.

2.1 Les matrices de co-occurrence de niveaux de gris (GLCM)

Les GLCM représentent une approche statistique [9] qui diffère des méthodes géométriques utilisant les primitives. La figure II.2 présente l'extraction de GLCM, avec $N_g = 4$ le nombre de nuances de gris considéré. Il est question d'étudier la fréquence d'apparition d'un couple de niveaux de gris. Les niveaux de gris sont associés à deux pixels de l'image séparés d'une distance spatiale, notée d, fixée. Les fréquences d'apparition des couples de nuances de gris sont rangées dans une matrice associée à la distance d et l'angle θ.

2.2 Les motifs binaires locaux (LBP)

Ojala et al. [8] présentent les LBP comme un descripteur qui est invariant aux niveaux de gris. La figure II.3 montre le fonctionnement des LBP d'Ojala et al. [8]. Soit un pixel V_0 de l'image dont il faut calculer le code LBP, nous considérons alors les 8 pixels voisins ordonnés de gauche à droite puis de haut en bas : $V_1, \ldots, V_8$. V_0 correspond au niveau de gris du pixel $I(x_0, y_0)$ positionné en (x_0, y_0) dans l'image

Name	Acronym	Dim.	Ker. fun.	Ref.	Year
Local methods					
Grey level co-occurrence matrices	GLCM	G^2	Eq. 12	[34]	1973
Grey level differences	GLD	G	Eq. 14	[120]	1976
Sum and difference histograms	SDH	$2(2G-1)$	Eqs. 16,17	[110]	1986
Texture spectrum (0)	TS0	3^8	Eq. 20	[38]	1990
Texture spectrum (Δ)	TSΔ	3^8	Eq. 21	[40]	1992
Rank transform	RT	9	Eq. 22	[129]	1994
Reduced texture units	RTU	45	Eq. 27	[58]	1995
Gray level texture co-occurrence spectrum	GLTCS+	4!	Eq. 28	[45]	1996
Local binary patterns	LBP, CLBP_S	2^8	Eq. 29	[82]	1996
Simplified texture spectrum	STS	3^4	Eq. 30	[128]	2003
Simplified texture units (+)	STU+	3^4	Eq. 31	[71]	2003
Simplified texture units ($\times$)	STU$\times$	3^4	Eq. 32	[71]	2003
Modified texture spectrum	MTS	2^4	Eq. 33	[128]	2003
Improved local binary patterns	ILBP	2^9-1	Eq. 34	[50]	2004
Gradient texture unit coding	GTUC	$2\cdot 3^7$	Eq. 37	[13]	2004
3D Local Binary Patterns	3DLBP	$4\cdot 2^8$	Eqs. 29,38	[46]	2006
Center-symmetric local binary patterns	CS-LBP	2^4	Eq. 40	[42]	2006
Median binary patterns	MBP	2^9-1	Eq. 41	[33]	2007
Local ternary patterns	LTP	$2\cdot 2^8$	Eqs. 44,45	[107]	2007
Centralized binary patterns	CBP	2^5	Eq. 47	[24]	2008
Improved center-symmetric local binary patterns (D)	D-LBP	2^4	Eq. 49	[126]	2009
Improved center-symmetric local binary patterns (ID)	ID-LBP	2^4	Eq. 50	[126]	2009
Improved local ternary patterns	ILTP	$2\cdot 2^9$	Eqs. 52,53	[75]	2010
Local quinary patterns	LQP	$4\cdot 2^8$	Eq. 56	[74]	2010
Binary gradient contours (1)	BGC1	2^8-1	Eq. 58	[22]	2011
Binary gradient contours (2)	BGC2	2^8-1	Eq. 59	[22]	2011
Binary gradient contours (3)	BGC3	$(2^4-1)^2$	Eq. 60	[22]	2011
Center-symmetric texture spectrum	CS-TSΔ	3^4	Eq. 61	[130]	2011
Improved center-symmetric texture spectrum	ICS-TSΔ	$2\cdot 3^2$	Eqs. 62,63	[130]	2011
Gradient-based local binary patterns	GLBP	2^8	Eq. 65	[41]	2011
Improved binary gradient contours (1)	IBGC1	$2\cdot(2^8-1)$	Eq. 67	This paper	
Global methods					
Binary texture co-occurrence spectrum	BTCS+	2^4	Eq. 68	[90]	1991
Coordinated clusters representation	CCR	2^9	Eq. 69	[61]	1996
Completed local binary patterns (C)	CLBP_C	2	Eq. 70	[31]	2010
Completed local binary patterns (M)	CLBP_M	2^8	Eq. 72	[31]	2010

Figure II.1 – Liste non exhaustive des descripteurs existant dans le domaine des images. Fernandez et al.[1]

I. L'image I étant de dimension $L \times H$, toutes les combinaisons d'entiers $x_0 = 1, \ldots, L$ et $y_0 = 1, \ldots, H$ sont considérées. La distribution des descripteurs LBP étant l'objet d'intérêt. La figure II.4 montre que les pixels voisins se positionnent par rapport au pixel central.

Ojala et al. souhaitent utiliser la distribution $\tilde{t}(V_0, V_1, \ldots, V_8)$ des pixels tout en la rendant invariante aux niveaux de gris du pixel V_0. En résumé, le processus de génération du code $LBP(V_0)$ est constitué de trois étapes.

1. La première étape consiste à modifier la valeur du pixel V_p, pour tout $p = 1, \ldots, 8$, pour être invariant au niveau de gris du pixel central V_0. Dès lors, la distribution des différences $V_i - V_0$ est indépendante de la distribution de V_0 d'après Ojala et al., ce qui constitue la première étape vers une invariance aux niveaux de gris.

2. La seconde étape est un résultat de l'étude sur la distribution de l'amplitude des différences $V_p - V_0$. L'amplitude des différences $V_p - V_0$ reste moins importante que le signe $s(V_p - V_0)$ de ses différences pour détecter les zones plates des zones avec des contours. L'étape de seuillage consiste à associer à chaque pixel voisin V_p l'index $s(V_p - V_0) = 1$ si le niveau de gris V_i est strictement supérieur au niveau de gris du pixel central V_0 ($V_p - V_0 > 0$) et 0 sinon.

3. La troisième et dernière étape consiste à construire un code LBP unique et spécifique à ce pixel V_0. Les voisins V_p ayant un niveau de gris vérifiant $s(V_p - V_0) = 1$ est codé par la puissance 2^p, pour $p = 1, \ldots, 8$. Et le tout est additionné pour former le code LBP de V_0.

En résumé, le code LBP du pixel V_0 est donné par :

$$\mathrm{LBP}(V_0) = \sum_{p=1}^{8} s(V_p - V_0) 2^p.$$

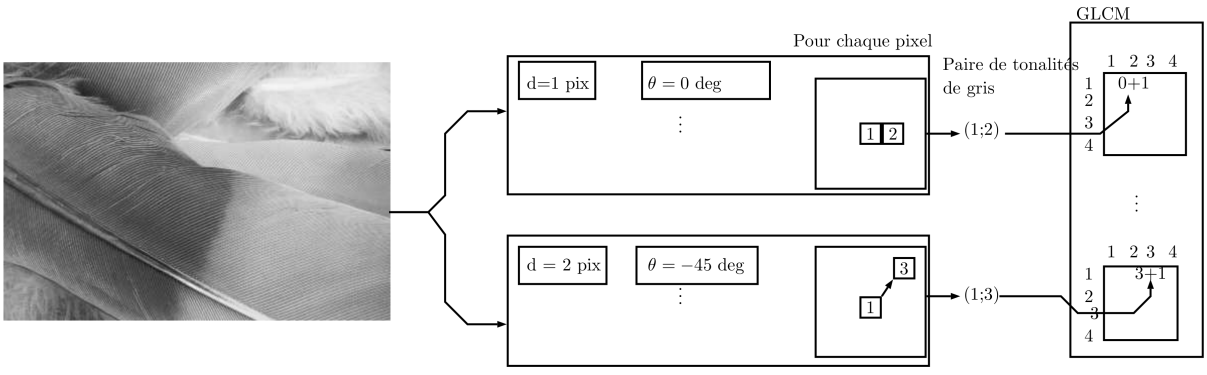


Figure II.2 – Extraction de GLCM sur une image

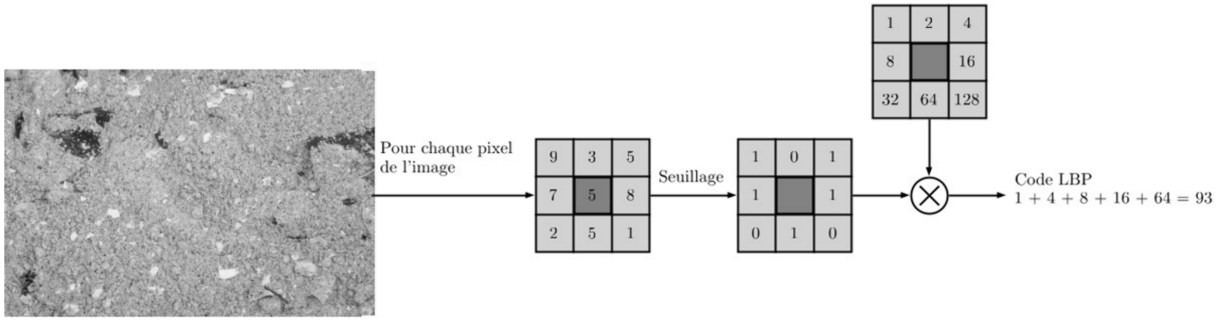


Figure II.3 – Extraction de codes LBP pour chaque pixel d'une image

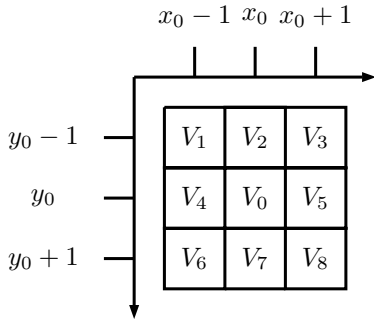


Figure II.4 – Représentation des huit pixels voisins de (x_0, y_0) avec V_0 comme niveaux de gris. Ce schéma permet de visualiser la position relative du niveau de gris V_i par rapport au niveau de gris V_0, $\forall i = 1, \ldots, 8$. L'unité utilisé dans ce schéma est le pixel.

Si le niveau de gris du pixel central V_0 tend vers les niveaux de gris limites 0 (resp. N_g), dans ce cas l'ensemble des niveaux de gris dans les pixels voisins sont positifs (resp. inférieurs à N_g) ; il n'est alors plus possible d'avoir des différences $V_p - V_0$ négatives (resp. positives). Le code LBP ainsi défini est indépendant du niveau de gris du pixel central V_0 tant que cela ne modifie pas les seuils $s(V_p - V_0)$ des pixels voisins. Après les trois étapes, il est possible de considérer la variable aléatoire $\mathrm{LBP}(V_0)$ comme image du vecteur $(V_0, V_1, \ldots, V_8)$ défini au début du paragraphe. La variable aléatoire $\mathrm{LBP}(V_0)$ est supposée indépendante sur toute l'image I et identiquement distribuée selon la distribution t. La densité de probabilité t est reliée de façon non explicite à la densité de probabilité $\tilde{t}$ du vecteur $(V_0, V_1, \ldots, V_8)$. Il est important de noter que l'ordre des pixels voisins imposé rend ce descripteur dépendant de l'orientation de l'image.

Les différentes familles de descripteurs cités plus haut et qui visent à quantifier numériquement les occurrences de motifs locaux dans le domaine spatial ou fréquentiel utilisent des dictionnaires définis *a priori*. Certain auteurs ont montré que la simple prise en compte des intensités dans un voisinage spatial limité, appelé « patch », peut être un descripteur efficace pour construire un dictionnaire tout à fait discriminant. En effet Varma et al. [10] montrent qu'à partir de patchs de petite taille (5x5) ou (7x7),

ils obtiennent des performances équivalentes voire supérieures à celles obtenues avec des descripteurs concaténant les réponses d'une série de bancs de filtres directionnels qu'ils avaient proposé précédemment [11]. C'est d'ailleurs l'une des méthodes les plus performantes sur le sujet. Dans le même ordre d'idées, à partir de travaux dans le domaine de l'acquisition compressée, Liu et al. [12] ont proposés de construire des descripteurs obtenus à partir des projections aléatoires des intensités des patchs.

3 Définition d'un dictionnaire *a posteriori* de « mots ou motifs visuels »

Les différentes méthodes étudiées dans le paragraphe précédent exploitent des descripteurs permettant de définir des dictionnaires *a priori*. L'intérêt de ces approches est donc leur facilité d'utilisation et d'implémentation mais la structuration *a priori* du dictionnaire constitue leur « faiblesse » car elle impose une extrême simplicité du dictionnaire sous la forme d'un codage binaire, octal etc. Certains travaux se sont donc attachés à développer des méthodes fondées sur un apprentissage du dictionnaire ce qui conduit généralement à de meilleures performances de classification. En règle générale, il s'agit d'un dictionnaire construit directement dans l'espace du descripteur local et dont les mots ou motifs caractérisent les échantillons les plus représentatifs. A partir d'une collection d'échantillons du descripteur local, généralement extrait à partir d'une grille spatiale régulière, l'estimation des éléments du dictionnaire est réalisée (voir figure II.5).

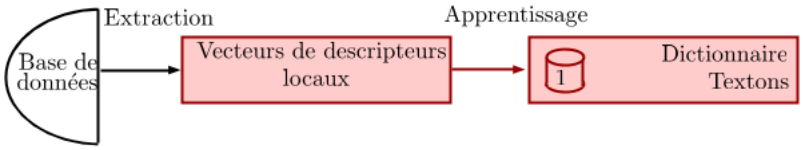


Figure II.5 – Schéma de l'apprentissage du dictionnaire

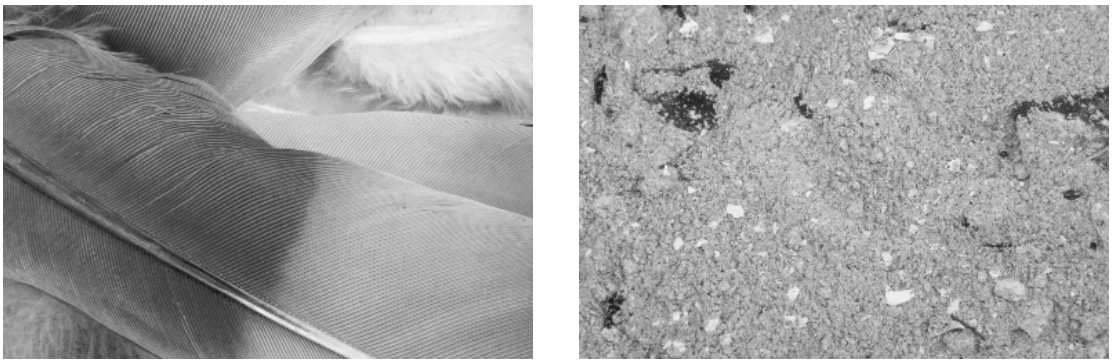
La construction du dictionnaire est donc intimement liée à la nature mathématique du descripteur et des propriétés géométrique de l'espace où il est défini. En pratique, afin de trouver les mots, ou « textons » dans le cadre des textures, il est nécessaire de résoudre une procédure optimale dont le coût calculatoire peut fortement varier en fonction de comment est modélisé le problème. Sur le plan de la modélisation, l'approche fondée sur un modèle génératif permettant de caractériser au mieux les variations d'apparence du descripteur. Le modèle génératif est probabiliste et sera souvent exploité dans un schéma bayésien plus ou moins complexe selon que l'on utilise des lois *a priori* sur le modèle ou si on se ramène à une

simple vraisemblance. Cette dernière est la plus souvent utilisée.

Dans une première sous-section est présenté un exemple visuel : l'apprentissage lorsque le descripteur local est une petite portion de l'image d'origine. Cette première sous-section est suivie par une qui présente une modélisation mathématique de l'apprentissage. Enfin, la troisième sous section présente un apprentissage dans lequel une forte indépendance des textons est privilégiée. Cet état de l'art, loin d'être exhaustif, montre l'étendue des solutions qui existent.

3.1 Exemple : apprentissage d'un dictionnaire

A titre d'illustration, nous allons nous focaliser sur la méthode de Varma et al. [11] qui se fonde sur la simple prise en compte des intensités dans un voisinage spatial limité, appelé communément « patch ». Ses auteurs montrent qu'à partir de patchs de petite taille (5x5) ou (7x7) pixels, ils obtiennent des performances supérieures à celles obtenues avec les méthodes à dictionnaire *a priori* et montrent également que le fait de travailler avec des voisinages concaténant des réponses de bancs de filtres directionnels qu'ils avaient proposés précédemment n'apporte que peu ou aucun gain en performances [10].



(a) exemple d'image orientée - plumes - (b) exemple d'image stochastique - cendres -

Figure II.6 – Exemples d'images texturées (a) anisotrope et (b) stochastique. Amsterdam Library of Textures (ALOT) [2]

Sur la base des travaux de Zucker et Terzopoulos [13], considérons le cas de la classification avec 2 classes : une classe présentant une texture anisotrope (orientée, voir figure II.6.(a)) et une autre plutôt stochastique (voir figure II.6.(b)). La méthode SMV va apprendre les textons (motifs caractéristiques) de chacune de ces 2 classes de texture.

Dans un premier temps, les textons de chacune des classes sont appris sur la base de données d'apprentissage. Pour se faire, pour chacune des classes d'apprentissage, des vecteurs de descripteurs sont

extraits des images. Ces descripteurs peuvent être des patchs locaux (voir figure II.7), des paramètres de texture (GLCM, LBP, ...). Une méthode de clustering (de type K-moyennes) est utilisée pour estimer les textons représentatifs de la classe étudiée (les patchs situés à droite de la figure II.7 représentent différents aspects d'une plume d'oiseau et pourraient servir de mots visuels). L'ensemble de ces textons pour chacune des classes forme ainsi le dictionnaire de textons.

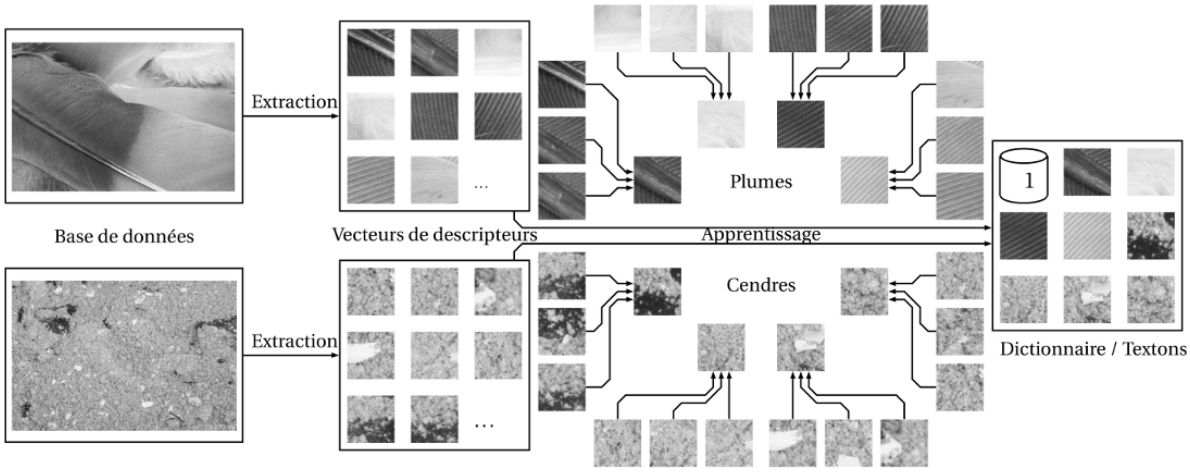


Figure II.7 – 1. Phase d'apprentissage. Chaque image est décomposée en 96 patchs. Dans chaque classe, un algorithme de K-moyennes construit 4 textons par classe. Le dictionnaire est constitué de l'ensemble des textons calculés. Amsterdam Library of Textures (ALOT) [2]

Cet apprentissage est qualifié de « hard-clustering » car chaque patch ne peut appartenir qu'à une classe. Comme une classe n'est représentée que par un texton, le patch appartient à cette classe car il est plus similaire à ce texton que tous les autres. Par cette méthode, tous les patchs sont représentés par au moins une classe de la base d'apprentissage, surtout pour les classes qui ne sont pas présentes dans cette base. Cette méthode est alors confrontée au problème de la pertinence de la classe estimée. La section suivante présente un modèle mathématique qui permet, entre autres, de retrouver le « hard-clustering » et de définir le « soft-clustering ».

3.2 Approches exploitant le modèle de densité de mélange de la vraisemblance

Apprendre un dictionnaire c'est, en premier lieu, apprendre des textons. Une classe peut donc être explicitée par plusieurs mots. Le vocabulaire utilisé couvre une classe dans une grande partie de sa complexité. L'idée est de représenter l'impact des mots choisis au moyen d'une variabilité autour de ce texton. Par exemple, une distribution paramétrique telle que la loi mélange de gaussiennes (MMG) peut

être utilisée pour représenter une classe.

Toutefois, certains travaux proposent, comme alternative à la distribution gaussienne, d'utiliser des distributions non-gaussiennes notamment lorsque certaines hypothèses sur le bruit peuvent être faites [14] ou si le caractère paramétrique du modèle est remis en cause [15].

Un modèle MMG apporte une grande flexibilité de par sa définition de somme pondérée de composantes de distribution gaussienne. En effet, chaque composante est la donnée d'une moyenne μ_p, d'une variance Σ_p et d'un poids w_p strictement positif. La notion de pondération évaluée et non-binaire associées aux composantes représente un argument de choix par rapport au « hard-clustering » présenté dans la section précédente. De ce fait, apprendre un dictionnaire par l'estimation des paramètres du modèle MMG est qualifié de « soft-clustering ».

La suite de cette section présente trois paragraphes : les notations utilisées par le modèle MMG et le vocabulaire associé ; l'état de l'art sur l'estimation des paramètres d'un modèle MMG suivi d'un exemple concret. Théoriquement, si on note P le nombre de composantes de la loi mélange alors le modèle MMG est donné par :

$$f(x \mid \theta, P) = \sum_{p=0}^{P-1} w_p \mathcal{N}(x \mid \mu_p, \Sigma_p)$$

où le vecteur de paramètres inconnus est $\theta = \{w_p; \mu_p; \Sigma_p\}$. La loi $\mathcal{N}(x \mid \mu_p, \Sigma_p)$ est une distribution gaussienne de moyenne μ_p et de matrice de covariance Σ_p. Ensuite ce modèle de référence peut se voir simplifié. Il est courant de trouver les hypothèses suivantes qui peuvent évidemment se combiner :

- Hypothèse diagonale : les matrices de covariance sont supposées être des matrices diagonales.
- Hypothèse d'homogénéité : toutes les composantes du vecteur x sont supposées avoir la même variance.
- Hypothèse d'homoscédasticité : toutes les composantes de la loi mélange sont supposées avoir la même covariance.
- Hypothèse d'équiprobabilité du modèle : les poids de la loi mélange sont supposés tous identiques et égaux à $1/P$.

Le choix de modélisation exploitant une ou plusieurs de ces hypothèses est le fruit d'un compromis entre performances de classification, nombres de données disponibles et charge de calcul.

Du point de vue de l'estimation, toutes les approches – maximum de vraisemblance, bayésiennes, *a posteriori* et moyenne *a posteriori* – peuvent être envisageables et, d'un point de vue général, nous en connaissons les exigences et les performances. Considérons la méthode du maximum de vraisemblance, quel que soit le modèle paramétrique, le problème du maximum de vraisemblance se résout en utilisant

un algorithme en deux étapes EM (Espérance-Maximization). La maximisation directe est en effet plus complexe car adressant un problème non-linéaire qui nécessite l'utilisation d'une méthode gloutonne d'échantillonnage. L'idée première de l'algorithme EM est de commencer avec un jeu de paramètres initiaux $\hat{\theta}$ et d'estimer un nouveau jeu de valeurs pour les paramètres θ tels que

$$f(x \mid \theta) > f(x \mid \hat{\theta}).$$

Le nouveau jeu de valeurs devient alors l'initialisation et le procédé est répété jusqu'à stabilisation de l'estimé.

Considérons un exemple avec les hypothèses d'homoscédasticité et d'équiprobabilité. Soit S un ensemble de patchs x qui doit être partitionné en P sous-ensembles $(S(p))_{p=1}^{P}$ aussi appelées composantes. Ce modèle hyper simplifié nous conduit au critère d'estimation suivant

$$\{\hat{\mu}_p\}_{p=1,\dots,P} = \underset{\{\mu_p\}\in\mathbb{R}^p}{arg\,\min} \sum_{p=1}^{P} \sum_{x\in S(p)} \parallel x - \mu_p \parallel_2^2$$

Ce critère, et son schéma de résolution, sont aussi communément regroupés sous la désignation de l'algorithme des P-moyennes où $S(p)$ représente le sous-ensemble des descripteurs appartenant à la composante p. Le dictionnaire se résume alors aux p moyennes $(\mu_p)_{p=1}^{P}$.

La loi mélange permet de décrire plusieurs méthodes d'apprentissage de la littérature depuis le « soft-clustering » en passant par l'algorithme des P-moyennes. L'augmentation de la pertinence de l'algorithme de classification passe alors par les choix effectués lors de la construction du dictionnaire *a posteriori*. Plus la densité de probabilité est fidèle à la dispersion des descripteurs paramétriques dans la nature, moins d'erreurs peuvent être faites. Néanmoins l'algorithme des P-moyennes utilisé pour certaines applications impliquant de très gros dictionnaires, reste un algorithme lent car similaire en complexité à un algorithme glouton de recherche des P plus proches voisins. Certaines solutions utilisent des arbres de codages comme l'algorithme des P-moyennes hiérarchiques ou des arbres aléatoires [16].

3.3 Approches exploitant un modèle de dépendance linéaire entre les descripteurs observés et les textons du dictionnaire

Apprendre un dictionnaire c'est estimer la forme optimale des classes pour assurer la pertinence de la classification. Lee et al. [17] indiquent que les algorithmes décrits précédemment ont une forte complexité

calculatoire, sûrement due à un grand dictionnaire « overcomplete ». Un mot appris peut être synonyme d'un mot précédemment appris. La parcimonie du dictionnaire est recherchée et elle peut être obtenue par une optimisation alternée entre les codes et le dictionnaire. La densité de probabilité f_E de l'écart du dictionnaire avec les descripteurs $x_e = x - \mu_p$ est proportionnelle à la valeur exponentielle de l'opposé du produit d'une constante α avec une fonction parcimonieuse $\phi(.)$:

$$f_E(x_e) \propto \exp\{-\alpha\phi(x_e)\}$$

Prenons comme exemple : $\phi(x_e) = \| x_e \|_1$ la fonction de pénalité L_1 ; $\phi(x_e) = (x_e^2 + \epsilon)^{1/2}$ la fonction de pénalité epsilon L_1 ; $\phi(x_e) = \log\{1 + x_e^2\}$ la fonction de pénalité logarithmique. Sans perte de généralité, seule la fonction de pénalité L_1 sera utilisée pour la suite de ce chapitre.

L'optimisation alternée se fait entre un dictionnaire parcimonieux D et le code a. Deux cas se présentent : 1) soit le texton μ_p est dissimilaire à tous les autres mots du dictionnaire ; 2) soit le texton peut être obtenu par une combinaison linéaire d'éléments du dictionnaire parcimonieux. Le dictionnaire parcimonieux D contient dans chacune de ses colonnes un texton représentatif et dissimilaire des autres textons de D. Le code a lui, respectivement : 1) est nul avec une seule valeur non nulle associée au texton de D ; 2) contient peut de valeurs non nulles, formant une combinaison linéaire de textons issus de D. l'écart du dictionnaire avec les descripteurs se réécrit alors

$$x_e = x - Da$$

Pour aller plus loin, il est possible d'intégrer l'erreur de classification dans l'apprentissage du dictionnaire. Cette erreur est paramétrée par une matrice de variance-covariance Σ_e. La modèle stochastique s'écrit :

$$f_E(x \mid a, D, \Sigma_e) = \frac{1}{|\Sigma_e|^{1/2}(2\pi)^{d/2}} \exp\left\{-\frac{1}{2}(x - Da)^T \Sigma_e^{-1}(x - Da)\right\}$$

avec D le dictionnaire parcimonieux et a est un vecteur de coefficients de la combinaison linéaire qui relie le descripteur x avec les différents textons du dictionnaire parcimonieux. Nous poserons également P_S le nombre de textons dans le dictionnaire parcimonieux D. Le problème considère donc deux types d'inconnues que sont d'une part les textons du dictionnaire et d'autre part les coefficients du mélange.

Afin de pouvoir résoudre ce problème d'estimation sous-déterminée, des connaissances *a priori* doivent être introduites. Cela a conduit à l'émergence de différentes familles d'algorithmes de types bayésiens

ou puisant dans les méthodes d'optimisation déterministes selon des lois *a priori* de Cauchy, Jeffrey, gaussienne généralisée, laplacien, Student τ etc .... Les lois *a priori* les plus étudiées ces dernières années sont celles exploitant l'*a priori* de parcimonie. Cela a permis d'obtenir différents algorithmes. Ainsi des algorithmes travaillant sur l'équivalence L_1/L_0 ont été proposés [18]. Un critère de classification s'écrit, par exemple :

$$\underset{D,a}{arg\min} \sum_{x \in S} \| x - Da \|_2^2 + \lambda \sum_{i=1}^{|S|} \| a_i \|_{L_1} \qquad \text{s. c.} \forall j, \| d_j \|_2^2 = 1$$

Ce clustering par gestion de l'erreur est une suite d'approches alternées qui mettent à jour le dictionnaire pour un jeu de coefficients fixés puis estiment les coefficients pour un dictionnaire fixé. La première étape consiste à déterminer la meilleure combinaison de textons de D pour coder chaque échantillon. Une généralisation de l'algorithme des P-moyennes exploite une étape de codage parcimonieux utilisant par exemple soit la méthode FOCUSS (FOCal Undeterminated System Solution de Gorodnitsky [19]), soit la méthode OMP (Orthogonal Matching Pursuit) à dictionnaire fixé. En effet, l'algorithme des P-moyennes se fonde sur la règle d'association suivante de l'échantillon x au texton j du dictionnaire :

$$\| x - d_j \|_2^2 \leq \| x - d_k \|_2^2 \qquad \text{pour } j \neq k$$

où d_j (respectivement d_k) est le texton numéroté j (respectivement k) du dictionnaire parcimonieux D. Soit P_s le nombre de textons dans le dictionnaire parcimonieux D. Posons alors e_j un vecteur parcimonieux de $\mathbb{R}^{P_s}$ tel qu'il soit nul sauf pour le j-ième élément du vecteur qui vaut 1. Nous réécrivons

$$\| x - De_j \|_2^2 \leq \| x - De_k \|_2^2 \qquad \text{pour } j \neq k$$

ce qui nous ramène explicitement à la minimisation du critère suivant :

$$\| X - DA \|_F^2 \qquad \text{s.c. } a = e_k \text{ et } \| a \|_0 = 1$$

Pour rappel, la norme $\| a \|_0$ renvoie le nombre d'éléments non nul dans le vecteur a, la norme de Frobenius d'une matrice M est la racine carrée de la somme du carré des valeurs absolues de la matrice

$$\| X - DA \|_F^2 = \left(\sum_{i=1}^{|S|} \sum_{j=1}^{P_s} |x_{i,j} - d_j a_i|^2 \right)^{1/2}$$

L'approche d'estimation du dictionnaire présentée ici est le résultat de deux étapes combinées. La

deuxième étape, l'estimation du dictionnaire, peut être réalisé avec l'algorithme P-SVD [20–23] ou bien l'algorithme des directions optimales (MOD) [24].

Algorithme II.1 : Pseudo-code d'une étape de l'algorithme P-SVD

Données : Un ensemble d'échantillons S, le dictionnaire D, les codes A
Résultat : Le dictionnaire D, les codes A

pour chaque $j = 1, \ldots, P_s$; `/* $d_j$ est la colonne $j$ de $D$ */`
faire
  $W \leftarrow \{w = 1, \ldots, |S| \mid a_{w_j} \neq 0\}$; `/* Index des codes utilisant $d_j$ */`
  $R_W \leftarrow S_W - DA_W$; `/* Résidu de la décomposition */`
  % Mise à jour du texton d et coefficients α associés
  $d, \alpha \leftarrow \underset{\|d\|_2=1, \alpha \in \mathbb{R}^{|W|}}{arg\,min} \sum_{w \in W} \| r_w + a_{w,j} d_j - \alpha_j d \|_2^2$;
  $d_j \leftarrow d$; `/* enregistre le texton corrigé */`
  $A_{W,j} \leftarrow \alpha$; `/* remplace les codes calculés et non nul au départ */`
fin

Dans le cadre de l'algorithme P-SVD les valeurs non nulles des codes α ne sont pas fixées et serons mises à jour en même temps que les textons contenus dans D. Dans cette seconde étape, l'estimation des valeurs non nulles est moins coûteuse en termes de complexité calculatoire que dans la première étape et peux être réalisée en parallèle de l'estimation des textons. L'algorithme II.1 illustre ce que fait l'algorithme P-SVD, mais avec quelques notations spécifiques : soient d_j un texton qui est la j-ième colonne de D ; x_w un échantillon de S ; $a_{w,j}$ la valeur du code de x_w pour le texton d_j ; $A_W = \{a_w \mid w \in W\}$ (resp. $S_W = \{x_w \mid w \in W\}$) une sous-matrice de A (resp. S) ; $R_W = \{r_w \mid w \in W\}$ une matrice vérifiant par défaut que les autres colonnes soient nulles $r_l = 0$ pour tout $l \in ([1, |S|] \cap \mathbb{N}) \backslash W$.

Ainsi au moins trois points de vue existent sur l'estimation d'un dictionnaire pour réaliser la classification. Nous avons choisi de les présenter par ordre croissant du nombre de paramètres qui coïncide avec un taux décroissant de faux positifs. Supposons maintenant que le choix du dictionnaire est arrêté, la section suivante présente comment utiliser le dictionnaire pour traduire un patch quelconque en quantités comparables.

4 Méthodes de codage

Les approches de type SMV croisent les principes génératif et discriminatif. Le caractère discriminatif s'appuie sur un ré-encodage, obtenue d'un descripteur global, des images à partir d'un dictionnaire regroupant les mots visuels significatifs. Ainsi, selon que l'on exploite, comme vu au paragraphe précédent, une hypothèse de loi mélange ou celle d'une dépendance linéaire entre les textons et les descripteurs

observés, l'objectif est d'obtenir un dictionnaire représentant un modèle génératif.

Cette section décrit comment traduire une image I en un vecteur H nommé code du patch. Soit P le nombre de clusters parmi toutes les classes, le code du patch s'écrit $(h_i)_{i=1}^{P}$. Précisons que l'ensemble S des échantillons se répartit en plusieurs images I qui sont des sous-ensembles de S. Trois méthodes sont proposées correspondant aux choix effectués pour la construction du dictionnaire en termes de paramètres estimés pour chaque classe. Dans les deux premières sous-sections, seules les moyennes sont utilisées pour estimer la probabilité d'appartenance à une classe. Ensuite, une sous-section présente un assignement doux qui tiens en compte des classes voisines dans le choix. La troisième et dernière sous-section utilise les dépendances linéaires estimées dans le dictionnaire.

4.1 Exemple : codage

Reprenons l'exemple issus des travaux sur la méthode de Varma et al. [11] avec les patchs. La partie précédente de l'exemple présentait l'apprentissage du dictionnaire et cet exemple présente le codage des échantillons dans le dictionnaire choisi. Comme l'algorithme SMV peut se faire avec un dictionnaire *a priori*, l'apprentissage du dictionnaire n'appartient pas aux phases propres de l'algorithme.

La première phase du SMV - appelée également phase d'apprentissage - consiste à estimer la fréquence d'apparition de chaque texton contenu dans les images de la base de données. Les fréquences d'apparition des mots visuels sont représentées par des histogrammes (voir figure II.8). Un histogramme ainsi défini est nommé descripteur global de l'image. Une classe d'image sera représentée par la moyenne des descripteurs globaux des images de la classe.

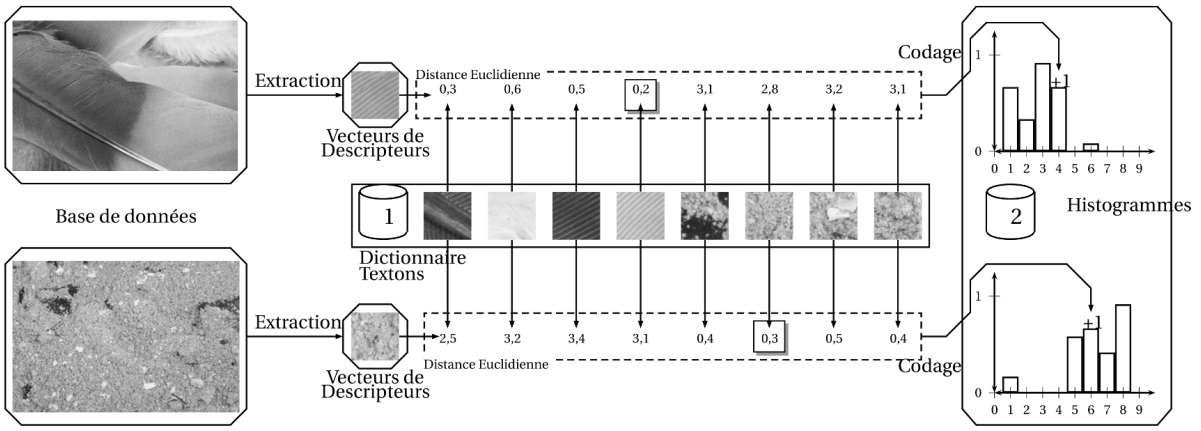


Figure II.8 – 2. Phase d'apprentissage. Calcul des fréquences d'apparition des textons dans l'image anisotrope et stochastique. Chaque patch est comparé séquentiellement à chaque texton au moyen d'une distance euclidienne. Le nombre de patchs le plus similaire au $k^{\text{ième}}$ texton représente la fréquence du $k^{\text{ième}}$ texton. Amsterdam Library of Textures (ALOT) [2]

Un histogramme, vu comme un court tableau de valeurs, est suffisant pour représenter une image texturée dans toute sa complexité et sa spécificité. Dans cet exemple basé sur des patchs, il est simple de lire les histogrammes. Pour l'image à la texture orientée, les plus fortes probabilités sont présentes sur les index des patchs estimés sur cette image. La section suivante présente une définition plus mathématique du codage par probabilité d'occurrence.

4.2 Probabilité d'occurrence

L'état de l'art sur le codage montre que la très grande majorité des méthodes de classification de type SMV que ce soit pour discriminer des images naturelles quelconques, ou des images de contenu texturé privilégient largement l'utilisation, pour le codage, d'une distribution multinomiale sur l'espace des descripteurs locaux.

Soient I une image, $x \in I$ un ensemble d'échantillons pour l'image et i un cluster. Supposons que lors de la construction du dictionnaire, un ensemble de P textons μ_p aient été estimées. Des lors, le code H de l'image I est construit au moyen de la formule suivante :

$$h_i = \frac{1}{|I|} \sum_{x \in I} \delta \left\{ \underset{0 \leq p \leq P-1}{arg\,\min} \parallel x - \mu_p \parallel_2^2 - i \right\}$$

où $|I|$ représente le nombre d'échantillons pour l'image I.

Comme le montrent Fernandez el al. dans leur article récent [1], il est intéressant de noter que concernant les textures, plusieurs propositions utilisant les descripteurs binaires ou ternaires se sont construites sur l'idée que h_k pouvait être spécifié *a priori* et non *a posteriori* évitant ainsi la phase coûteuse d'apprentissage. En effet, si nous considérons le cas des motifs binaires locaux (LBP, le dictionnaire est connu *a priori* puisqu'il est donné par les 2^K valeurs entières où K désigne le nombre de voisins utilisé pour construire le descripteur. La formule du code s'écrit alors :

$$h_i = \frac{1}{|I|} \sum_{x \in I} \delta \left\{ x - i \right\}$$

Les deux approches montrées ici sont qualifiées de « hard-clustering ». Elle signifie que tous les descripteurs locaux appartiennent à une unique classe (représentées par son texton μ_p). Et cette modélisation peut être affinée comme le présente la sous-section suivante.

4.3 Assignement doux

Coder une image inconnue consiste à la représenter dans l'espace des images dans lequel il y a plusieurs classes. Sans parler d'appartenance à une classe, un échantillon peut se situer à égale distance de plusieurs textons. Le codage doux ou soft-assignement consiste à prendre en compte, pour un même descripteur, plusieurs mots visuels. Il s'agit, par exemple, de l'implémentation suivante [25] :

$$h_i = \frac{1}{|I|} \sum_{x \in I} \frac{\exp\left\{-\beta \parallel x - \mu_i \parallel_2^2\right\}}{\sum_{\mu_q \in M_Q(x)} \exp\left\{-\beta \parallel x - \mu_q \parallel_2^2\right\}} \delta\{\mu_i \in M_Q(x)\}$$

où $M_Q(x)$ désigne les Q plus proches textons μ_p du descripteurs x et β un paramètre de lissage contrôlant le caractère doux de code. Sans les comparer outre-mesure, montrons maintenant le code associé aux dépendances linéaires.

4.4 Histogrammes issus des dépendances linéaires

Une autre famille de codeurs, dans l'esprit des approches exploitant une dépendance linéaire entre le dictionnaire et les descripteurs observés, considère un codage direct pour le codage des images. Il s'agit donc, par exemple, de calculer pour chaque descripteur le jeu de paramètres suivants :

$$a = \underset{\alpha}{arg\min} \parallel x - D\alpha \parallel_2^2 + \lambda \parallel \alpha \parallel_{l_1}$$

Un descripteur H peut alors être construit en sommant tous les jeux de coefficients normalisés d'une image I. Soit $\tilde{a}$ une fonction qui à un échantillon $x \in I$ associe la valeur normalisée du code $a/ \parallel a \parallel$, alors le descripteur H vérifie :

$$H = \frac{1}{|I|} \sum_{x \in I} \tilde{a}(x).$$

L'algorithme SMV est basé sur la comparaison d'images I au travers d'une représentation spécifique : le code H. Le code est indépendant de données telles que la taille de l'image, le nombre de canaux colorés pour n'être qu'un simple tableau de valeurs.

5 Conclusion

Suite à la description de toutes les images avec un code sur un dictionnaire de textons, un dictionnaire de descripteurs globaux est défini. Pour imager la suite du fonctionnement de l'algorithme SMV, reprenons

l'exemple des descripteurs patchs de Varma et al. [11]. La phase d'apprentissage terminée, chaque image dispose de son propre code.

Dans la phase de test, l'algorithme SMV prend en entrée une image à classifier. Puis, comme pour l'apprentissage du dictionnaire, les vecteurs de descripteurs de cette image sont extraits. A partir du dictionnaire estimé, l'image de test est codée en estimant la fréquence d'apparition de chaque texton (cf phase d'apprentissage). L'image de test est ainsi représentée par un descripteur global dit « de test » . L'image de test est associée à la classe la plus proche, c'est-à-dire la classe minimisant la distance entre le descripteur global de test et le descripteur global des classes d'apprentissage (voir figure II.9). En pratique une distance du χ^2 est utilisée.

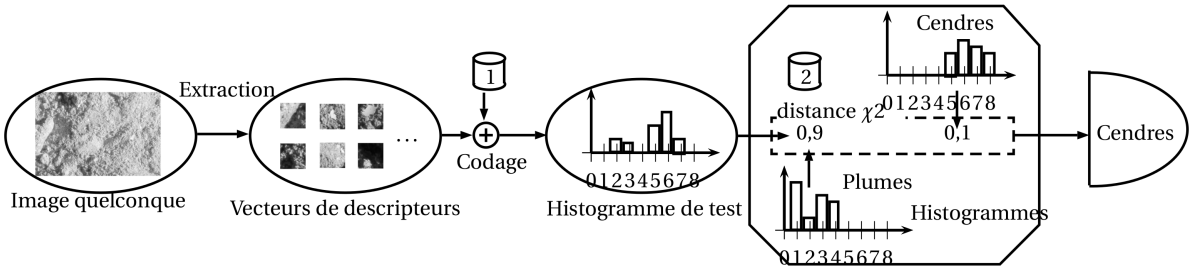


Figure II.9 – 3. Phase de test. Une image inconnue entre dans l'algorithme. Après extraction des vecteurs de descripteurs l'image est codée. L'histogramme de l'image codée est comparé aux histogrammes de l'image anisotrope et de l'image stochastique. ALOT

Le chapitre 1 a été l'occasion de présenter les méthodes utilisées dans le contexte des approches de type « sac de mots visuels ». Nous avons notamment montré la distinction entre approches à dictionnaire *a priori* et celles *a posteriori*. Dans ce qui suit, nous allons proposer un nouveau cadre d'étude et développer une méthodologie de type « Sac de mots visuels » dans le contexte de la modélisation stochastique. En effet, de nombreuses études récentes ont permis de mettre en avant l'intérêt, dans le contexte de la classification d'images texturées, de la modélisation stochastique des coefficients de sous-bandes de décomposition multi-échelle. Ce nouveau cadre d'étude ouvre de belles perspectives d'étude. Notamment, nous montrerons que cela nous conduit à nous intéresser à des problématiques de modélisation paramétrique permettant de prendre en compte la diversité intra-classe. Aussi dans un contexte paramétrique, nous proposerons un schéma innovant et complet fondé sur une architecture de type « Sac de mots visuels ».

Chapitre III

Définition et estimation d'un dictionnaire intrinsèque à l'espace des modèles stochastiques

Contenu du chapitre

1 Introduction

Les travaux de la communauté en analyse des images texturées vont de pair avec l'apparition de nouveaux outils de décomposition par trames telle que les ondelettes. Ces différents travaux exploitent tous le fait qu'il est possible de discriminer les contenus texturaux par le biais de la modélisation des lois empiriques des coefficients des échelles au moyen de distributions paramétriques unimodales. Ainsi, à partir de ce type d'approches, nous nous attacherons dans le présent chapitre à montrer comment nous pouvons exploiter le cadre de la modélisation stochastique dans le but de construire une approche fondée sur une méthodologie de type SMV. Dans un premier temps, nous allons rappeler le cadre de l'analyse par trames et les principales familles de lois proposées dans la « littérature ». Nous verrons ensuite comment déployer les notions de dictionnaires dans l'espace paramétrique nous conduisant à définir les notions de lois *a priori* sur variétés. La loi *a priori* permettant de prendre en compte la diversité intra-classe est un des éléments clés de l'approche. Elle permet, notamment, d'introduire la notion de composantes du dictionnaire à travers la définition des hyper paramètres d'une loi *a priori*. Nous verrons que cela nous conduit à définir l'équivalent de la loi mélange de gaussiennes dans l'espace euclidien au cas de la géométrie riemannienne. Nous terminerons ce chapitre par la présentation d'un algorithme d'estimation des hyper paramètres afin de pouvoir déployer l'approche SMV.

2 Rappels sur les approches paramétriques par trames de décomposition

2.1 Trames de décomposition

La figure III.1 présente une décomposition en ondelettes décimée d'une image. Cette décomposition est faite sur N_s échelles et N_o orientations. Les N coefficients de la décomposition $x = (x_n)_{n=1}^{N}$ en ondelettes sont alors séparés en sous-bandes de détail $(x_{s,o})_{s,o=1}^{s=N_s,o=N_o}$ et d'approximation x_a.

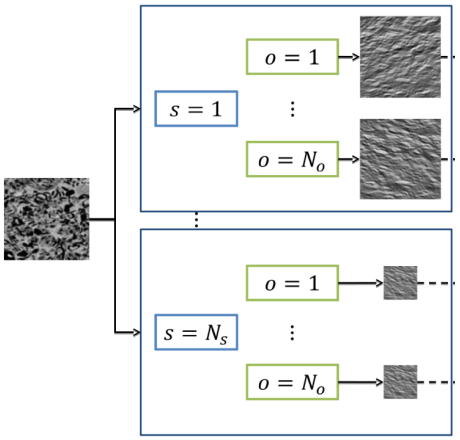


Figure III.1 – Schéma présentant à la fois l'image et les sous-bandes résultant d'une décomposition en ondelettes d'une image.

2.1.1 Définition

La communauté scientifique a construit plusieurs transformations de l'image offrant une invariance en échelle et orientation. Le qualificatif « trames par analyse » présente une transformation d'images qui contient plusieurs échelles et plusieurs orientations. La liste suivante présente des exemples de trames par analyse :

- la transformée en cosinus discrets [26]
- la transformée en sinus discrets [26]
- la transformée de Karhunen-Loève [26]
- les filtres de Gabor [27, 28]
- les bancs de filtres [29]
- les décompositions par pyramides
  - la pyramide Laplacienne [30]
  - Steerable Pyramids [31, 32]
- décomposition en ondelettes [33]
  - décimée [4]
  - non-décimée

2.1.2 Mesures de dissimilarité

Ce paragraphe montre que la dissimilarité calculée entre une image I et une image I' coïncide avec la dissimilarité entre les vecteurs de descripteurs θ et θ' estimés respectivement sur les images I et I'.

Autrement dit, l'application d'une trame par analyse ne modifie pas la dissimilarité existante entre deux vecteurs de descripteurs. Supposons que l'image I soit de dimension $L \times H$ (respectivement l'image I' est de dimension $L' \times H'$). Soit $V = (V_n)_{n=1}^{HL}$ une collection de variables aléatoires constituée de tous les niveaux de gris de l'image I (respectivement, $V' = (V'_n)_{n=1}^{H'L'}$ pour l'image I'). Soit T une décomposition en trames par analyse. Soit $x = (x_n)_{n=1}^{N}$ une collection de variables aléatoires définie comme le résultat de la transformation T de la collection V (respectivement $x' = (x'_n)_{n=1}^{N}$ pour V'). Soient t et t' les distributions empiriques des variables aléatoires V et V' respectivement. Sans perte de généralité, soient p et p' les distributions empiriques respectives de x et x'. Si m est la mesure de dissimilarité usuelle entre les distributions p et p', alors il existe une mesure de dissimilarité m_T vérifiant :

$$m_T(t \parallel t') = m(p(T(V)) \parallel p(T(V'))) = m(p \parallel p') \tag{III.1}$$

Considérons ici un exemple dans le plan $\mathbb{R}^2$ muni de sa base orthonormée usuelle $(\vec{e}_1, \vec{e}_2)$. T peut être, par exemple, une transformation linéaire dans le plan $\mathbb{R}^2$ vers le système de coordonnées $(\vec{u}_1, \vec{u}_2) = (M\vec{e}_1, M\vec{e}_2)$, avec M une matrice 2×2 de passage entre les deux espaces de coordonnées, supposée orthogonale $M^T M = I_2$. La distance euclidienne entre $a = a_1\vec{e}_1 + a_2\vec{e}_2$ et $b = b_1\vec{e}_1 + b_2\vec{e}_2$ correspond à une distance euclidienne entre Ma et Mb après changement de coordonnées :

$$m_{\text{euc}}(a,b) = \sqrt{a^T b} = \sqrt{a^T \left(M^{-1}M\right)^T M^{-1} Mb} = \sqrt{(Ma)^T \left(MM^T\right)^{-1} Mb} = \sqrt{(Ma)^T Mb} = m(Ma, Mb)$$

Le choix de la trame par analyse est associé à deux autres choix, premièrement le choix du modèle paramétrique utilisé pour modéliser p (voire utiliser directement les distributions empiriques) ; deuxièmement le choix de la mesure de dissimilarité, contraint par le modèle stochastique de p qui peut être étendu au cas non euclidien.

2.2 Propriétés statistiques des décompositions

L'utilisation des trames par analyse pour la classification d'images texturées est justifiée [34] par, d'une part, les propriétés statistiques des champs texturaux et, d'autre part, les développements réalisés en perception visuelle. En recherche d'images par contenu texturé [4], les trames par analyse permettent l'obtention de performances acceptables sur de grandes bases de données. Elle trouvent, de plus, un fondement dans les études physiologiques du cortex visuel [35–38]. L'utilisation de trames par analyse peut être vue [26] comme une tentative de modélisation du système visuel humain. [39–41]. L'autre raison

de l'utilisation des trames par analyse réside dans les propriétés statistiques des sous-bandes.

2.2.1 Optimalité

Soit I une image de dimension $L \times H$. Soit K le nombre de pixels dans le voisinage du pixel défini par son niveau de gris V_n, le vecteur $\vec{V}_n$ est composé de tous les niveaux de gris dans le voisinage, autrement dit, le vecteur $\vec{V}_n$ est de dimension K, pour $n = 1, \ldots, HL$. Le $k^{\text{ième}}$ élément du vecteur $\vec{V}_n$ est noté $V_{n,k}$, pour $k = 1, \ldots, K$. Soit $\vec{V} = (\vec{V}_n)_{n=1}^N$ une collection de vecteurs aléatoires indépendants et identiquement distribués (iid). Soit C_V la matrice de covariance du vecteur aléatoire $\vec{V}_n$. Soit $\vec{x} = (\vec{x}_n)_{n=1}^N$ une collection de vecteurs aléatoires iid sur l'espace transformé $T(V)$. Soit un voisinage de K coefficients autour de x_n, le vecteur $\vec{x}_n$ est composé de toutes les valeurs des coefficients dans le voisinage, autrement dit, le vecteur $\vec{x}_n$ est de dimension K, pour $n = 1, \ldots, N$. Le $k^{\text{ième}}$ élément du vecteur $\vec{x}_n$ est noté $x_{n,k}$, pour $k = 1, \ldots, K$. Unser [26] considère deux aspects différents de l'optimalité dans le contexte de l'analyse de textures :

Le critère d'entropie. Choisir une trame par analyse qui produit des statistiques d'ordre 1 pour les vecteurs aléatoires $\vec{x}_{s,o}$ les plus « différentes » possible. La solution, qui présente la plus grande différence en variance des distributions, est celle qui minimise le critère d'entropie [42] suivant :

$$H(T) = -\sum_{n=1}^{N} \frac{\vec{x}_n^T C_V \vec{x}_n}{Tr(C_V)} \log \left\{ \frac{\vec{x}_n^T C_V \vec{x}_n}{Tr(C_V)} \right\}$$

où $\vec{x}_n^T$ est la transposée du vecteur $\vec{x}_n$, $Tr(C_V)$ la trace de la matrice C_V.

Le critère d'énergie. Choisir une trame par analyse produisant des variables non-corrélées. Condition remplie lorsque le critère d'énergie suivant est maximal et égal à 1 [43] :

$$E(T) = -\sum_{n=1}^{N} \frac{\left(\vec{x}_n^T C_V \vec{x}_n\right)^2}{\|C_V\|_{\text{HS}}^2}$$

avec $\|\cdot\|_{\text{HS}}^2$ la norme de Hilbert-Schmidt d'une matrice, ayant comme propriété d'être invariante à toute transformation de similarité [26]. La non corrélation est une condition nécessaire mais pas forcément suffisante pour l'indépendance des variables. Autrement dit la distribution $p(\vec{x}) = \prod_{k=1}^{K} p_k(x_k)$ du vecteur aléatoire $\vec{x}$ s'écrit comme le produit des distributions p_k des variables aléatoires composées des marginales $x_k = (x_{n,k})_{n=1}^N$.

En résumé, les trames par analyse sont choisies pour un critère psycho-visuel et pour les propriétés

statistiques obtenues dans la distribution des coefficients d'ondelettes. De plus, une modélisation de la distribution de coefficients de trames par analyse permettrait d'atteindre un compromis entre complexité calculatoire et approche invariante par échelle.

2.2.2 Indépendance

Une image I est décomposée en plusieurs sous-bandes de décomposition. Généralement les sous-bandes de décomposition sont définies pour une échelle $s = 1, \ldots, N_s$ et une orientation $o = 1, \ldots, N_o$. Soit $x_{s,o} = (x_{s,o,n})_{n=1}^{N_{s,o}}$ la collection de variables aléatoires telle que $x_{s,o,n}$ soit un coefficient de sous-bande de décomposition d'échelle s et d'orientation o. Pour modéliser de manière globale une texture I définie par une échelle $s = 1, \ldots, N_s$ et une orientation $o = 1, \ldots, N_o$, l'approche proposée est de modéliser la distribution de $x_{s,o}$. Deux hypothèses sont largement utilisées dans la communauté, il s'agit de :

indépendance intra-bande. la variable aléatoire $x_{s,o,n}$ est supposée indépendante et identiquement distribuée ;

$$p(x_{s,o}) \simeq p(x_{s,o,n}, n = 1, \ldots, N_{s,o}) = \prod_{n=1}^{N_{s,o}} p(x_{s,o,n})$$

indépendance inter-bandes. la variable aléatoire $x_{s,o}$ est indépendante de $x_{s',o'}$ tant que $(s; o) \neq (s'; o')$.

$$p(x) \simeq p(x_a, x_{s,o}, s = 1, \ldots, N_s, o = 1, \ldots, N_o) = p(x_a) \prod_{s=1}^{N_s} \prod_{o=1}^{N_o} p(x_{s,o})$$

Les deux hypothèses d'indépendance permettent d'écrire une formule de la densité de probabilité pour la variable aléatoire x :

$$p(x) \simeq p(x_a) \prod_{s=1}^{N_s} \prod_{o=1}^{N_o} \prod_{n=1}^{N_{s,o}} p(x_{s,o,n}).$$

La distribution des coefficients de trames par analyse est supposée probabiliste par l'équation précédente, et cela même pour des textures dites directionnelles. L'hypothèse d'indépendance est très forte mais il est possible de définir une distribution paramétrique pour chaque sous-bande de détail $x_{s,o}$ indépendamment des autres. A noter que plusieurs auteurs n'utilisent pas la sous-bande d'approximation x_a [4, 44–46]. Allili propose d'utiliser les mélanges de gaussiennes généralisées [47] pour la modélisation de la sous-bande d'approximation, ce qui amène un gain en performances. Dans la suite de ce document, la distribution des coefficients d'approximation est ignorée, menant à la distribution :

$$p(x) \simeq \prod_{s=1}^{N_s} \prod_{o=1}^{N_o} \prod_{n=1}^{N_{s,o}} p(x_{s,o,n}).$$

Nous parlons de modélisation univariée d'images texturées lorsque les deux hypothèses d'indépendance sont vérifiées : indépendance inter-bandes et indépendance intra-bande. Les hypothèses d'indépendances ne prennent en compte aucune dépendance, néanmoins les informations conservées restent discriminantes [4, 46, 48] pour réaliser la classification d'images texturées.

Ce paragraphe a présenté l'état de l'art sur les statistiques des trames par analyse. Continuons avec un paragraphe présentant des densités de probabilités déduites des statistiques.

2.3 Modélisation paramétrique des décompositions

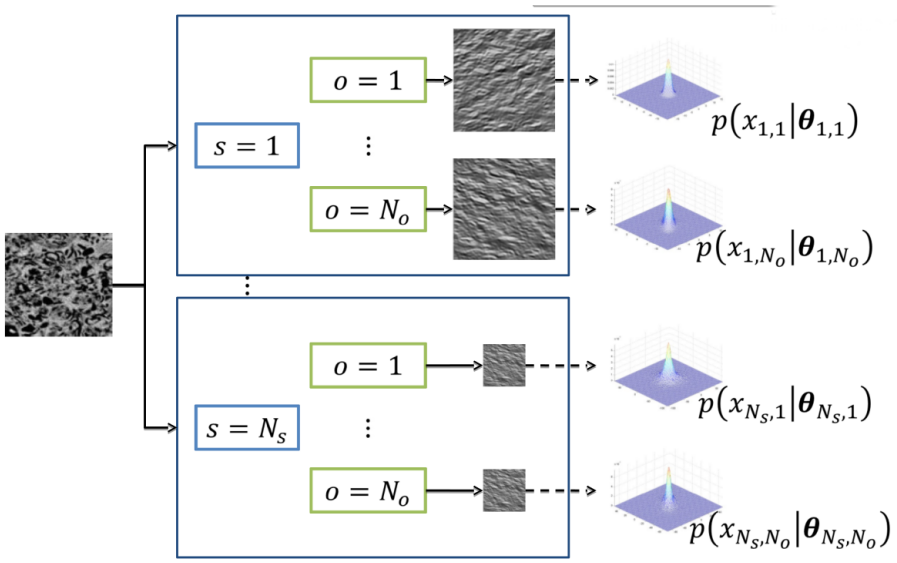


Figure III.2 – Schéma explicitant la décomposition en ondelettes d'une image

La figure III.2 présente une décomposition en ondelettes décimées d'une image. Cette décomposition est faite sur N_s échelles et N_o orientations. Les N coefficients de la décomposition $x = (x_n)_{n=1}^{N}$ en ondelettes sont alors séparés en sous-bandes de détails $(x_{s,o})_{s,o=1}^{s=N_s,o=N_o}$ et d'approximation x_a. Une décomposition en ondelettes est composée d'une sous-bande de détails par échelle et par orientation en plus d'une sous-bande d'approximation x_a. Les coefficients d'une sous-bande de détails $x_{s,o}$, pour $s = 1, \ldots, N_s$ et $o = 1, \ldots, N_o$ sont supposés être des réalisations indépendantes du modèle paramétrique $\mathcal{P}$. Soit $N_{s,o}$ le nombre de coefficients d'ondelettes présents à la sous-bande d'échelle s et d'orientation o, alors $x_{s,o}$ représente une collection de variables aléatoires iid $x_{s,o} = (x_{s,o,n})_{n=1}^{N_{s,o}}$. De même, soit N_a le nombre de coefficients d'ondelettes présents dans la sous-bande d'approximation, alors $x_a = (x_{a,n})_{n=1}^{N_a}$ représente une variable aléatoire. Les définitions de $N_{s,o}$ et N_a permettent de retrouver le nombre de coefficients d'ondelette $N = N_a + \sum_{s=1}^{N_s} \sum_{o=1}^{N_o} N_{s,o}$. Bien que présente, la distribution de la sous-bande x_a est laissée

de côté sans impacter les performances de classification [4, 44–46]. En revenant à la distribution paramétrique $\mathcal{P}$, la densité de probabilité est exprimée respectivement avec la lettre p et le vecteur paramétrique associé par θ. Un vecteur paramétrique $\theta_{s,o}$ est estimé au sens du maximum de vraisemblance sur chaque sous-bande $x_{s,o} = (x_{s,o,n})_{n=1}^{N_{s,o}}$. Une image ainsi décomposée est représentée par une collection de vecteurs paramétriques $\theta = (\theta_{s,o})_{s,o=1}^{s=N_s,o=N_o}$.

2.3.1 Définition

Le vecteur de descripteurs locaux issus des trames par analyse est nommé vecteur paramétrique. Il est défini mathématiquement comme l'ensemble des vecteurs paramétriques estimés sur chaque sous-bande séparément. La distribution des coefficients $x_{s,o}$ est approchée par un modèle paramétrique $\mathcal{P}$. Soit $\hat{\theta}_{s,o}$ le vecteur paramétrique estimé au sens du maximum de vraisemblance sur les réalisations de $x_{s,o}$. Après avoir estimé le vecteur $\hat{\theta}_{s,o}$ pour tout $s = 1, \ldots, N_s$ et tout $o = 1, \ldots, N_o$, une image est représentée par le vecteur combiné $\theta = (\hat{\theta}_{s,o})_{s,o=1}^{s=N_s,o=N_o}$. Le principal avantage de la modélisation paramétrique reste la réduction de la dimension des vecteurs de descripteurs locaux par rapport à l'utilisation de distributions empiriques pour chaque sous-bande [49].

La supposition d'indépendance inter-bandes et intra-bande est utilisée conjointement à un modèle paramétrique univarié. Parmi l'ensemble des modèles univariés voici quelques exemples comme la gaussienne généralisée (GGD) [4, 33], le mélange de gaussiennes généralisées [47], la Gamma généralisée ($G\Gamma D$) [46] ou encore la gaussienne généralisée asymétrique (aGGD) [48]. Les hypothèses d'indépendances peuvent ensuite être rendues moins contraignantes, cela implique l'utilisation d'un modèle paramétrique multivarié. Trois exemples de dépendances sont présentées dans les lignes suivantes. Les modèles de Markov cachés (WD-HMM) sont utilisés pour modéliser les dépendances inter-bandes [50].

Pour les deux exemples suivants, plus d'un modèle multivarié est cité. Pour modéliser les dépendances couleurs [51–53], la distribution à copule Student [54] ou la gaussienne généralisée multivariée (MGGD) [51] ont été utilisées. Le dernier exemple porte sur la modélisation des dépendances intra-bande, les modèles paramétriques comme la distribution gaussienne multivariée (MG) [26], la distribution à copule gaussienne [45], ou le Spherically Invariant Random Vector (SIRV) [55, 56] ont été proposés. Sans plus de justifications ou références, tous les modèles multivariés peuvent être utilisés pour la modélisation des dépendances intra-bande, de même pour la dépendance couleur ou la dépendance inter-bandes.

2.3.2 Mesures de dissimilarité

Ce mémoire s'intéresse aux mesures de dissimilarité entre des distributions. Il est important (voir équation (III.1) page 28) que la mesure de dissimilarité entre les distributions des niveaux de gris d'une image soit cohérente avec la mesure de dissimilarité entre les distributions des coefficients de trames par analyse. La distribution des coefficients est supposée paramétrique, sous couvert du respect de certaines hypothèses d'indépendance. L'égalité (III.1) (voir page 28) entre les mesures de dissimilarité associée à l'hypothèse d'un modèle paramétrique fait que la mesure est égale quelque soit le choix fait pour la paramétrisation. Il est alors question d'invariance à la paramétrisation de la mesure de dissimilarité.

Une mesure de dissimilarité entre des distributions est un des éléments de la géométrie de l'espace des distributions. La géométrie de l'information est un domaine scientifique issue de la théorie de l'information d'une part et de l'analyse géométrique d'autre part. La géométrie de l'information fournit des mesures de dissimilarité basées directement sur les modèles stochastiques comme la distance riemannienne, la divergence de Kullback-Leibler et la divergence de Jeffrey.

Voici la définition de la divergence de Kullback-Leibler [4, 44, 46, 50],

$$\mathrm{KLD}(p(x|\theta) \parallel p(x|\theta')) = \int_U p(x|\theta) log \left\{ \frac{p(x|\theta)}{p(x|\theta')} \right\} .dx.$$

Comme la divergence de Kullback-Leibler n'est pas symétrique, les développements plus récents utilisent une divergence de Jeffrey [45, 48, 56] donnée par

$$\mathrm{JD}(p(x|\theta), p(x|\theta')) = \mathrm{KLD}(p(x|\theta) \parallel p(x|\theta')) + \mathrm{KLD}(p(x|\theta') \parallel p(x|\theta))$$

qui est une « symétrisation » de la divergence de Kullback-Leibler. Les notations seront les suivantes :

- x la variable aléatoire formée par les coefficients de trames par analyse ;
- p représente la densité de probabilité de la distribution paramétrique ;
- θ est le vecteur paramétrique de la distribution paramétrique ;
- θ' est un vecteur paramétrique de référence ou texton ;
- U est le support de la densité de probabilité de la distribution paramétrique.

La divergence de Kullback-Leibler n'est pas symétrique ce qui est explicité dans la formule par la séparation des deux entrées avec le symbole $\parallel$. Notons également que pour un modèle paramétrique au choix parmi GGD, GΓD, aGGD, la divergence de Kullback-Leibler entre deux distributions est explicite et ne s'exprime qu'à partir des vecteurs paramétriques θ et θ'. La divergence de Jeffrey hérite de la divergence

de Kullback-Leibler une formulation explicite pour ces mêmes modèles.

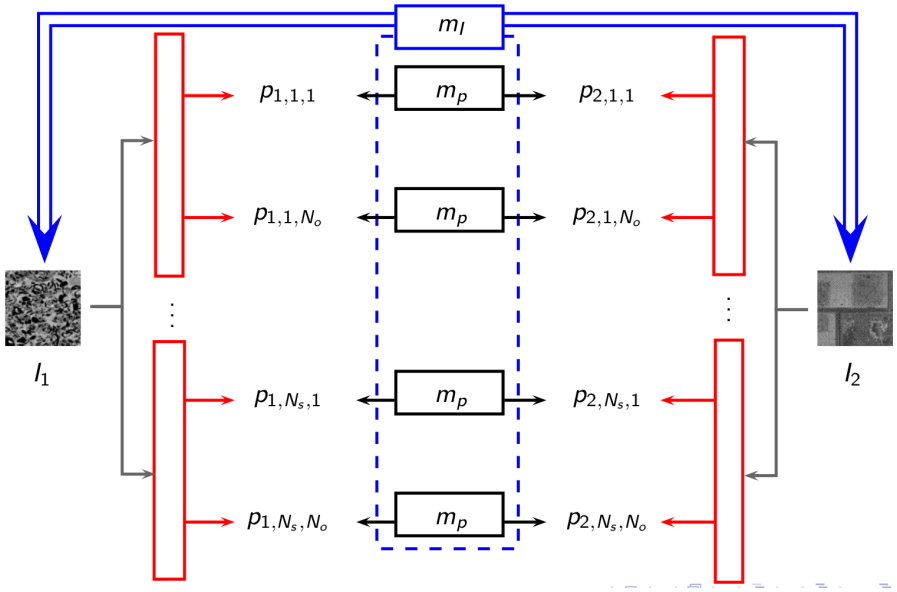


Figure III.3 – Par indépendance de la distribution des coefficients et séparabilité de la mesure de dissimilarité, la mesure de dissimilarité m_I entre deux images correspond à la somme des mesures de dissimilarité m_p sur chaque sous-bande.

La figure III.3 présente deux images I_1 et I_2 qui ont été décomposées en ondelettes. Les coefficients d'ondelettes de l'image I_i, pour tout $i = 1, 2$, sont représentés, sur chaque sous-bande, par la variable aléatoire $x_{i,s,o}$ indépendante et identiquement distribuée qui possède la densité de probabilité $p_{i,s,o}$. Considérons maintenant deux hypothèses d'indépendance : l'indépendance inter-bandes et l'indépendance intra-bande. Dès lors, la densité de probabilité p_i de l'image I_i n'est que le produit des densités de probabilité de chaque sous-bande :

$$p_i(x_{i,a}, (x_{i,s,o})_{s,o=1}^{s=N_s,o=N_o}) = p_{i,a}(x_{i,a}) \prod_{s=1}^{N_s} \prod_{o=1}^{N_o} p_{i,s,o}(x_{i,s,o})$$

La mesure de dissimilarité m_I entre les deux images I_1 et I_2 est associée à une mesure entre les distributions des coefficients d'ondelettes (propriété des trames par analyse) :

$$m_I(I_1 \parallel I_2) \simeq m_p(p_1 \parallel p_2)$$

nous appliquons alors l'hypothèse d'indépendance inter-bandes :

$$m_I(I_1 \parallel I_2) \simeq m_p \left(p_{1,a}(x_{1,a}) \prod_{s=1}^{N_s} \prod_{o=1}^{N_o} p_{1,s,o} \parallel p_{2,a}(x_{2,a}) \prod_{s=1}^{N_s} \prod_{o=1}^{N_o} p_{2,s,o} \right)$$

sans prendre en compte la distribution des coefficients d'approximation x_a

$$m_I(I_1 \parallel I_2) \simeq m_p \left(\prod_{s=1}^{N_s} \prod_{o=1}^{N_o} p_{1,s,o} \parallel \prod_{s=1}^{N_s} \prod_{o=1}^{N_o} p_{2,s,o} \right)$$

supposons maintenant que la mesure de dissimilarité m_p est « séparable » (comme la distance euclidienne, la divergence de Kullback-Leibler ou la divergence de Jeffrey). m_p correspond à la somme des mesures de dissimilarité sur chaque sous-bande :

$$m_I(I_1 \parallel I_2) \simeq \sum_{s=1}^{N_s} \sum_{o=1}^{N_o} m_p(p_{1,s,o} \parallel p_{2,s,o}) \tag{III.2}$$

L'inéquation III.2 présente le symbole $\simeq$, ce symbole représente une égalité approchée de la véritable mesure de dissimilarité $m_I(I_1 \parallel I_2)$. L'erreur $||m_I(I_1 \parallel I_2) - \sum_{s=1}^{N_s} \sum_{o=1}^{N_o} m_p(p_{1,s,o} \parallel p_{2,s,o})||^2$ n'admet pas de formule explicite car $m_I(I_1 \parallel I_2)$ n'en possède pas. Il est donc difficile d'évaluer l'erreur de cette approximation. Néanmoins les bonnes performances [4, 44, 46, 57] des algorithmes de classifications basés sur le modèle paramétrique indique que l'erreur est négligeable.

Avec ce paragraphe se termine les généralités autour des densités de probabilité. La suite de cette section porte sur une étude plus fine de trois distributions paramétriques utilisées dans ce mémoire : la GGD, la GΓD et le modèle SIRV.

2.4 Distribution gaussienne généralisée

Do et Vetterli [4] proposent d'utiliser une décomposition en ondelettes orthogonales sur les images. Soit $x_{s,o}$ une variable aléatoire correspondant aux coefficients d'ondelettes de la sous-bande d'échelle $s = 1, \ldots, N_s$ et d'orientation $o = 1, \ldots, N_o$. La décomposition en ondelettes utilise $N_o = 3$ orientations et $N_s = 2$ ou 3 échelles de décomposition.

2.4.1 Densité de probabilité

La gaussienne généralisée (GGD) est définie par un paramètre d'échelle α strictement positif et un paramètre de forme β strictement positif. La densité de probabilité explicite d'une GGD s'écrit :

$$p(x|\alpha, \beta) = \frac{1}{2\alpha\Gamma(1/\beta + 1)} \exp\left\{ - \left(\frac{|x|}{\alpha} \right)^{\beta} \right\} \tag{III.3}$$

avec Γ la fonction Gamma définie par la formule $\Gamma(z) = \int_{\mathbb{R}^+} t^{z-1}e^{-t}.dt$. La moyenne des sous-bandes de détails est nulle. Nous ne considérons donc pas de paramètres de localisation.

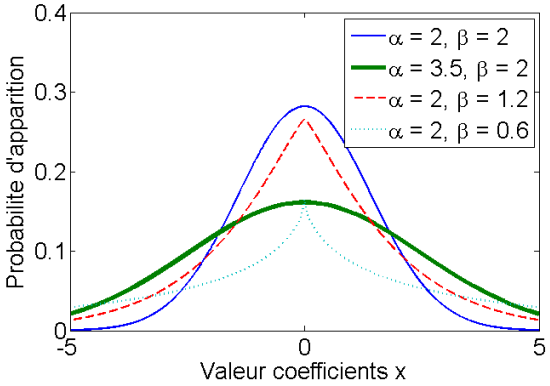


Figure III.4 – Densité de probabilité de la distribution gaussienne généralisée centrée pour les couples de paramètres (α, β) égaux à : $(2; 2)$ pour le trait fin plein ; $(3, 5; 2)$ pour le trait gras plein ; $(2; 1, 2)$ pour le trait discontinu ; $(2; 0, 6)$ pour le trait en pointillé

La figure III.4 représente la densité de probabilité de la GGD pour quatre jeux de paramètres. Lorsque le paramètre de forme β est égal à 2 la GGD coïncide avec une distribution gaussienne, ce qui est représenté avec les deux courbes en trait plein, la courbe en trait gras plein correspond à un paramètre d'échelle $\alpha = 3, 5$ alors que la courbe en trait fin plein correspond au paramètre d'échelle $\alpha = 2$. La courbe en trait discontinu correspond à un couple de paramètres $(\alpha, \beta) = (2; 1.2)$, la courbe en pointillés correspond à un couple de paramètres $(\alpha, \beta) = (2; 0.6)$. Le paramètre de forme β modifie la forme de la densité de probabilité afin d'obtenir des queues plus lourdes et une distribution plus pointue en l'origine pour une valeur décroissante du paramètre de forme β. Inversement, la GGD tend vers une distribution uniforme lorsque le paramètre de forme β tend vers l'infini.

2.4.2 Estimation des paramètres

Les estimations au sens du maximum de vraisemblance des deux paramètres α et β ne sont pas faites conjointement. Soient N réalisations de la variable aléatoire $x = (x_n)_{n=1}^N$ indépendantes et identiquement distribuées sachant que x suit une GGD. Tout d'abord, $\hat{\beta}$ est solution de :

$$1 + \frac{1}{\hat{\beta}}\Psi\left(\frac{1}{\hat{\beta}}\right) - \frac{\sum_{n=1}^N |x_n|^{\hat{\beta}} \log\{|x_n|\}}{\sum_{n=1}^N |x_n|^{\hat{\beta}}} + \frac{1}{\hat{\beta}} \log\left\{\frac{\hat{\beta}}{N}\sum_{n=1}^N |x_n|^{\hat{\beta}}\right\} = 0 \qquad \text{(III.4)}$$

avec Ψ la fonction Digamma définie comme $\Psi(z) = \frac{\partial \log\{\Gamma(z)\}}{\partial z} = \frac{\Gamma'(z)}{\Gamma(z)}$. La méthode de Newton-Raphson est utilisée pour estimer la valeur de β. L'algorithme III.1 montre la mise à jour de la suite $(\beta_k)_{k=1}^{\infty}$ qui

converge vers le paramètre de forme $\hat{\beta}$. Maintenant l'estimateur au sens du maximum de vraisemblance du paramètre d'échelle α qui dépend de $\hat{\beta}$ s'écrit :

$$\hat{\alpha} = \left(\frac{\hat{\beta}}{N} \sum_{n=1}^{N} |x_n|^{\hat{\beta}} \right)^{1/\hat{\beta}} .$$

Algorithme III.1 : Pseudo-code de l'algorithme d'estimation des paramètres de GGD [4]

Données : Un ensemble de N réalisations $(x_n)_{n=1}^{N}$
Résultat : Les estimés au sens du maximum de vraisemblance des paramètres $\hat{\alpha}$, $\hat{\beta}$
Initialisation de β avec un estimateur de moments [4]
$f(\beta) \leftarrow 1 + \frac{1}{\hat{\beta}} \Psi\left(\frac{1}{\beta}\right) - \frac{\sum_{n=1}^{N} |x_n|^{\beta} \log\{|x_n|\}}{\sum_{n=1}^{N} |x_n|^{\beta}} + \frac{1}{\beta} \log\left\{ \frac{\beta}{N} \sum_{n=1}^{N} |x_n|^{\beta} \right\}$;
pour chaque *k prenant des valeurs de 1 à* N_{iter} **faire**
  $\beta \leftarrow \beta - f(\beta)/f'(\beta)$; `/* Mise à jour */`
fin
$\hat{\beta} \leftarrow \beta$; `/* Estimé de` β `*/`
$\hat{\alpha} \leftarrow \left(\frac{\hat{\beta}}{N} \sum_{n=1}^{N} |x_n|^{\hat{\beta}} \right)^{1/\hat{\beta}}$; `/* Estimé de` α `*/`

2.4.3 Existence et unicité

L'équation III.4 montre que l'estimateur au sens du maximum de vraisemblance du paramètre $\hat{\beta}$ est indépendant du paramètre d'échelle α de la GGD. Les démonstrations d'existence et d'unicité de cet estimateur peuvent se faire quelque soit la valeur de α. Varanasi [58] indique que l'équation III.4 admet une unique racine en probabilité. De plus, la racine du gradient de la vraisemblance en β de la GGD existe et est unique [59–61]. L'estimateur au sens du maximum de vraisemblance du paramètre de forme $\hat{\beta}$ de la GGD est asymptotiquement optimal.

Soit $\hat{\beta}$ un paramètre de forme pour la GGD, alors Varanasi et Aazhang [61] montrent que la variance σ^2 existe dans $\mathbb{R}^+$ et est unique. Par définition de la variance σ^2, nous avons

$$\sigma^2 = \int_{\mathbb{R}} x^2 p(x|\alpha,\beta).dx - \left(\int_{\mathbb{R}} x p(x|\alpha,\beta).dx \right)^2$$

Appliquons la propriété de la densité de probabilité qu'est la parité de cette fonction

$$\sigma^2 = 2 \int_{\mathbb{R}^+} x^2 p(x|\alpha,\beta).dx = \frac{2}{2\alpha\Gamma(1/\beta+1)} \int_{\mathbb{R}} x^2 \exp\left\{ -\left(\frac{x}{\alpha}\right)^{\beta} \right\}.dx$$

Ce qui donne, après changement de variables $t = x^\beta/\alpha^\beta$ et simplification à

$$\sigma^2 = \alpha^2 \frac{\Gamma(2/\beta)}{\Gamma(1/\beta)}. \tag{III.5}$$

Pour une valeur de paramètre de forme β fixé, l'estimateur de la variance existe et est unique. Par l'équation III.5, l'estimateur au sens du maximum de vraisemblance du paramètre d'échelle $\hat{\alpha}$ de la GGD existe et est unique.

2.4.4 Mesures de dissimilarité

Avec la définition de la densité de probabilité de la GGD, il est possible d'expliciter la formule [4] de la divergence de Kullback-Leibler (KLD) entre deux distributions GGD ayant comme vecteurs paramétriques respectifs $\theta = (\alpha, \beta)$ et $\theta' = (\alpha', \beta')$:

$$\mathrm{KLD}(p(x|\theta) \parallel p(x|\theta')) = \log\left\{\frac{\alpha'\Gamma(1/\beta'+1)}{\alpha\Gamma(1/\beta+1)}\right\} + \left(\frac{\alpha}{\alpha'}\right)^{\beta'} \frac{\Gamma((\beta'+1)/\beta)}{\Gamma(1/\beta)} - \frac{1}{\beta}. \tag{III.6}$$

La définition de la divergence de Jeffrey permet d'obtenir une formule explicite à partir de la formule (III.6) [48] :

$$\mathrm{JD}(p(x|\theta), p(x|\theta')) = \left(\frac{\alpha}{\alpha'}\right)^{\beta'} \frac{\Gamma((\beta'+1)/\beta)}{\Gamma(1/\beta)} + \left(\frac{\alpha'}{\alpha}\right)^{\beta} \frac{\Gamma((\beta+1)/\beta')}{\Gamma(1/\beta')} - \frac{1}{\beta} - \frac{1}{\beta'}.$$

Nous avons utilisé la GGD à plusieurs reprises, surtout à titre de comparaison [62–67]

2.5 Distribution Gamma généralisée

Choy et Tong [46] proposent d'approcher la distribution des coefficients d'ondelettes $x_{s,o}$ avec une distribution Gamma généralisée (GΓD). Encore une fois, les coefficients en ondelettes $x_{s,o}$ sont supposés indépendants des coefficients en ondelettes provenant de sous-bandes différentes.

2.5.1 Densité de probabilité

La GΓD dépend de trois paramètres : le paramètre d'échelle α, le paramètre de puissance β et le paramètre de forme λ tous strictement positifs. La densité de probabilité de la variable aléatoire x s'exprime :

$$p(x|\alpha, \beta, \lambda) = \frac{\beta|x|^{\beta\lambda-1}}{2\alpha^{\beta\lambda}\Gamma(\lambda)} \exp\left\{-\left(\frac{|x|}{\alpha}\right)^{\beta}\right\}$$

avec Γ la fonction Gamma définie par la formule $\Gamma(z) = \int_{\mathbb{R}^+} t^{z-1}e^{-t}.dt$.

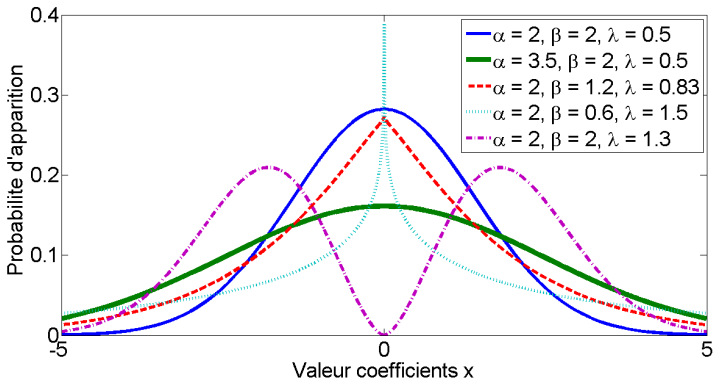


Figure III.5 – Densité de probabilité de la distribution Gamma généralisée centrée pour les triplés de paramètres (α, β, λ) égaux à : $(2; 2; 0,5)$ pour le trait fin plein ; $(3,5; 2; 0,5)$ pour le trait gras plein ; $(2; 1,2; 0,83)$ pour le trait discontinu ; $(2; 0,6; 1,6)$ pour le trait en pointillé ; $(2; 2; 1,3)$ pour le trait et points

La figure III.5 représente la densité de probabilité de la GΓD pour quatre jeux de paramètres. Lorsque les paramètres de puissance $\beta = 2$ et de forme $\lambda = 1/\beta$, la GΓD coïncide avec une distribution gaussienne, ce qui est représenté avec les deux courbes en trait plein, la courbe en trait gras plein correspond à un paramètre d'échelle $\alpha = 3,5$ alors que la courbe en trait fin plein correspond au paramètre d'échelle $\alpha = 2$. La courbe en trait discontinu correspond à un triplé de paramètres $(\alpha, \beta, \lambda) = (2; 1.2; 0.83)$, la courbe en pointillés correspond à un triplé de paramètres $(\alpha, \beta, \lambda) = (2; 0,6; 1,6)$. Enfin, la courbe en trait discontinu avec des points correspond à un triplé de paramètres $(\alpha, \beta, \lambda) = (2; 2; 1,3)$. Le paramètre de puissance β modifie la forme de la densité de probabilité afin d'obtenir des queues plus lourdes et une distribution plus pointue en l'origine pour une valeur décroissante du paramètre de puissance β. Inversement, la GΓD tend vers une distribution uniforme lorsque le paramètre de puissance β tend vers l'infini. Lorsque le paramètre de forme λ dépasse largement la valeur seuil $1/\beta$, la distribution obtenue gagne un deuxième mode qui est positionné de manière symétrique par rapport à l'axe des ordonnées $x = 0$.

2.5.2 Estimation des paramètres

Song [68] propose d'utiliser la méthode d'estimation de la forme indépendante de l'échelle (SISE). L'estimation des paramètres est effectuée en deux temps, d'abord l'estimation du paramètre de puissance β est effectuée indépendamment des paramètres d'échelle α et de forme λ. Soit N le nombre de réalisations

de la variable aléatoire $x = (x_n)_{n=1}^N$. Le paramètre de puissance $\hat{\beta}$ vérifie le système d'équations :

$$\mathcal{S}_N(\beta) := \log\{\mathcal{M}_N(2\beta)\} - 2\log\{\mathcal{M}_N(\beta)\} - \log\{1 + \beta(\mathcal{R}'_N(\beta) - \mathcal{R}'_N(0))\} = 0 \quad \text{(III.7)}$$

$$\mathcal{T}_N(\beta) := \frac{\mathcal{M}_N(2\beta)}{\mathcal{M}_N^2(\beta)} - (1 + \beta(\mathcal{R}'_N(\beta) - \mathcal{R}'_N(0))) = 0 \quad \text{(III.8)}$$

avec $\mathcal{M}_N = \frac{1}{N}\sum_{n=1}^N |x_n|^t$, $\mathcal{M}'_N = \frac{1}{N}\sum_{n=1}^N |x_n|^t \log\{|x_n|\}$ et $\mathcal{R}'_N = \mathcal{M}'_N(t)/\mathcal{M}_N(t)$ des cartographies définies sur un intervalle $(\beta_0 - \delta, \infty)$ avec δ un réel strictement positif et β_0 la véritable valeur du paramètre de puissance. Le théorème 6 de Song [68] indique que les équations (III.7) et (III.8) admettent la même solution.

Algorithme III.2 : Pseudo-code de l'algorithme d'estimation des paramètres de GΓD

Données : Un ensemble de N réalisations $(x_n)_{n=1}^N$
Résultat : L'estimé au sens du maximum de vraisemblance du paramètre $\hat{\beta}$
Initialisation de β_1 ;
Choix arbitraire de la fonction $F \in \{S_N, T_N\}$;
pour chaque *k prenant des valeurs de 1 à K* **faire**
  $\beta_{k+1} = \beta_k - F(\beta_k)/F'(\beta_k)$; `/* Mise à jour */`
fin
$\hat{\beta} = \beta_K$; `/* Estimé de` β `*/`

L'algorithme III.2 reprend le processus d'estimation des paramètres d'une GΓD, il converge de manière quadratique rapide [68]. Dans un second temps, les paramètres d'échelle $\hat{\alpha}$ et de forme $\hat{\lambda}$ sont estimés par les formules :

$$\hat{\lambda} = (\hat{\beta}(\mathcal{R}'_N(\hat{\beta}) - \mathcal{R}'_N(0))^{-1} \quad \text{(III.9)}$$

$$\hat{\alpha} = \left(\frac{1}{\hat{\lambda}}\mathcal{M}_N(\hat{\beta})\right)^{1/\hat{\beta}} \quad \text{(III.10)}$$

2.5.3 Existence et unicité

Soient δ un réel strictement positif et β_0 la valeur du paramètre de puissance. Song [68] assure que les estimateurs des paramètres d'échelle α et de forme λ sont consistants. Pour le paramètre de puissance $\hat{\beta}$, le système d'équation (III.7) admet une solution unique sur l'ensemble $(\beta_0 - \delta, +\infty)$ pour une valeur strictement positive de δ. La suite, $(\hat{\beta}_k)_{k=1}^\infty$ définie par la méthode de Newton-Raphson converge en probabilité vers le paramètre de puissance réel β_0.

2.5.4 Mesures de dissimilarité

Avec la définition de la densité de probabilité de la GΓD, il est possible d'expliciter la formule de la divergence de Kullback-Leibler (KLD) entre deux distributions GΓD ayant comme vecteurs paramétriques respectives $\theta = (\alpha, \beta, \lambda)$ et $\theta' = (\alpha', \beta', \lambda')$ [44] :

$$\mathrm{KLD}(p(x|\theta) \parallel p(x|\theta')) = \log\left\{\frac{\beta\alpha'^{\beta'\lambda'}\Gamma(\lambda')}{\beta'\alpha^{\beta\lambda}\Gamma(\lambda)}\right\} + \left(\lambda - \lambda'\frac{\beta'}{\beta}\right)\Psi(\lambda) + \left(\frac{\alpha}{\alpha'}\right)^{\beta'}\frac{\Gamma(\lambda+\beta'/\beta)}{\Gamma(\lambda)} - \lambda. \quad \text{(III.11)}$$

La définition de la divergence de Jeffrey permet d'obtenir une formule explicite [44] à partir de la formule (III.11) :

$$\begin{aligned}\mathrm{JD}(p(x|\theta), p(x|\theta')) =& (\beta\lambda - \beta'\lambda')\left(\log\left\{\frac{\alpha}{\alpha'}\right\} + \frac{\Psi(\lambda)}{\beta} - \frac{\Psi(\lambda')}{\beta'}\right) + \\ & \left(\frac{\alpha}{\alpha'}\right)^{\beta'}\frac{\Gamma(\lambda+\beta'/\beta)}{\Gamma(\lambda)} + \left(\frac{\alpha'}{\alpha}\right)^{\beta}\frac{\Gamma(\lambda'+\beta/\beta')}{\Gamma(\lambda')} - \lambda - \lambda'.\end{aligned}$$

Nous avons utilisé la distribution Gamma généralisée dans [66]. ∎

2.6 Spherically Invariant Random Vector

Lasmar et Berthoumieu [55] proposent de modéliser la distribution des coefficients associés à leurs 8 coefficients voisins : le vecteur aléatoire $(\vec{x}_n)_{n=1}^N$ est constitué des coefficients d'ondelettes situés spatialement en :

$$\begin{array}{ccc}(2^s(i-1), 2^s(j+1)), & (2^s i, 2^s(j+1)), & (2^s(i+1), 2^s(j+1)),\\ (2^s(i-1), 2^s j), & (2^s i, 2^s j), & (2^s(i+1), 2^s j),\\ (2^s(i-1), 2^s(j-1)), & (2^s i, 2^s(j-1)), & (2^s(i+1), 2^s(j-1)),\end{array}$$

avec i et j deux entiers vérifiant $n = i + (j-1) * 128/(2^s)$ pour une image d'origine de taille 128×128 pixels. Sans perte de généralité, cette dépendance est nommée dépendance spatiale intra-sous-bande 3×3. Par extension, considérons la dépendance spatiale $p \times q$ avec p et q deux entiers naturels. La distribution « Spherically Invariant Random Vector » (SIRV) est proposée pour modéliser ce vecteur aléatoire $(\vec{x}_n)_{n=1}^N$ de dimensions pq (un voisinage de $p \times q$ coefficients dans la sous-bande correspond à un vecteur aléatoire avec pq composantes). Contrairement aux deux exemples de distributions univariées fondées sur l'hypothèse d'indépendance, ici le vecteur aléatoire réside dans un espace vectoriel de

dimension pq et une flèche est présente au-dessus du nom du vecteur aléatoire pour se rappeler de son caractère multi-dimensionnel. Les réalisations du vecteur aléatoire $(\vec{x}_n)_{n=1}^N$ sont supposées indépendantes et identiquement distribuées au sein d'une même sous-bande.

2.6.1 La densité de probabilité jointe

$\vec{x}$ est un vecteur aléatoire suivant un modèle SIRV, s'il est le résultat d'un produit entre la racine carrée d'une variable aléatoire positive τ, qui est nommée multiplieur, et un vecteur aléatoire $\vec{g}$ suivant une gaussienne multivariée circulaire indépendante et centrée avec comme matrice de covariance $M = \mathbb{E}[\vec{g}^T\vec{g}]$:

$$\vec{x} = \sqrt{\tau} \times \vec{g} \tag{III.12}$$

Nous introduisons par la même les notations

- de transposée du vecteur $\vec{g}$ en ajoutant un T majuscule en exposant du vecteur comme suit $\vec{g}^T$;
- de trace $Tr(M)$ de la matrice M soit la somme de tous ses éléments diagonaux ;
- d'espérance mathématique $\mathbb{E}$, comme nous travaillons sur une variable aléatoire unique, la valeur attendue est calculée à partir de la distribution du vecteur aléatoire $\vec{g}$ soit :

$$\mathbb{E}[\vec{g}^T\vec{g}] = \int \dots \int_{\mathbb{R}^{pq}} \vec{g}^T\vec{g}\,\phi(\vec{g} \mid M).d\vec{g}$$

- la densité de probabilité ϕ d'une gaussienne multivariée centrée de matrice de covariance M :

$$\phi(\vec{g}|M) = (2\pi)^{-pq/2}(|M|)^{-1/2} \exp\left\{-\frac{1}{2}\vec{g}^T M^{-1}\vec{g}\right\}$$

La densité de probabilité jointe du vecteur aléatoire $\vec{x}$ est paramétrée par une matrice de covariance M et par la densité du multiplieur $p_\tau(\tau)$:

$$X(\vec{x} \mid p_\tau(\tau), M) = \int_{\mathbb{R}^+} \frac{p_\tau(\tau)}{(2\pi)^{pq/2}|\tau M|^{1/2}} \exp\left\{-\frac{\vec{x}^T M^{-1}\vec{x}}{2\tau}\right\} d\tau \tag{III.13}$$

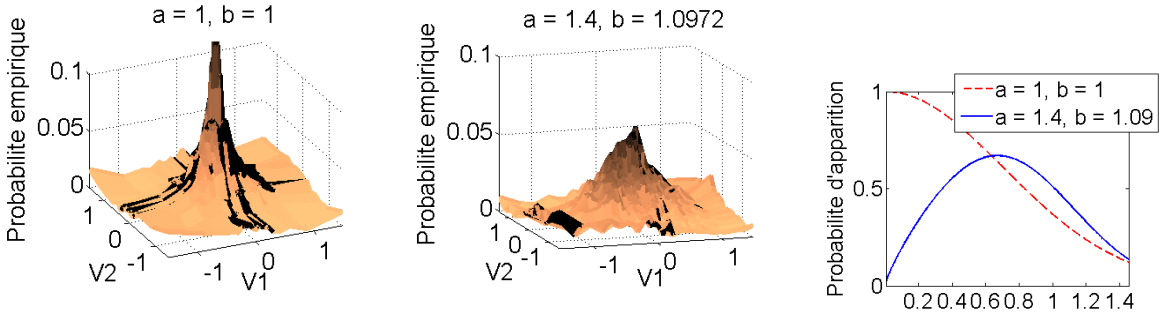
Le modèle SIRV peut approcher des distributions usuelles par le choix de la densité de probabilité du multiplieur. Lasmar et Berthoumieu [55] proposent de modéliser la distribution du multiplieur τ par une distribution Weibull :

$$p_\tau(\tau \mid a, b) = a\frac{\tau^{a-1}}{b^a} \exp\left\{-\left(\frac{\tau}{b}\right)^a\right\}, \qquad \forall \tau > 0 \tag{III.14}$$

avec a le paramètre de forme et b le paramètre d'échelle. L'application de la densité de probabilité du multiplieur (III.14) dans la formule de la probabilité jointe (III.13) a pour résultat :

$$p(\vec{x} \mid a, b, M) = \frac{a}{b^a} \int_{\mathbb{R}^+} \frac{\tau^{a-1}}{(2\pi)^{pq/2}|\tau M|^{1/2}} \exp\left\{-\frac{\vec{x}^T M^{-1} \vec{x}}{2\tau} - \left(\frac{\tau}{b}\right)^a\right\} d\tau \tag{III.15}$$

Il reste encore à déterminer la formule explicite de cette intégrale.



(a) $\vec{x}_1$ de paramètres $a_1 = 1$, $b_1 = 1$, $M = \begin{bmatrix} 1 & 0,2 \\ 0,2 & 1 \end{bmatrix}$

(b) $\vec{x}_2$ de paramètres $a_2 = 1,4$, $b_2 = 1,09$, $M = \begin{bmatrix} 1 & 0,2 \\ 0,2 & 1 \end{bmatrix}$

(c) distribution de $\sqrt{\hat{\tau}_1}$ en rouge et $\sqrt{\hat{\tau}_2}$ en bleu

Figure III.6 – Représentation de la distribution empirique de deux vecteurs aléatoires $\vec{x}_1$ et $\vec{x}_2$ dans la figure (a) et (b) respectivement. $\vec{x}_1$ et $\vec{x}_2$ ont été générées suivant un modèle SIRV de dimension 2, avec une distribution Weibull pour la loi du multiplieur τ_1 et τ_2 respectivement.

Les figures III.6.(a) à III.6.(c) présentent des résultats théoriques et pratiques sur la distribution d'un vecteur aléatoire en dimension 2. Un vecteur aléatoire $\vec{g}$ qui suit une distribution gaussienne bivariée centrée de matrice de covariance :

$$M = \begin{bmatrix} 1 & 0,2 \\ 0,2 & 1 \end{bmatrix}$$

est généré. Ensuite une variable aléatoire τ_1 suivant une distribution Weibull de paramètre de forme $a = 1$ et $b = 1$ est générée. Le vecteur aléatoire $\vec{x}_1 = \sqrt{\tau_1}\vec{g}$ construit comme le produit de la racine carrée de τ_1 avec $\vec{g}$ admet, d'après la définition, une distribution faisant partie des modèles SIRV. La figure III.6.(a) représente la distribution empirique de $N = 10^4$ réalisations du vecteur aléatoire $(\vec{x}_{1,n})_{n=1}^N$. $v1$ et $v2$ représentent respectivement la distribution de $(x_{1,n,1})_{n=1}^N$ et de $(x_{1,n,2})_{n=1}^N$, sachant que $\vec{x}_{1,n} = (x_{1,n,1}; x_{1,n,2})$ pour tout $n = 1, \ldots, N$. La distribution théorique de $\sqrt{\tau_1}$ est affichée par une courbe rouge discontinue dans la figure III.6.(c).

Dans une deuxième étape, une variable aléatoire τ_2 est générée, suivant une distribution Weibull, de

paramètre de forme $a = 1,4$ et $b = 1,0972$. La variable aléatoire $\vec{x}_2 = \sqrt{\tau_2}\vec{g}$ construite comme le produit de la racine carrée de τ_1 avec $\vec{g}$ admet aussi une distribution faisant partie des modèles SIRV. La figure III.6.(b) représente la distribution empirique de $N = 10^4$ réalisations de la variable aléatoire $(\vec{x}_{2,n})_{n=1}^N$. Sans perte de généralité la variable aléatoire $(\vec{g}_n)_{n=1}^N$ n'a été générée qu'une seule fois pour créer chacune des deux collection $(\vec{x}_{1,n})_{n=1}^N$ et $(\vec{x}_{2,n})_{n=1}^N$. Il est ainsi possible d'observer la différence apportée par le multiplieur. La distribution théorique de $\sqrt{\tau_2}$ est affichée par une courbe bleue pleine dans la figure III.6.(c).

Le paramètre de forme a modifie la forme de la distribution, lorsque a est petit la distribution est plus piquée, tandis que lorsque a est grand, la distribution est plus aplatie et converge vers une distribution uniforme sur l'ensemble de définition. Dans les figures III.6.(a) et (b) la matrice de corrélation M est la même, dès lors la différence entre les distributions empirique est plus difficile à apprécier pour des paramètres de forme a élevés, car dans ce cas les distributions sont toute très aplaties.

Il est possible de remarquer dans la figure III.6.(c) que la distribution de $\sqrt{\tau}_2$ à un mode à racine carrée de tau égale à $\sqrt{\tau} = 0,65$ strictement positif contrairement à la distribution de $\sqrt{\tau}_1$ (dont le mode est nul) mais que la distribution de $\vec{x}_2$ (figure III.6.(b)) a toujours le même mode $(0;0)$ que la distribution de $\vec{x}_1$ (figure III.6.(a)). De plus la distribution de $\sqrt{\tau_1}$ est très différente de la distribution de $\sqrt{\tau_2}$ alors que la différence entre les distributions de $\vec{x}_1$ et $\vec{x}_2$ semble moins importante.

2.6.2 Estimation des paramètres

Soit h_{pq} la fonction génératrice de densité donnée par [69, 70]

$$h_{pq}(x) = \int_0^{+\infty} \frac{p_\tau(\tau)}{\tau^{pq}} e^{-x/\tau}.d\tau \tag{III.16}$$

h_{pq} est utilisée dans la formule du maximum de vraisemblance $\hat{M}_{mv}$ de la matrice de covariance

$$\hat{M}_{mv} = \frac{1}{N}\sum_{n=1}^{N} \frac{h_{pq+1}(\vec{x}_n^T \hat{M}_{mv}^{-1}\vec{x}_n)}{h_{pq}(\vec{x}_n^T \hat{M}_{mv}^{-1}\vec{x}_n)} \vec{x}_n \vec{x}_n^T. \tag{III.17}$$

Il est nécessaire de comprendre que l'estimateur au sens du maximum de vraisemblance $\hat{M}_{mv}$ dépend de la densité de probabilité du multiplieur p_τ. Chitour et Pascal [71] ont démontré que l'équation III.17 admet une solution unique, l'algorithme d'estimation itératif qu'ils définissent converge vers une solution au point fixe $\hat{M}$. Le multiplieur possède comme densité de probabilité une distribution Weibull, par conséquent la formule explicite de l'estimateur du maximum de vraisemblance $\hat{M}_{mv}$ reste à être trouvée (III.15). Un

estimateur du point fixe $\hat{M}$ est préféré à un estimateur au sens du maximum de vraisemblance.

Gini et Greco [72] proposent un estimateur au sens du maximum de vraisemblance approché (aussi connue comme la méthode du point fixe), pour cela ils définissent une fonction f comme

$$f\left(\hat{M}\right) = \frac{pq}{N}\sum_{n=1}^{N}\frac{\vec{x}_n\vec{x}_n^T}{\vec{x}_n^T\hat{M}^{-1}\vec{x}_n} \tag{III.18}$$

et l'unique point fixe de la fonction f est l'estimateur de la matrice de covariance $\hat{M} = f\left(\hat{M}\right)$. La section % 1 de l'algorithme III.3 présente l'estimation du point fixe.

La suite $(M_k)_{k=0}^{+\infty}$ ainsi définie converge vers un point fixe de la fonction f [73] ayant même norme de Frobenius que l'initialisation M_0. Une fois obtenue la matrice $\hat{M}$ gérant les corrélations de la variable aléatoire $\vec{x}$, l'étape suivante consiste à estimer au sens du maximum de vraisemblance des paramètres d'échelle b et de puissance a de la distribution Weibull du multiplieur τ à partir des réalisations $(\vec{x}_n)_{n=1}^N$ de la variable aléatoire et la matrice estimée $\hat{M}$. Dans un premier temps, la section % 2 de l'algorithme III.3 réalise l'estimation du paramètre de forme $\hat{a}$ [74]. Dans un second temps, la section % 3 de l'algorithme III.3 estime au sens du maximum de vraisemblance le paramètre d'échelle $\hat{b}$ en fonction des réalisations $(\vec{x}_n)_{n=1}^N$ de la variable aléatoire, de la matrice estimée $\hat{M}$ et du paramètre de forme $\hat{a}$ estimé.

2.6.3 Existence et unicité

Les couples $(p_\tau; M)$ ne sont pas uniques pour représenter la distribution du vecteur aléatoire $\vec{x}_n$. Considérons un réel strictement positif r, alors si la matrice de covariance estimée est égale à rM à la place de M, le multiplieur aurait une dynamique réduite de r. Quelque soit la valeur du réel $r > 0$ et pour une distribution Weibull, le vecteur aléatoire $\vec{x}_n$ suit un modèle SIRV de paramètres $(b/r; a; rM)$. Sans perte de généralité, posons $t = \tau/r$. La variance de la variable aléatoire t ainsi créée vérifie $\mathbb{E}[t^2] = \mathbb{E}[\tau^2]/r^2$. La variable aléatoire t suit une distribution Weibull de paramètres de forme $a_t = a$ et d'échelle $b_t = b/r$.

Le but est de restreindre les ensembles de paramètres différents utilisés pour représenter le vecteur aléatoire $\vec{x}$. La transformation en t du multiplieur τ montre que la variance est un homéomorphisme du réel $r > 0$ dans l'ensemble des réels strictement positifs. Supposons que $\mathbb{E}[t^2] = 1$ un réel strictement positif. Alors les paramètres de forme a_t et d'échelle b_t de cette distribution vérifient l'équation $b_t = (\Gamma(1/a_t + 1))^{-1}$. Ce qui donne une valeur pour

$$r = b\Gamma\left(\frac{1}{a} + 1\right).$$

Algorithme III.3 : Pseudo-code de l'algorithme d'estimation des paramètres pour un modèle SIRV

Données : Un ensemble de N réalisations $(\vec{x}_n)_{n=1}^N$
Résultat : Les estimés au sens du maximum de vraisemblance des paramètres $\hat{a}$, $\hat{b}$, $\hat{M}$

% 1. estimation de la matrice de covariance M
$M \leftarrow \frac{1}{N}\sum_{n=1}^{N} \vec{x}_n \vec{x}_n^T$; /* Initialisation */
pour chaque *k prenant des valeurs de 1 à K* **faire**
  pour chaque *n prenant des valeurs de 1 à N* **faire**
    $\tau_n \leftarrow \frac{1}{pq} \vec{x}_n^T M^{-1} \vec{x}_n$; /* Mise à jour */
  fin
  $M \leftarrow \frac{1}{N}\sum_{n=1}^{N} \frac{\vec{x}_n \vec{x}_n^T}{\tau_n}$; /* Mise à jour */
fin
$\hat{M} \leftarrow M$; /* Estimé de la matrice */
% 2. Estimation du paramètre de forme a
pour chaque *n prenant des valeurs de 1 à N* **faire**
  $\hat{\tau}_n \leftarrow \frac{1}{pq} \vec{x}_n^T \hat{M}^{-1} \vec{x}_n$;
fin
$a \leftarrow 1$; /* Initialisation */
$f : a \rightarrow \frac{\sum_{n=1}^{N} \hat{\tau}_n^a \log\{\hat{\tau}_n\}}{\sum_{n=1}^{N} \hat{\tau}_n^a - \log\{\hat{\tau}_n\}} - \frac{1}{a}$;
pour chaque *k prenant des valeurs de 1 à K* **faire**
  $a \leftarrow a - f(a)/f'(a)$; /* Mise à jour */
fin
$\hat{a} \leftarrow a$; /* Estimé de a */
% 3. Estimation du paramètre d'échelle b
$\hat{b} \leftarrow \left(\frac{1}{N}\sum_{n=1}^{N} \hat{\tau}_n^{\hat{a}}\right)^{1/\hat{\alpha}}$;

Le couple $(\hat{a}; \hat{b}\Gamma\left(\frac{1}{\hat{a}}+1\right)\hat{M})$ caractérise de manière unique le modèle SIRV lié au vecteur aléatoire $\vec{x}_n$.

2.6.4 Mesure de dissimilarité

La densité jointe donnée par l'équation (III.13) avec une distribution Weibull pour le multiplieur n'admet pas de forme explicite. Par conséquent la divergence de Kullback-Leibler n'admet pas de forme explicite. Lasmar [48] propose de modéliser le vecteur augmenté $\vec{y} = [\tau; \vec{g}]$ de dimension $pq+1$ plutôt qu'une modélisation uniquement fondée sur $\vec{x}$. La réalisation est représentée par la matrice :

$$y = \begin{pmatrix} \tau_1 & \dots & \tau_n & \dots & \tau_N \\ g_{1,1} & \dots & g_{1,n} & \dots & g_{1,N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ g_{pq,1} & \dots & g_{pq,n} & \dots & g_{pq,N} \end{pmatrix}$$

En utilisant la modélisation SIRV,et puisque τ et le noyau gaussien $\vec{g}$ sont indépendants, la densité jointe de $\vec{y}$ est :

$$Y(\vec{y} \mid \theta) = p_\tau(\tau \mid a, b) \times \phi(\vec{g} \mid M)$$

Sous un principe d'isométrie, entre les espaces des lois $X \Leftrightarrow (a, b, M)$ et $Y \Leftrightarrow (a, b, M)$, il y aurait équivalence à développer une mesure de dissimilarité entre les distributions liées à $\vec{x}$ et à $\vec{y}$. Sans perte de généralité, le reste de ce mémoire utilise la distribution jointe $Y(\vec{y} \mid \theta)$ et la note $X(\vec{x} \mid \theta)$.

La divergence de Kullback-Leibler entre les distributions de modèles SIRV de paramètre $\theta = (a; b; M)$ et $\theta' = (a'; b'; M')$ est définie comme la somme d'une divergence de Kullback-Leibler entre les distributions gaussienne, multivariées centrées de matrices de covariance M et M' et d'une divergence entre les distributions Weibull de paramètres $(a; b)$ et $(a'; b')$

$$\mathrm{KLD}(X(\vec{x} \mid \theta) \parallel X(\vec{x} \mid \theta')) = \mathrm{KLD}(p_\tau(\tau \mid a, b) \parallel p_\tau(\tau \mid a', b')) + \mathrm{KLD}(\phi(\vec{g} \mid M) \parallel \phi(\vec{g} \mid M')). \quad \text{(III.19)}$$

La formule explicite [55] de la divergence de Kullback-Leibler s'écrit :

$$\begin{aligned} \mathrm{KLD}(X(\vec{x} \mid \theta) \parallel X(\vec{x} \mid \theta')) = & \log\left\{\frac{ab'^{a'}}{a'b^{a'}}\right\} + \left(\frac{b}{b'}\right)^{a'} \Gamma\left(\frac{a'}{a}+1\right) + \gamma\left(\frac{a'}{a}-1\right) - 1 + \\ & \frac{1}{2}\left[Tr(M'^{-1}M) + \log\left\{\frac{|M'|}{|M|}\right\} - pq\right] \end{aligned} \quad \text{(III.20)}$$

avec $\gamma = -\Gamma'(1) \simeq 0,577$ la constante d'Euler-Mascheroni, $\Gamma(z) = \int_0^{+\infty} t^{z-1}e^{-t}.dt$ la fonction Gamma,

$|M|$ (respectivement $|M'|$) le déterminant de la matrice de covariance M (respectivement M'). Par simple extension, voici la formule de la divergence de Jeffrey jointe entre deux modèles SIRV à multiplieur distribué selon une distribution Weibull :

$$\mathrm{JD}(p(\vec{x} \mid \theta) \parallel p(\vec{x} \mid \theta')) = \log\left\{\frac{b^a b'^{a'}}{b'^a b^{a'}}\right\} + \left(\frac{b}{b'}\right)^{a'} \Gamma\left(\frac{a'}{a}+1\right) + \left(\frac{b'}{b}\right)^{a} \Gamma\left(\frac{a}{a'}+1\right) + \gamma\left(\frac{a'}{a}+\frac{a}{a'}-2\right) + \frac{1}{2}\left[Tr(M'^{-1}M) + Tr(M^{-1}M')\right] - pq - 2$$

Nous avons utilisé le modèle SIRV avec multiplieur Weibull dans deux publications en 2013 [64, 75].

Cette section nous a permis de faire un état de l'art sur les descripteurs paramétriques en présentant trois modèles paramétriques qui seront utilisés dans la suite de ce mémoire. Le modèle SIRV est défini comme un modèle prenant en compte les dépendances spatiales, nous pouvons espérer obtenir des performances de classifications accrues par rapport aux hypothèses d'indépendances. Dans la suite du document, nous allons revenir sur le schéma global de la méthode SMV dans l'optique de montrer comment les descripteurs paramétriques peuvent être utilisés

3 Conséquences de la diversité sur les descripteurs

Une image texturée est le résultat d'un processus physique. Il est possible de partitionner le processus physique en terminologie élémentaire dont une liste non exhaustive serait : la puissance en watts de la lampe qui éclaire la scène, la position et l'orientation de la lampe par rapport au couple objet-appareil photo, le lacet, le tangage et le roulis de l'objet par rapport à l'appareil photo, la distance entre l'objet et l'appareil photo, la réflectivité de tout ou partie de l'objet, la transparence de l'objet, le bruit d'acquisition de l'appareil photo ... Lors de l'acquisition d'une image, il est possible de fixer plusieurs de ces paramètres. Néanmoins, une simple différence dans un paramètre laissé libre peut présenter des changements importants dans l'image résultante. La « diversité naturelle » est le terme qui est associé à l'ensemble des changements pouvant survenir lors de l'acquisition d'une image texturée. Dans une première partie, nous montrons l'impact au niveau image de la diversité. Dans une seconde partie, nous montrons l'impact au niveau des descripteurs paramétriques de ce même changement.

3.1 Impact au niveau image

Dans ce chapitre, nous présentons des images acquises sous différentes conditions d'illumination ou de prise de vue. Les images ont été acquises par les scientifiques du laboratoire « Intelligent Sensory Information System », Université d'Amsterdam [2]. Dans une première partie, le processus d'acquisition des images texturées sera décrit au moyen du schéma III.7. Ensuite un paragraphe présente une discussion sur les images acquises.

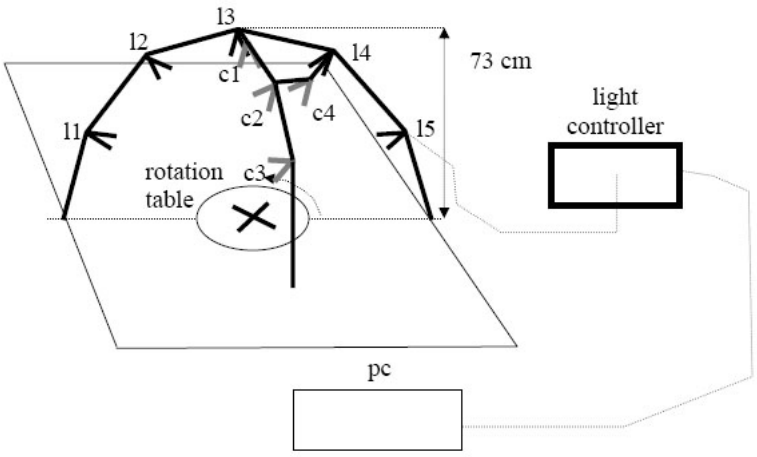
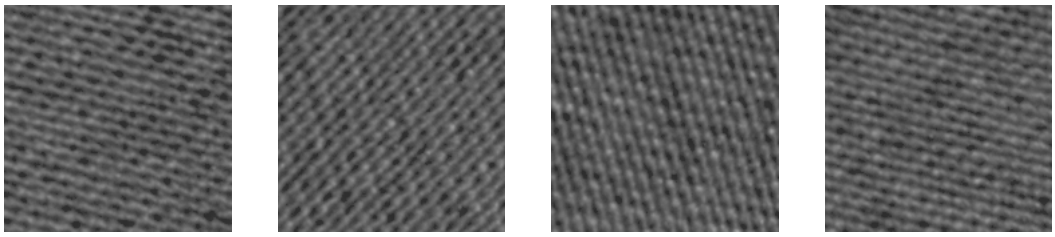


Figure III.7 – Configuration de l'acquisition des images texturées (ALOT)

Le schéma III.7 présente l'installation matérielle réalisée pour l'acquisition. La description des différentes parties de ce schéma est réalisée en deux temps : le point de vue et l'orientation de l'éclairage. Commençons par le point de vue de la caméra. Les objets à photographier sont posés sur une « rotation table » (tablette munie d'un dispositif pour la faire tourner dans le plan de la table), et les acquisitions se font pour un jeu d'angles : 0, 60, 120 et 180 degrés. L'acquisition à angle de 180 degrés est utilisé principalement pour mesurer l'invariance au bruit des algorithmes de classification. 4 appareils photos ont été utilisés pour les acquisitions, numérotés de c1 à c4, trois appareils, respectivement c1, c2 et c3, sont positionnés perpendiculairement à l'axe des sources lumineuses situées à un élévation de 80, 60 et 40 degrés respectivement. De plus, l'appareil c4 est positionné à 60 degrés à droite de c2 qui est élevé de 60 degrés. La position de c4 par rapport à c2 à pour conséquence de changer l'orientation des sources lumineuses d'exactement 60 degrés.

Nous présentons maintenant les différentes conditions d'éclairage. Les images sont acquises avec seule

une lampe d'allumée parmi les 5, amenant à 5 conditions d'éclairage différent nommées l1 à l5. De plus, allumer les 5 lampes en même temps permet d'approcher un éclairage hémisphérique (notée l8), sans pour autant couvrir un hémisphère entier. Notez que, pour l'appareil photo c4, l'objet subit une rotation afin d'avoir l'aspect visuel de l'appareil c2. L'appareil photo c4 permet d'apprécier des éclairages tournés d'un angle de 60 degrés en direction de l'appareil. En bref, les appareils photo c2 et c4 permettent d'acquérir le même objet, avec des conditions d'éclairage ayant subit une rotation de 60 degrés azimuth. Ajoutons que pour l'appareil photo c1 qui observe l'objet presque depuis son dessus, les acquisitions des images après une rotation de la table peuvent être vues comme des rotations de la position de la source lumineuse (après une correction de l'orientation de l'image).



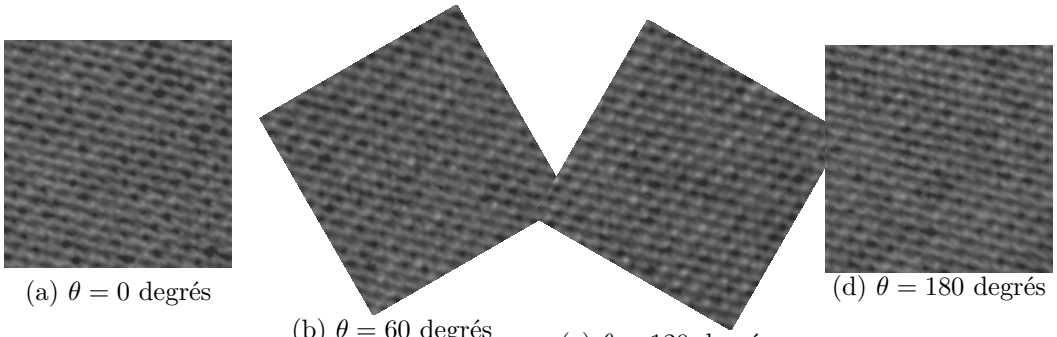
(a) $\theta = 0$ degrés (b) $\theta = 60$ degrés (c) $\theta = 120$ degrés (d) $\theta = 180$ degrés

Figure III.8 – Coton (Vêtement rouge), éclairage haut et rotation d'angle θ. Amsterdam Library of Textures (ALOT)

La figure III.8 présente un seul et même objet. Les conditions d'éclairage et le point de vue de l'appareil photo sont fixes et le plateau où l'objet est fixé tourne. Les images présentées sont prises successivement à des angles de 0 degré (Figure III.8.(a)), 60 degrés (Figure III.8.(b)), 120 degrés (Figure III.8.(c)) et 180 degrés (Figure III.8.(d)). Les images sont très directionnelles et il est possible de visualiser le résultat de la rotation de l'objet : la direction du tissu est modifié de l'angle de rotation θ. Il est important que ces 4 images soient confondues en une seule et même classe, autrement dit avoir une forte similarité.

Avant d'afficher les images de la figure III.9, une correction de l'orientation θ est appliquée à chaque image de la figure III.8 une rotation d'angle égal à $-\theta$. La correction de la rotation est suffisante pour comparer les 4 images. De manière plus générale, toutes les textures ne montrent pas une telle homogénéité et une correction de l'orientation peut ne pas être suffisante.

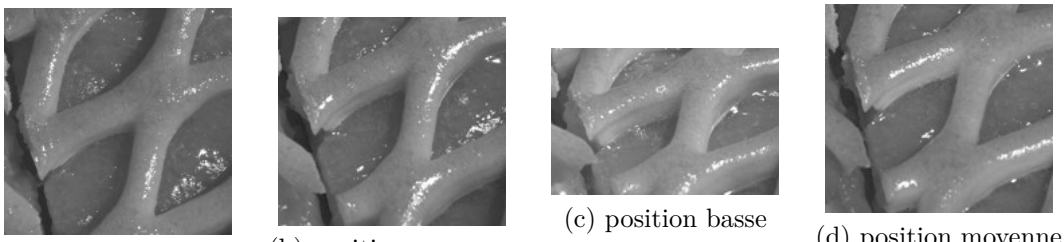
La figure III.10 présente un même objet vu depuis quatre positions différentes : une position presque perpendiculaire à l'objet (Figure III.10.(a)), une position proche de l'horizontale (Figure III.10.(c)), deux positions intermédiaires écartées de 60 degrés (Figures III.10.(b) et III.10.(d)). La figure III.10.(c) permet d'apprécier la profondeur du relief de l'objet ainsi que l'absence de certains détails non visibles dans la



(a) $\theta = 0$ degrés (b) $\theta = 60$ degrés (c) $\theta = 120$ degrés (d) $\theta = 180$ degrés

Figure III.9 – Coton (Vêtement rouge), éclairage haut et rotation d'angle θ. Une rotation est appliquée aux images afin de corriger l'orientation de l'objet. Amsterdam Library of Textures (ALOT)

figure III.10.(a). Mais le fort relief de l'objet montre qu'il est possible d'occulter une partie de l'information qui se trouvait dans la figure III.10.(a). Les deux figures III.10.(a) et III.10.(c) sont complémentaires car elles apportent ensemble plus d'information sur la texture que chacune séparément. C'est une richesse importante lorsque l'algorithme de classification est en phase d'apprentissage.



(a) position haute (b) position moyenne (c) position basse (d) position moyenne droite

Figure III.10 – Gâteau aux pommes, éclairage haut droit et modification du point de vue de l'appareil photo. Amsterdam Library of Textures (ALOT)

La figure III.11 présente un seul et même objet mais acquis sous des conditions d'illumination différentes. La lampe éclairant la scène peut être située soit à gauche de l'objet III.11.(a), soit située à gauche mais surélevée par rapport à l'objet III.11.(b), soit située à la verticale de l'objet III.11.(c), ou située à droite mais surélevée par rapport à l'objet III.11.(d) et située à droite de l'objet III.11.(e). Les tâches blanches de saturation apparaissent sur les faces des volumes de la texture correspondant à l'orientation de la source lumineuse. Explicitement les tâches sont positionnées de gauche III.11.(a) à droite III.11.(e). Dans ces mêmes images, les volumes de la texture créent des ombres, ce qui a pour effet d'accentuer les volumes de la texture. L'ombre remplace l'information comprise dans les maillages du tapis par du

noir III.11.(a), III.11.(b), III.11.(d) et III.11.(e). Sauf pour l'éclairage à la verticale III.11.(c) où l'ombre est absente, ici ce sont les tâches de saturation qui remplacent certaines informations par du blanc.

(a) éclairage gauche (b) éclairage haut gauche (c) éclairage haut (d) éclairage haut droit (e) éclairage droit

Figure III.11 – Tapis, différentes configurations d'éclairage et même orientation. Amsterdam Library of Textures (ALOT)

Au moyen de trois exemples simples, les figures III.8, III.10 et III.11 présentent la diversité existante pour représenter un seul et même objet. La diversité est surtout le résultat d'une modification géométrique de la scène. Les modifications géométriques ont modifié le contenu des images en ajoutant des pixels noirs (pour l'ombre) ou blancs (pour les tâches de saturation) à une image qui en avait moins. Autrement dit, les modifications géométriques influent sur la distribution des niveaux de gris des images. Les descripteurs statistiques présentés dans la section 2 vont donc être impactés par les différences dans les distributions des niveaux de gris des images. Afin d'étudier le résultat de la diversité intra-classe, des descripteurs paramétriques vont être extraits des images choisies comme exemple (figures III.8, III.10 et III.11). Nous présentons dans ce qui suit l'expérimentation, les résultats et une courte discussion sur les résultats obtenus.

3.2 Impact au niveau des descripteurs

Dans cette sous-section, nous proposons d'étudier l'impact que peut avoir la diversité intra-classe sur des descripteurs paramétriques. Une série d'images exemples est donnée dans les figures III.8, III.10 et III.11. Il est donc question ici d'extraire de chaque image un vecteur de descripteurs. L'extraction des descripteurs est précédée par une normalisation de l'image. Après l'extraction des descripteurs il est également possible d'effectuer une nouvelle normalisation [31]. Le pré-traitement utilisé consiste à modifier la distribution des niveaux de gris de l'image. Les niveaux de gris de l'image peuvent être considérés comme des réalisations d'une variable aléatoire réelle $x = (x_n)_{n=1}^N$. Ensuite la distribution de $x = (x_n)_{n=1}^N$ est normalisée. Pour cela, il est nécessaire de calculer les deux moments de premier ordre. La variable aléatoire x peut être normalisée par soustraction du moment d'ordre 1 et division par le moment

d'ordre 2 :

$$\tilde{z}_n = \frac{x_n - m_1}{\sqrt{m_2}}.$$

Sans perte de généralité, le symbole $x = (x_n)_{n=1}^N$ représentera la variable aléatoire normalisée $(\tilde{z}_n)_{n=1}^N$.

Suite au pré-traitement l'extraction du descripteur est réalisée. Ici les coefficients de la décomposition en ondelettes stationnaire 2D sont supposés indépendants et identiquement distribués, suivant le modèle paramétrique gaussienne généralisée (GGD). L'étude qui suit, peut être effectuée avec une autre décomposition en trames par analyse et un autre modèle paramétrique sans perte de généralité. Le choix du descripteur paramétrique obéit à certaines contraintes en pratique. Soit une sous-image de dimensions 512×512 pixels extraite de l'image nommé patch. $P = 6$ est le nombre de patchs extraits d'une image de dimension 1536×1024 pixels (cf. l'image III.8.(a)), sachant que le centre de 2 patchs est distant d'un minimum de 500 pixels.

Aussi il est possible d'extraire P patchs différents. Ce paragraphe précise donc la démarche utilisée pour estimer le descripteur paramétrique. Pour la suite, soit I un patch normalisé extrait de l'image. La décomposition en ondelettes stationnaire 2D de I dispose de $N_o = 3$ orientations (horizontale, verticale et diagonale) sur $N_s = 2$ échelles de décomposition. Au total, la décomposition de I est faite en 6 sous-bandes. Soit $x_{s,o} = (x_{s,o,n})_{n=1}^N$ une variable aléatoire dont les réalisations correspondent aux coefficients de sous-bande définie par la paire $(s; o)$. $x_{s,o}$ est supposée indépendante de $x_{s',o'}$ pour tout $(s; o) \neq (s'; o')$. $x_{s,o}$, supposée indépendante et identiquement distribuée, suit une distribution gaussienne généralisée (GGD) de paramètres $\theta_{s,o} = (\alpha_{s,o}; \beta_{s,o})$. En pratique, le vecteur paramétrique $\theta_{s,o}$ est estimé avec l'estimateur au sens du maximum de vraisemblance présenté à la section 2.4.2. Le descripteur paramétrique de I est défini comme une combinaison des vecteurs paramétriques estimés au sens du maximum de vraisemblance sur chaque sous-bande

$$\theta = (\theta_{s,o})_{s,o=1}^{s=N_s,o=N_o}$$

Pour rappel, la GGD est paramétrée par 2 paramètres qui sont le paramètre d'échelle α et le paramètre de forme β. Puis la décomposition est faite en 6 sous-bandes. Par conséquent, la dimension du descripteur est $12 = 6 \times 2$ ce qui n'est pas représentable dans le plan. Donc les vecteurs paramétrique $\hat{\theta}_{s,o}$ seront présentés pour une sous-bande. Sans perte de généralité, seule la première sous-bande $\hat{\theta}_{1,1}$ sera représentée. Les figures III.12, III.13 et III.14 présentent alors le plan paramétrique de la GGD avec le paramètre d'échelle α de la GGD en abscisse et le paramètre de forme β de la GGD en ordonnée, spécifique à la première sous-bande (première échelle de décomposition, horizontale). Pour chaque image, ce sont P vecteurs

paramétriques qui sont estimés au sens du maximum de vraisemblance sur la même sous-bande.

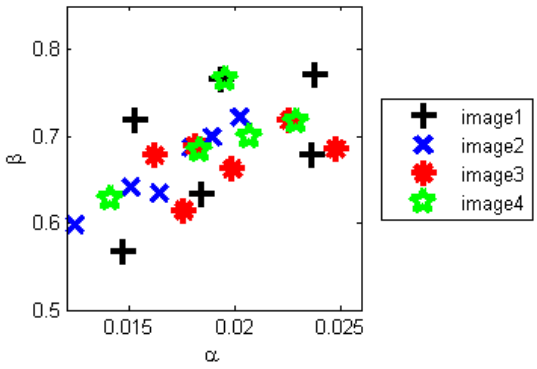


Figure III.12 – Gâteau aux pommes, éclairage haut droit et rotation d'angle θ. Les images ont été normalisées en niveaux de gris avant de les décomposer au moyen d'une décomposition en ondelettes stationnaire 2D. Affichage des paramètres d'échelle α et de forme β d'une GGD pour la sous-bande H1. Les plus noirs (respectivement les croix bleues, les flocons rouges et les étoiles vertes) représentent des vecteurs de paramètres de GGD (α, β) estimés au sens du maximum de vraisemblance sur une sous-bande H1 d'une image ayant subit une rotation d'angle $rs = 0°$ (respectivement 60°, 120° et 180°)

La figure III.12 présente les vecteurs paramétriques $\theta_{i,s,o}$ estimés au sens du maximum de vraisemblance sur la première sous-bande. Les vecteurs $\hat{\theta}_{1,s,o}$ (respectivement $\hat{\theta}_{2,s,o}$, $\hat{\theta}_{3,s,o}$ et $\hat{\theta}_{4,s,o}$), estimés sur l'image III.8.(a) (respectivement sur les images III.8.(b), III.8.(c) et III.8.(d)), sont représentés dans le plan (α, β) par des signes plus noirs (respectivement des croix bleues, des flocons rouges et des étoiles vertes). Dans cet exemple, l'objet d'intérêt a subit une rotation entre chaque prise de la photo. Du point de vue des descripteurs, ils montrent tous une même diffusion autour d'un point central $(0,02; 0,7)$. Par conséquent la rotation n'a, dans cet exemple particulier, aucune conséquence sur les vecteurs paramétriques estimés. Remarquez que pour des textures avec un contenu fortement orienté, la distribution de la variable aléatoire $x_{s,o}$ subit plus de changements.

La figure III.13 présente les vecteurs paramétriques $\hat{\theta}_{i,s,o}$ estimés au sens du maximum de vraisemblance sur la première sous-bande de patchs extraits des images III.10.(a) à (d). Les vecteurs $\hat{\theta}_{1,s,o}$ (respectivement $\hat{\theta}_{2,s,o}$, $\hat{\theta}_{3,s,o}$ et $\hat{\theta}_{4,s,o}$), estimés sur l'image III.10.(a) (respectivement sur les images III.10.(b), III.10.(c) et III.10.(d)), sont représentés dans le plan (α, β) par des plus noirs (respectivement des croix bleues, des flocons rouges et des étoiles vertes). Dans cet exemple, l'objet d'intérêt est pris en photo à plusieurs hauteurs différentes, les images sont « rognées » pour n'avoir que le contenu textural. Une variabilité importante est observée dans la figure, avec une localisation différente suivant l'image. Les images III.10.(b) et III.10.(d) sont acquises à hauteur moyenne (l'image III.10.(a) étant acquise en position haute), et les vecteurs paramétriques estimés se situent en haut à gauche de la figure III.13 (voir les croix bleues et les étoiles vertes). Les vecteurs $\hat{\theta}_{3,s,o}$, représentés par des flocons rouges, se situent

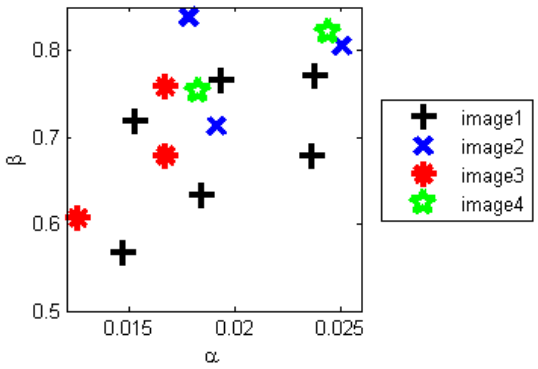


Figure III.13 – Gâteau aux pommes, éclairage haut droit et modification du point de vue de l'appareil photo. Les images ont été normalisées en niveaux de gris avant de les décomposer au moyen d'une décomposition en ondelettes stationnaire 2D. Affichage des paramètres d'échelle α et de forme β d'une GGD pour la sous-bande H1. Les plus noirs (respectivement les croix bleues, les flocons rouges et les étoiles vertes) représentent des vecteurs de paramètres de GGD (α, β) estimés au sens du maximum de vraisemblance sur sous-bande H1 d'une image prise avec un appareil photo en position haute (respectivement moyenne, basse et moyenne droite)

dans la partie inférieure gauche de la figure III.13. Les vecteurs $\hat{\theta}_{3,s,o}$ proviennent d'une image acquise en position basse. Par cet exemple, le point de vue lors de l'acquisition semble avoir un impact sur les paramètres estimés. Néanmoins cet exemple ne montre pas l'effet obtenu en rognant une image, jusqu'à obtenir quatre images de même taille avant la comparaison.

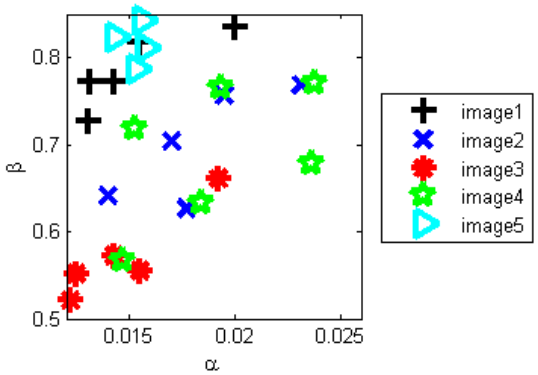


Figure III.14 – Gâteau aux pommes, différentes configurations d'éclairage et même orientation. Les images ont été normalisées en niveaux de gris avant de les décomposer au moyen d'une décomposition en ondelettes stationnaire 2D. Affichage des paramètres d'échelle α et de forme β d'une GGD pour la sous-bande H1. Les plus noirs (respectivement les croix bleues, les flocons rouges, les étoiles vertes et les triangles cyan) représentent des vecteurs de paramètres de GGD (α, β) estimés au sens du maximum de vraisemblance sur sous-bande H1 d'une image dans lequel l'éclairage est sur la gauche de l'objet (respectivement gauche haute, haute, droite haute et droite)

La figure III.14 représente les vecteurs paramétriques $\theta_{i,s,o}$ estimés au sens du maximum de vraisemblance sur la première sous-bande de patchs extraits des images III.11.(a) à (e). Les vecteurs $\theta_{1,s,o}$ (respectivement $\theta_{2,s,o}$, $\theta_{3,s,o}$, $\theta_{4,s,o}$ et $\theta_{5,s,o}$), estimés sur l'image III.11.(a) (respectivement sur les images III.11.(b), III.11.(c), III.11.(d) et III.11.(e)), sont représentés dans le plan (α, β) par des plus noirs (respec-

tivement des croix bleues, des flocons rouges, des étoiles vertes et des triangles cyan). Dans cet exemple, l'objet d'intérêt est pris en photo avec, pour chaque prise, un éclairage de la scène différent. La figure III.14 montre que la modification de l'éclairage modifie sensiblement la localisation des vecteurs paramétriques dans l'espace $(\alpha;\beta)$. Lorsque l'objet est éclairé par haut dessus III.11.(c) l'image présente très peu de zones d'ombres ; les vecteurs paramétriques, représentés par des flocons rouges, se situent dans la partie inférieure gauche. A l'opposé, lorsque la source lumineuse se trouve à l'horizontale à gauche (voir figure III.11.(a), respectivement à droite voir figure III.11.(e)) les ombres sont très présentes dans l'image. Les vecteurs $\theta_{1,s,o}$ (resp. $\theta_{5,s,o}$) correspondant à une source lumineuse qui se trouve à l'horizontale à gauche (respectivement à droite), sont représentés par des signes plus noir (resp. des triangles cyan). Les vecteurs se situent dans la partie supérieure de la figure III.14. Remarquons de plus que les signes plus noir recouvrent un espace disjoint de l'espace recouvert par les triangles cyan. Cela s'interprète ainsi : plus les ombres sont présentes, plus le paramètre β est élevé indiquant une distribution avec des queues moins importantes. Mais ce type d'interprétation ne peut pas être généralisé étant donné que seul une projection du descripteur paramétrique de l'image est observée.

Nous avons présenté le problème de la diversité intra-classe pour promouvoir la classification avec plusieurs clusters par classe [65, 67, 75].

Cette section s'est appuyée sur des images présentant une diversité simple (rotation, point de vue et éclairage de la scène) pour montrer que la diversité intra-classe peut avoir de fortes conséquences sur l'espace des descripteurs paramétriques. Par simplicité, les descripteurs ont été projetés sur un sous-espace de dimension 2 (le plan $(\alpha;\beta)$) afin d'observer les conséquences de la diversité intra-classe. La combinaison des exemples de diversité simples entre eux ou avec d'autres sources de diversité (différence d'échelle, perspective) doivent augmenter l'écart présent entre les descripteurs paramétriques. Dans un but de classification d'images texturées, la diversité peut amener des faux positifs et des erreurs de classification. Parmi les méthodes proposées par la « littérature » pour réduire l'impact de la diversité, ce mémoire se focalise sur 2 méthodes : l'algorithme des K-Moyennes et le mélange de gaussiennes sur l'espace des descripteurs. Le point commun de ces méthodes est de représenter l'ensemble des descripteurs paramétriques d'une même classe par une collection de clusters. Nous proposons dans la section suivante des algorithmes de clustering sur variété afin d'apprendre la forme de ces clusters.

4 Proposition pour la construction d'un dictionnaire intrinsèque

La diversité intra-classe impose aux descripteurs (paramétriques ou non paramétriques) une représentation partitionnée en clusters. Le but est alors de sélectionner une algorithme de clustering afin d'estimer les clusters. Le principe des algorithmes de clustering est présenté dans une première sous-section. Un état de l'art de la géométrie de l'espace des descripteurs paramétriques est présenté en second. Enfin, la troisième sous-section nous vous présentons le mélange de gaussiennes concentrées.

4.1 Clustering de l'espace des descripteurs locaux

4.1.1 Définition du clustering

Un algorithme de clustering est un algorithme de classification non supervisé, autrement dit les véritables clusters ne sont pas connus *a priori* et doivent être estimés par l'algorithme. Des algorithmes de clustering avancés estiment même le nombre N_{cl} de clusters présents dans la population proposée. Par exemple, les critères d'information d'Akaike [76, 77] ou bayésiens [78] pourraient être envisagés dans l'optique d'établir automatiquement le nombre de clusters N_{cl}. Sans perte de généralité, le nombre de clusters N_{cl} est fixé *a priori* et sera égal entre chaque classe.

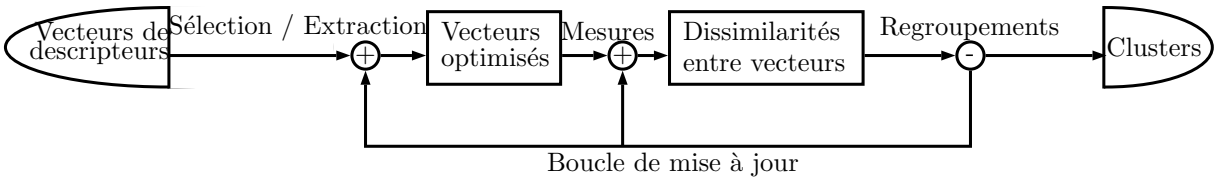


Figure III.15 – Étapes du clustering [3]

La figure III.15 présente les trois étapes principales du clustering selon Jain et Dubes [3]. Premièrement, soit certains des descripteurs locaux sont sélectionnés soit une transformation homéomorphe est appliquée sur les descripteurs locaux afin d'obtenir des vecteurs optimisés. Les vecteurs optimisés montrent la plus grande dissimilarité possible, tout en assurant une complexité calculatoire faible pour la seconde étape. La deuxième étape, justement, consiste à mesurer la dissimilarité entre des paires de descripteurs optimisés. Par exemple, la distance euclidienne est une mesure de dissimilarité possible entre des descripteurs optimisés [3]. Nous montrerons également que la divergence de Jeffrey peut être utilisée sous certaines conditions sur la troisième étape. Enfin, les descripteurs optimisés sont regroupés en clusters. Cette étape de regroupement peut être réalisée par de nombreux algorithmes différents.

La communauté utilise de nombreux qualificatifs pour les algorithmes de clustering et ses résultats.

Le résultat du clustering peut être « dur », ce qui signifie que les vecteurs sont partitionnés en groupes. Le résultat peut être « diffus », ce qui signifie que chaque vecteur appartient à tous les clusters à un certain degré. Les algorithmes sont ordonnés sous des termes tel que clustering hiérarchique, clustering partitionné, clustering probabiliste et clustering basé sur des graphes. Prenons l'ensemble du clustering hiérarchique : chaque vecteur de descripteurs descend le long d'un arbre de décision jusqu'à être associé à une feuille de l'arbre qui est un cluster. Quelque soit l'algorithme, il est important de caractériser un cluster sans utiliser la totalité des vecteurs qui lui sont associés.

4.1.2 Application aux images texturées

Cette partie est dédiée à l'adaptation de la méthode de « Sac de Mots Visuels » (SMV) aux vecteurs paramétriques. Sans perte de généralité, les descripteurs locaux sont utilisés explicitement comme descripteurs optimisés. Dès lors, il est important de comprendre l'espace des descripteurs paramétriques et des métriques qui régissent cet espace. Dans une seconde sous-section, nous introduisons la notion de variété riemannienne avec la distance associée. En résumé, la divergence de Jeffrey qui est une approximation de la distance riemannienne, est utilisée en pratique pour mesurer la dissimilarité entre deux descripteurs locaux. Nos deux hypothèses *a priori* décrites de cette façon font que l'algorithme de clustering n'utilisera pas de boucle de mise à jour comme présenté dans la figure III.15.

Parmi les méthodes de clustering existantes dans la littérature nous nous focaliserons sur une méthode probabiliste : le mélange de gaussiennes. Une distribution gaussienne est paramétrée par une position μ et une variance σ^2 et, comme nous parlons de mélange, ce sont N_{cl} couples de paramètres (μ_i, σ_i^2) qui seront estimés pour une classe de texture.

4.1.3 Exemple de clustering : les K-moyennes

Considérons un cadre simplifié : toutes les variances σ_i^2 sont égales. Par conséquent, le mélange de gaussiennes concentrées est une méthode de clustering qui se rapproche d'une méthode partitionnée plus connue sous le nom de K-Moyennes (voir l'algorithme III.4). Soient K le nombre de descripteurs locaux pour la classe d'images texturées et N_{cl} le nombre de clusters fixé *a priori*. Le symbole θ_k représente un descripteur local identifié par l'entier $k = 1, \ldots, K$. Le texton du cluster numéro $i = 1, \ldots, N_{cl}$ est noté $\bar{\theta}_i$.

L'association du descripteur local θ_k à un cluster i se fait par le symbole i_k. Le symbole de Kronecker $\delta(i_k, i)$ est une fonction renvoyant 0 presque partout et 1 seulement si $i_k = i$. N_i représente le nombre de

Algorithme III.4 : Pseudo-code pour un algorithme de K-moyennes

Données : Un ensemble de K descripteurs locaux $(\theta_k)_{k=1}^{K}$ et N_{cl} le nombre de clusters
Résultat : Un ensemble de N_{cl} barycentres $\left(\bar{\theta}_i\right)_{i=1}^{N_{cl}}$

Initialisation des N_{cl} barycentres $\left(\bar{\theta}_i\right)_{i=1}^{N_{cl}}$;
Calcul de l'inertie intra cluster I_{intra} ;
répéter
  $I_{\text{old}} \leftarrow I_{\text{intra}}$;
  pour chaque *descripteur local* θ_k ; /* Assignation */
  faire
    $i_k \leftarrow \underset{i}{argmin}\, m(\bar{\theta}_i \| \theta_k)$;
  fin
  pour chaque *cluster* i ; /* Mise à jour des barycentres */
  faire
    $N_i \leftarrow \sum_{k=1}^{K} \delta(i_k, i)$;
    $\bar{\theta}_i \leftarrow \underset{\theta}{argmin} \left\{ \frac{1}{N_i} \sum_{k=1}^{K} m(\theta \| \theta_k) \delta(i_k, i) \right\}$;
  fin
  $I_{\text{intra}} \leftarrow \sum_{i=1}^{N_{cl}} \frac{1}{N_i} \sum_{k=1}^{K} m(\bar{\theta}_i \| \theta_k) \delta(i_k, i)$;
jusqu'à $I_{old} \geq I_{intra}$;

descripteurs locaux dans le cluster numéro i. Enfin, $m(\cdot \parallel \cdot)$ représente la mesure de dissimilarité entre deux descripteurs paramétriques qui peut ne pas être symétrique.

Cette section continue avec deux sous-sections. La première sous-section présente la variété riemannienne, la distance riemannienne et la divergence de Jeffrey. La seconde sous-section présente l'algorithme de clustering utilisé : le mélange de gaussiennes concentrées.

4.2 Géométrie intrinsèque à l'espace des descripteurs

La divergence de Jeffrey est une proposition de la littérature associée à la divergence de Kullback-Leibler tout en étant symétrique contrairement à cette dernière. L'espace de paramètres Θ décrit une variété riemannienne dont la matrice d'information de Fisher $G(\theta)$ [79] définit la géométrie. La matrice d'information de Fisher agit comme une métrique sur une variété riemannienne Θ. Cela signifie que l'espace Θ est localement continu et dérivable, il existe un système de cartes couvrant toute la surface de l'espace Θ. Les rares intersections des cartes sont soumises à des fonctions de transition. En termes simples, s'il existe un plan tangent en $\theta \in \Theta$, alors le plan tangent en $\theta' \in \Theta$ est exprimé comme une image du premier plan tangent après une translation et une rotation. La métrique G Sur une variété riemannienne est calculée en un point $\theta \in \Theta$ et correspond à la géométrie dans l'espace tangent à Θ en θ. Les distances qui en découlent correspondent à la longueur de la courbe la plus courte entre deux

vecteurs θ, θ'. Soit géodésique γ le nom de la courbe la plus courte de Θ reliant θ à θ'.

Définition 1. Soit ∇ le symbole du gradient d'une fonction, comme γ est une géodésique sur Θ elle est également de dimension d. Soient $\boldsymbol{\gamma_i}$ la i-ème composante de γ et $\nabla\gamma(t) = (\partial\boldsymbol{\gamma_i}(t)/\partial t)_{i=1}^{d}$ un vecteur colonne de dimension d nommé gradient, $(\nabla\gamma(t))^T$ la transposition d'un vecteur.

La distance riemannienne [80] (distance de Rao) avec la matrice d'information de Fisher G est définie par :

$$\mathrm{GD}(\theta, \theta') = \int_0^1 \sqrt{(\nabla\gamma(t))^T G(\gamma(t))\nabla\gamma(t)}.dt \tag{III.21}$$

Dans un premier paragraphe nous présenterons la matrice d'information de Fisher gouvernant la métrique riemanienne de Θ. Dans un second paragraphe, nous montrons que la divergence de Jeffrey, la divergence de Kullback-Leibler et le carré de la distance riemannienne coïncident localement.

4.2.1 Description de l'espace des descripteurs paramétrique : notion de variété riemannienne

Soit $\boldsymbol{\theta_i}$ la i-ème composante du vecteur de paramètres θ.

Définition 2 (Matrice d'information de Fisher)**.** Soient $\mathcal{P}$ un modèle paramétrique sur les données x, Θ l'ensemble de tous les vecteurs de paramètres de $\mathcal{P}$, $p(x \mid \theta)$ une vraisemblance suivant $\mathcal{P}$.

L'élément en position (i, j) de la matrice d'information de Fisher se définit [81] par l'opposé de l'espérance de la dérivée seconde de la log-vraisemblance :

$$g_{i,j}(\theta) = -\int_{\mathcal{X}} p(x \mid \theta) \frac{\partial^2 \log\{p(x \mid \theta)\}}{\partial\boldsymbol{\theta_i}\partial\boldsymbol{\theta_j}}.dx$$

Si la log-vraisemblance n'est pas deux fois dérivable, nous pouvons exprimer la matrice d'information de Fisher par l'espérance du produit des dérivées partielles d'ordre 1 de la log-vraisemblance :

$$g_{i,j}(\theta) = \int_{\mathcal{X}} p(x \mid \theta) \frac{\partial \log\{p(x \mid \theta)\}}{\partial\boldsymbol{\theta_i}} \frac{\partial \log\{p(x \mid \theta)\}}{\partial\boldsymbol{\theta_j}}.dx \tag{III.22}$$

La matrice d'information de Fisher décrit les dépendances entre les composantes du vecteur de paramètres θ en termes d'information, ce qui signifie que pour certaines vraisemblances $p(x \mid \theta)$, comme les lois à paramètre de puissance, les composantes des paramètres présentent des dépendances qui doivent être prises en compte par la loi *a priori* informative $p(\theta)$.

Théorème 1. *Soient $\mathcal{P}$ un modèle paramétrique sur les données x, Θ l'ensemble de tous les vecteurs*

de paramètres de $\mathcal{P}$, $p(x \mid \theta)$ une vraisemblance suivant $\mathcal{P}$ et $G(\theta)$ la matrice d'information de Fisher associée.

Alors Θ est une variété riemannienne avec $G(\theta)$ comme géométrie [82].

4.2.2 Formule explicite de la divergence de Jeffrey

Dans ce mémoire, les modèles stochastiques choisis et présentés sont la GGD, la GΓD et le modèle SIRV avec multiplieur Weibull. Le modèle SIRV avec multiplieur Weibull n'admet pas de forme explicite pour sa densité de probabilité. La log-vraisemblance n'existe pas encore et la dérivée d'ordre 2, la matrice d'information de Fisher, n'admet pas de forme explicite. La formule explicite de la distance riemannienne ne peut pas être définie pour le modèle SIRV avec multiplieur Weibull sans utiliser des approximations. Malheureusement, il n'existe pas de formule explicite pour la distance riemannienne pour les distributions GGD, GΓD ou modèle SIRV avec multiplieur Weibull. Les modèles SIRV, la distribution GGD ou la distribution GΓD utilisées dans ce mémoire ne présentent pas de forme explicite pour la distance de Rao (GD). Entre une distance GD intrinsèque sans forme explicite et une distance euclidienne extrinsèque, un jeu de mesures de dissimilarités peux être envisagé. La communauté de la géométrie de l'information propose des formules explicites comme des approximations de la distance GD. La divergence de Kullback-Leibler ou la divergence de Jeffrey sont deux exemples de ces approximations. Une divergence se démarque de la distance et de la métrique puisqu'une divergence n'est pas symétrique et ne vérifie pas l'inégalité triangulaire.

Définition 3. Soit Θ une variété riemannienne. Soient θ, θ_1 et θ_2 trois éléments de Θ. Soit D une fonction définie par

$$D(.\|.)\Theta^2 \longrightarrow \mathbb{R}^+.$$

vérifiant

1. Pour tout $\theta_1, \theta_2 \in \Theta$, nous avons $D(\theta_1\|\theta_2) \geq 0$;
2. Pour tout $\theta_1, \theta_2 \in \Theta$, nous avons $D(\theta_1\|\theta_2) = 0$ si et seulement si $\theta_1 = \theta_2$;
3. Pour tout $\theta \in \Theta$. Il existe une matrice semi-définie positive $M(\theta)$ tel que pour tout $d\theta$ petit tel que $\theta + d\theta \in \Theta$, nous avons
$$D(\theta + d\theta\|\theta) \overset{d\theta=0}{=} \frac{1}{2} d\theta^T M(\theta) d\theta$$

La fonction D est appelée divergence [83].

Dans la définition de la divergence, apparaît le symbole $\|$ entre les deux entrées, sa présence indique que la divergence n'est, généralement, pas symétrique. Le terme de divergence regroupe plusieurs mesures telles que la divergence de Kullback-Leibler, la divergence de Jeffrey, les divergences de Bregman mais également la distance euclidienne. Le choix de la divergence est relatif à la similarité de la divergence avec la distance de Rao et également l'existence de la forme explicite de la divergence.

Dans le cadre de l'application à la classification d'images texturées, l'ensemble des vraisemblances suivent un même modèle paramétrique. La proposition suivante démontre qu'une divergence sur l'espace des paramètres peut être déduite de la divergence de Kullback-Leibler.

Théorème 2. *Soient $\mathcal{P}$ un modèle paramétrique et Θ l'ensemble des vecteurs de paramètres associés. Soient θ_1 et θ_2 deux vecteurs de paramètres, $p(x \mid \theta_i)$ la densité de probabilité de vecteur de paramètres θ_i, pour $i = 1, 2$. La fonction KL définie par*

$$KL(\theta_1 \| \theta_2) = \int_{\mathcal{X}} p(x \mid \theta_1) \log \left\{ \frac{p(x \mid \theta_1)}{p(x \mid \theta_2)} \right\} .dx$$

est une divergence.

Lemme 3. *Soient $\mathcal{P}$ un modèle paramétrique et Θ l'ensemble des vecteurs de paramètres associés. Soient θ et θ' deux vecteurs de paramètres, $d\theta = \theta' - \theta$ le vecteur de déplacement entre θ et θ'. La fonction KL est localement équivalente avec la distance de Rao au carré :*

$$KL(\theta + d\theta \parallel \theta) \overset{d\theta=0}{=} \frac{1}{2}(GD(\theta, \theta + d\theta))^2 \tag{III.23}$$

KL est équivalente localement avec la distance de Rao au carré, mais elle n'est pas symétrique. Le caractère non symétrique résulte dans la définition de deux barycentres orientés et nécessite de définir proprement un ordre dans l'espace Θ et de mettre constamment le plus petit à gauche par exemple. Mais il existe des divergences pouvant être vues comme une divergence de Kullback-Leibler symétrique, par exemple la divergence de Jeffrey (J) est une forme symétrisée de la divergence de Kullback-Leibler.

Définition 4. Soit $\mathcal{P}$ l'ensemble de tous les modèles stochastiques, vu comme une variété riemannienne. Soient p_1 et p_2 deux éléments de $\mathcal{P}$. La divergence de Jeffrey de p_1 et de p_2 est définie par :

$$\mathrm{JD}(p_1, p_2) = \mathrm{KLD}(p_1 \parallel p_2) + \mathrm{KLD}(p_2 \parallel p_1) \tag{III.24}$$

La divergence de Jeffrey défini également une divergence dans l'espace des paramètres Θ, même si le

point important reste l'équivalence locale à la distance de Rao avec une propriété de symétrie absente de la divergence KL.

Théorème 4. *Soient $\mathcal{P}$ un modèle paramétrique et Θ l'ensemble des vecteurs de paramètres associés. Soient θ_1 et θ_2 deux vecteurs de paramètres, $p(x \mid \theta_i)$ la densité de probabilité de vecteur de paramètre θ_i, pour $i = 1, 2$. La fonction J définie par*

$$J(\theta_1, \theta_2) = KL(\theta_1 \| \theta_2) + KL(\theta_2 \| \theta_1)$$

est une divergence.

Lemme 5. *Soient $\mathcal{P}$ un modèle paramétrique et Θ l'ensemble des vecteurs de paramètres associés. Soient θ et θ' deux vecteurs de paramètres, $d\theta = \theta' - \theta$ le vecteur de déplacement entre θ et θ'. La divergence J hérite de la KL l'équivalence locale avec la distance de Rao au carré :*

$$J(\theta, \theta + d\theta) \overset{d\theta=0}{=} (GD(\theta, \theta + d\theta))^2 \tag{III.25}$$

La divergence de Jeffrey est donc une approximation possible de la distance de Rao (au carré) qui est utilisée pour comparer deux distributions suivant le même modèle paramétrique. Elle a, en outre, l'avantage d'avoir une forme explicite en fonction des paramètres. Les densités paramétriques qui nous intéressent et pour lesquelles la JD a une forme explicite sont la GGD, la GΓD ou la distribution Weibull.

Nous avons présenté la géométrie de l'information dès les premières publications [63–67, 75, 84]. La section suivante présente comment la divergence est utilisée pour définir une loi *a priori* sur la variété des vecteurs de paramètres Θ.

4.3 Mélange de gaussiennes concentrées

Ce paragraphe présente une des contributions majeures de notre travail. En effet, dans le cadre de caractérisation d'images texturées, nous proposons un modèle bayésien permettant de prendre en compte la diversité intra-classe. Pour cela, nous définissons une loi *a priori* qui modélise des échantillons d'une même classe dans l'espace paramétrique. Une telle loi est donc à valeurs dans l'espace propre des paramètres qui forme une variété riemannienne. Elle sera donc définie avec les outils de la géométrie différentielle. Pour cela nous proposerons la distribution gaussienne concentrée et son extension au cas des mélanges. Suite à la définition de ces lois utilisées comme *a priori* dans un modèle hiérarchique bayésien,

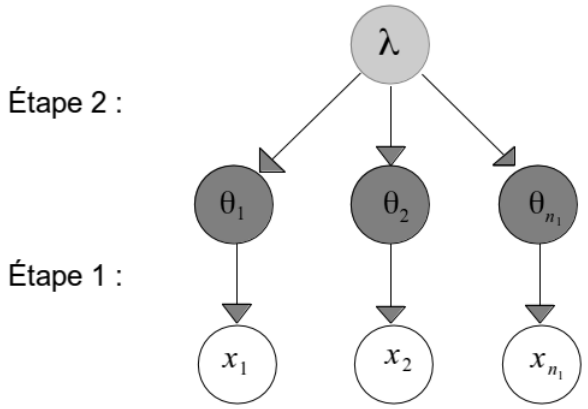


Figure III.16 – Modèle hiérarchique associé à la classification bayésienne

nous proposerons un schéma d'estimation des hyper paramètres les caractérisant. Conformément au modèle hiérarchique, différents modèles de vraisemblance seront proposés dans le chapitre suivant. Nous nous focaliserons sur la GGD, la distribution Weibull ou le modèle SIRV pour la prise en compte de dépendances spatiales de type intra-bande.

4.3.1 Modèle hiérarchique

Soit $\mathcal{P}_p$ une famille de modèles paramétriques de réalisations $(\theta_k)_{k=1}^K$, $p(\theta_k \mid \lambda)$ définit alors la densité de probabilité associée au modèle paramétrique $\mathcal{P}_p$. L'espace $\mathcal{P}_p$ est une variété riemannienne équipée de la métrique de Fisher-Rao (III.22).

Afin de prendre en compte un tel modèle, nous proposons d'utiliser, comme le montre la figure III.16, un modèle hiérarchique bayésien qui se découpe en deux niveaux. Le premier niveau caractérise la loi conditionnelle à $(\theta_1, \ldots, \theta_K)$ et λ, le vecteur x_n (c'est-à-dire les textures) est indépendant, identiquement distribué et de densité $p(x_n \mid \theta_k, \lambda)$ pour $k = 1, \ldots, K$ appartenant à la famille paramétrique :

$$\{p(x \mid \theta, \lambda) | \theta \in \Theta \text{ et } \lambda \in \Lambda\}$$

Le second niveau, conditionnellement à λ, permet de définir la loi des vecteurs θ_k indépendants, identiquement distribués et de densité $p(\theta_k \mid \lambda)$ appartenant à la famille

$$\{p(\theta \mid \lambda) | \lambda \in \Lambda\}$$

Soit d la dimension de l'espace des paramètres Θ et m la dimension de l'espace des hyper paramètres Λ.

Le modèle bayésien proposé permet d'associer aux classes d'images texturées des vecteurs de paramètres sur la variété Θ. A titre d'illustration, la figure III.17 représente le concept de classe sur la variété paramétrique Θ. Chacune des classes est symbolisée par une ellipse désignant la fluctuation intra-classe. Dans le paragraphe suivant, nous proposons un premier modèle permettant de représenter la forme des classes de texture dans la variété Θ au moyen des hyper paramètres que sont les statistiques d'ordre 1 et d'ordre 2, représentées respectivement par le barycentre $\bar{\theta}$ et la variance σ^2. En fait, nous généraliserons ce concept en introduisant une loi mélange comme dans le cas des méthodes SMV. Chaque barycentre sur la variété reprend alors un mot du dictionnaire.

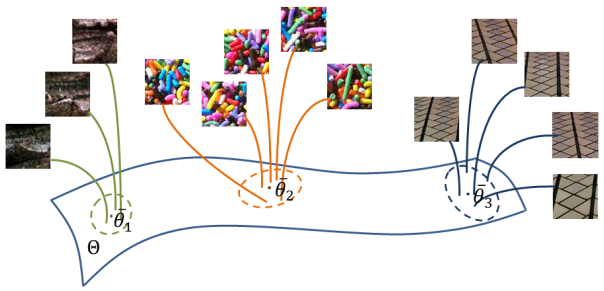


Figure III.17 – Représentation des classes de texture sur la variété

4.3.2 Gaussienne concentrée

Nous souhaitons utiliser un mélange de gaussiennes sur une variété riemannienne comme loi *a priori*. Pour bien discuter de distribution gaussienne sur une variété, nous mettrons de côté la loi mélange pour se focaliser sur une unique composante gaussienne. Nous parlerons du mélange dans la section suivante.

Dans le contexte applicatif de la classification de signaux et d'images, les vecteurs de paramètres θ se répartissent sur la variété Θ. Soit $(\theta_k)_{k=1}^K$ un jeu de vecteurs de paramètres d'une même classe dont la distribution est paramétrée par une statistique d'ordre 1 et une statistique d'ordre 2 : respectivement le barycentre $\bar{\theta}$ et la variance σ^2 des vecteurs de paramètres au sein de la variété.

Parmi les modèles stochastiques possibles pour $p(\theta \mid \lambda)$, nous nous focaliserons dans un premier temps sur la distribution gaussienne sur une variété puis à partir de cette définition, nous généraliserons au cas du mélange de lois gaussiennes pour son caractère universel.

Définition. Peu de travaux se sont intéressés à ce problème de définition des lois sur variétés. Concernant l'extension de la distribution gaussienne au cas des variété, nous citerons les travaux de Pennec [85]

qui définissent la distribution gaussienne dans le cas de la variété des matrices définies positive (SPD). Dans le cas de la variété SPD, la distance géodésique existe. La définition de la distribution gaussienne est alors donnée par

$$p(\theta \mid \bar{\theta}, C) = Z \exp\left\{-\frac{1}{2}(\gamma_{\bar{\theta},\theta}(1))^T C \gamma_{\bar{\theta},\theta}(1)\right\} \tag{III.26}$$

avec une géodésique $\gamma_{\bar{\theta},\theta}$ définie sur la variété Θ, C_i une matrice de concentration et Z une constante de normalisation. Z est obtenue comme l'inverse de l'intégrale de la distribution gaussienne sur tout l'espace tangent $\mathcal{T}_{\bar{\theta}}(\Theta)$ à la variété Θ en $\bar{\theta}$:

$$Z^{-1} = \int_{\mathcal{T}_{\bar{\theta}}(\Theta)} \exp\left\{-\frac{1}{2}(\gamma_{\bar{\theta},\theta}(1))^T C \gamma_{\bar{\theta},\theta}(1)\right\} . dLog_{\bar{\theta}}(\theta) \tag{III.27}$$

Nous nous inspirons de cette dernière pour définir une loi *a priori* informative et intrinsèque, dans le sens où la loi *a priori* dépend de la matrice d'information de Fisher $G(\theta)$ qui est aussi la métrique de l'espace.

Proposition 1. *Soient $\mathcal{P}_p$ un modèle paramétrique sur les données x, Θ l'ensemble de tous les vecteurs de paramètres de $\mathcal{P}_p$, $p(x \mid \theta)$ la loi de vraisemblance suivant $\mathcal{P}_p$ et $G(\theta)$ la matrice d'information de Fisher associée.*

Soient $\mathcal{T}_{\bar{\theta}}(\Theta)$ l'espace tangent à la variété Θ en $\bar{\theta}$, $Log_{\bar{\theta}}(\theta)$ la carte logarithmique qui associe à chaque paramètre θ un unique vecteur $\gamma'(\theta)$ dans l'espace tangent, $Exp_{\bar{\theta}}(\theta)$ la carte exponentielle qui est la fonction réciproque de $Log_{\bar{\theta}}(\theta)$.

Soient t un vecteur de l'espace tangent $\mathcal{T}_{\bar{\theta}}(\Theta)$, f une fonction définie sur $\mathcal{T}_{\bar{\theta}}(\Theta)$ et à valeurs dans $\mathbb{R}^+$. Le changement de variable entre une intégration sur l'espace tangent $\mathcal{T}_{\bar{\theta}}(\Theta)$ et une intégration sur la variété Θ fait apparaître un élément de volume riemannien $|G(\theta)|^{1/2}$. L'égalité explicite est donnée par :

$$\int_{\mathcal{T}_{\bar{\theta}}(\Theta)} f(Exp_{\bar{\theta}}(\gamma)) . dLog_{\bar{\theta}}(\theta) = \int_{\Theta} f(\theta)|G(\theta)|^{1/2} . d\theta$$

Définition 5. Soient $\mathcal{P}_p$ un modèle paramétrique sur les données x, Θ l'ensemble de tous les vecteurs de paramètres de $\mathcal{P}_p$, $p(x \mid \theta)$ la loi de vraisemblance suivant $\mathcal{P}_p$ et $G(\theta)$ la matrice d'information de Fisher associée. La loi *a priori* intrinsèque s'exprime en fonction du barycentre $\bar{\theta}$ et de la variance σ^2 comme suit :

$$p(\theta \mid \lambda) = Z \exp\left\{-\frac{1}{2\sigma^2} J(\theta, \bar{\theta})\right\} \tag{III.28}$$

avec l'hyper-paramètre $\lambda = (\bar{\theta}, \sigma^2)$, et Z la constante de normalisation.

Cette définition requiert le calcul de la constante de normalisation qui est donnée par la proposition 2. Nous nommerons distribution gaussienne concentrée la loi *a priori* intrinsèque.

Proposition 2. *Soit $p(\theta \mid \lambda)$ la loi* a priori *intrinsèque (III.28), alors la constante de normalisation vaut*

$$Z \simeq \frac{1}{\sigma^d (2\pi)^{d/2}} \tag{III.29}$$

Démonstration. La distance de Rao est donnée par (III.21) à condition de disposer de l'expression des géodésiques. Dans le cas des modèles paramétriques GGD, GΓD ou modèle SIRV avec marginale Weibull ce n'est malheureusement pas le cas. Aussi, nous sommes contraint de passer par une approximation. En effet, nous allons supposer ses géodésiques linéaires, ce qui est vrai asymptotiquement pour $\theta \to \bar{\theta}$. Dans ce cas, la loi *a priori* intrinsèque s'écrit :

$$p(\theta \mid \lambda) = Z \exp\left\{-\frac{1}{2\sigma^2}(\bar{\theta}-\theta)^T G(\bar{\theta})(\bar{\theta}-\theta)\right\}$$

en fixant la matrice de concentration $C = G(\bar{\theta})/\sigma^2$ nous retrouvons bien la loi gaussienne définie sur une variété proposée par Pennec [85] qui nous donne la formule de Z (III.27) :

$$Z^{-1} = \int_\Theta \exp\left\{-\frac{1}{2\sigma^2}(\bar{\theta}-\theta)^T G(\bar{\theta})(\bar{\theta}-\theta)\right\}.d\theta$$

Nous rappelons alors la formule de la loi gaussienne multivariée centrée et réduite :

$$\int_H (2\pi)^{-d/2} \exp\left\{-\frac{1}{2}(\bar{\eta}-\eta)^T(\bar{\eta}-\eta)\right\}.d\eta = 1$$

Nous effectuons un premier changement de variable $\tilde{\eta} = \eta/\sigma$:

$$\int_H (2\pi)^{-d/2}\sigma^{-d} \exp\left\{-\frac{1}{2\sigma^2}(\bar{\eta}-\eta)^T(\bar{\eta}-\eta)\right\}.d\eta = 1$$

qui est suivi par le changement de variable $A(\bar{\theta}-\theta) = \bar{\eta}-\eta$ avec $A^TA = G(\bar{\theta})$:

$$\begin{aligned} 1 &= \int_\Theta (2\pi)^{-d/2}\sigma^{-d}|A| \exp\left\{-\frac{1}{2\sigma^2}(\bar{\theta}-\theta)^T A^T A(\bar{\theta}-\theta)\right\}.d\theta \\ &= \int_\Theta (2\pi)^{-d/2}\sigma^{-d}|G(\bar{\theta})|^{1/2} \exp\left\{-\frac{1}{2\sigma^2}(\bar{\theta}-\theta)^T G(\bar{\theta})(\bar{\theta}-\theta)\right\}.d\theta \end{aligned}$$

En effet, la propriété de morphisme du déterminant permet d'écrire $|G(\bar{\theta})| = |A^T||A| = |A|^2$.

Nous concluons, $|G(\bar{\theta})|^{1/2}$ étant l'élement de volume riemannien, par :

$$Z \simeq \frac{1}{\sigma^d (2\pi)^{d/2}}$$

□

La loi *a priori* intrinsèque est définie au moyen de la géométrie de la variété Θ, qui ne doit pas être confondue avec la loi *a priori* de Jeffrey

$$p(\theta) \propto |G(\theta)|^{1/2}$$

La loi *a priori* de Jeffrey ne prend en compte que le déterminant et n'utilise pas toute la géométrie de l'espace Θ. La loi de Jeffrey est l'équivalent de la loi uniforme sur la variété. La gaussienne concentrée proposée est une loi *a priori* qui est intrinsèque à la variété Θ, la section suivante développe la règle de décision construite à partir de notre modèle de loi *a priori* intrinsèque.

Effectuer la décision sur les coefficients d'ondelettes $\mathcal{X}_i$. Soit $\mathcal{X}$ l'ensemble des coefficients d'ondelettes d'une classe, cela signifie que $\mathcal{X} = (x_n)_{n=1}^N$. La vraisemblance des coefficients par rapport au modèle paramétrique vrai s'écrit donc :

$$p(\mathcal{X} \mid \mathcal{H}) = \prod_{n=1}^{N} p(x_n \mid \mathcal{H})$$

avec $\mathcal{H}$ une variable aléatoire, appelée décision, indiquant l'appartenance à la classe. Dans ce mémoire, nous utilisons des modèles paramétriques avec θ le vecteur de paramètres. Par le théorème de Bayes, nous pouvons définir la vraissemblance marginalisée :

$$p(\mathcal{X} \mid \mathcal{H}) = \int_{\Theta} p(\mathcal{X}, \theta \mid \mathcal{H}).d\theta = \int_{\Theta} p(\mathcal{X} \mid \theta, \mathcal{H}) p(\theta \mid \mathcal{H}).d\theta$$

Sans perte de généralité, nous allons nous focaliser sur la gaussienne concentrée définie sur Θ. Soit λ le vecteur d'hyper paramètres - constitué de la position $\bar{\theta}$ et de la variance σ^2. L'estimateur empirique du vecteur d'hyper-paramètre qui maximise la vraisemblance aux coefficients est solution de l'équation :

$$\lambda_{\mathrm{MLE}} = arg \max_{\lambda} p(\mathcal{X} \mid \lambda)$$

Ce qui conduit à :

$$\lambda_{\text{MLE}} = arg\max_{\lambda} \int_{\mathcal{T}_{\bar{\theta}}(\Theta)} p(\mathcal{X} \mid \theta, \lambda) p(\theta \mid \lambda) d\text{Log}_{\bar{\theta}}(\theta)$$

et, sans perte de généralité, la distribution des coefficients est supposée indépendante des hyper paramètres λ :

$$\lambda_{\text{MLE}} = arg\max_{\lambda} \int_{\mathcal{T}_{\bar{\theta}}(\Theta)} p(\mathcal{X} \mid \theta) p(\theta \mid \lambda) . d\text{Log}_{\bar{\theta}}(\theta)$$

Le problème rencontré concerne l'évaluation de l'intégrale :

$$\int_{\mathcal{T}_{\bar{\theta}}(\Theta)} p(\mathcal{X} \mid \theta) p(\theta \mid \lambda) d\text{Log}_{\bar{\theta}}(\theta)$$

De nombreux auteurs se sont intéressés à ce problème. Différentes solutions ont été proposées parmi lesquelles nous trouvons les approches de types Monte-Carlo, variationnelle ou celles utilisant l'approximation de Laplace.

L'approximation de Laplace a été appliquée avec succès par de nombreux auteurs [86–88]. Sous certaines conditions de régularité, il a été démontré notamment par Kass et Steffy en 1989 du lien d'équivalence qui pouvait être fait avec les méthodes par échantillonnage par exemple. Plus récemment, Miyata en 2004 [89] a complété l'étude visant à démontrer la pertinence de l'approximation à l'ordre deux fondée sur le maximum *a posteriori*.

Dans cette partie du mémoire, nous allons proposer, sur la base des travaux de Miyata [89], une méthode d'estimation des hyper paramètres de notre modèle hiérarchique en exploitant l'approximation de Laplace dans le cas de l'*a priori* intrinsèque.

1) Rappel sur l'approximation de Laplace. L'obtention d'un estimateur des hyper paramètres λ passe par l'évaluation de l'intégrale

$$\int_{\mathcal{T}_{\bar{\theta}}(\Theta)} p(\mathcal{X} \mid \theta) p(\theta \mid \lambda) . d\text{Log}_{\bar{\theta}}(\theta)$$

Cette intégrale est aussi appelée la vraisemblance marginalisée. Nous introduisons l'approximation de Laplace de l'intégrale de la forme :

$$\int_{\mathcal{T}_{\bar{\theta}}(\Theta)} e^{-N h_N(\theta)} p(\theta \mid \lambda) . d\text{Log}_{\bar{\theta}}(\theta) \text{ avec } h_N(\theta) = -\frac{1}{N} \log \{p(\mathcal{X} \mid \theta)\}$$

Théorème 6. *Supposons que $\hat{\theta}$ est le mode asymptotique d'ordre N^{-1} de $h_N(\theta)$ et que la paire $(h_N(\theta), \hat{\theta})$*

vérifient les hypothèses de régularité nécessaires à l'approximation de Laplace. Alors il s'en suit que pour k suffisamment grand

$$\int_{\mathcal{T}_{\hat{\theta}}(\Theta)} e^{-Nh_N(\theta)} p(\theta \mid \lambda).dLog_{\hat{\theta}}\theta = (2\pi)^{d/2}|\Sigma|^{-1/2} p(\mathcal{X} \mid \hat{\theta}) p(\hat{\theta} \mid \lambda)|G(\hat{\theta})|^{1/2} \left(1 + \mathcal{O}(N^{-1})\right)$$

avec

$$\Sigma = \frac{1}{k}[\mathcal{H}h_N(\hat{\theta})]^{-1}$$

et

$$\mathcal{H}h_N(\theta) = \frac{\partial^2 h_N(\theta)}{\partial\theta\partial\theta^T} = \begin{pmatrix} \frac{\partial^2 h_N(\theta)}{\partial\theta_1^2} & \cdots & \frac{\partial^2 h_N(\theta)}{\partial\theta_1\partial\theta_d} \\ \vdots & \ddots & \vdots \\ & & \end{pmatrix}$$

Définition 6. $\hat{\theta}$ est le mode asymptotique d'ordre N^{-2} de $-h_N$ si $\hat{\theta}$ converge vers le mode exact de $-h_N$ et $\nabla h_N(\hat{\theta}) = \mathcal{O}\left(\frac{1}{N^2}\right)$.

Un autre point intéressant de l'étude de [89, 90] est de donner aussi une relation explicite entre l'estimateur du maximum de vraisemblance et du mode asymptotique.

Nous pouvons alors faire deux remarques. La première remarque est donnée par Heyde et Johnstone [91]. L'estimateur de vraisemblance (MLE) $\hat{\theta}_{\text{MLE}}$ de $p(\mathcal{X} \mid \theta)$ est le mode asymptotique d'ordre N^{-1} pour $-h_N$ car $\nabla h_N(\hat{\theta}_{\text{MLE}}) = \mathcal{O}(N^{-1})$ et $\hat{\theta}_{\text{MLE}}$ converge vers le mode exact de $-h_N(\theta)$ qui est le mode *a posteriori*, quand N tend vers l'infini.

Notre deuxième remarque porte sur une égalité. Soit $\hat{\theta}_{\text{MLE}}$ l'estimateur MLE de $p(\mathcal{X} \mid \theta)$ alors :

$$\hat{\theta} = \hat{\theta}_{\text{MLE}} - [\mathcal{H}h_N(\hat{\theta}_{\text{MLE}})]^{-1}\nabla h_N(\hat{\theta}_{\text{MLE}})$$

sachant que $\nabla h_N(\hat{\theta}) = \mathcal{O}(N^{-2})$.

2) *A priori* intrinsèque et approche bayésienne empirique. Dans ce paragraphe, nous reprenons l'estimateur empirique que nous confrontons à l'*a priori* intrinsèque. Sachant que

$$\lambda_{\text{MLE}} = arg\max_{\lambda} \int_{\mathcal{T}_{\hat{\theta}}(\Theta)} \prod_{n=1}^{N} p(x_n \mid \theta) p(\theta \mid \lambda) d\text{Log}_{\hat{\theta}}(\theta)$$

et considérant l'approximation de Laplace

$$\int_{\Theta} \prod_{n=1}^{N} p(x_n \mid \theta) p(\theta \mid \lambda).d\theta \sim (2\pi)^{d/2} |\Sigma|^{1/2} \prod_{n=1}^{N} p(x_n \mid \hat{\theta}) p(\hat{\theta} \mid \lambda) |G(\hat{\theta})|^{1/2}$$

Nous obtenons :

$$\lambda_{\text{MLE}} \simeq arg\max_{\lambda} (2\pi)^{d/2} |\Sigma(\hat{\theta})|^{1/2} |G(\hat{\theta})|^{1/2} \left(\prod_{n=1}^{N} p(x_n \mid \hat{\theta}) \right) p(\hat{\theta} \mid \lambda)$$

avec

$$\begin{aligned} \Sigma(\hat{\theta})^{-1} &= \frac{1}{N \mathcal{H} h_N(\hat{\theta})} \\ \mathcal{H} h_N(\hat{\theta}) &= \frac{\partial^2}{\partial\theta \partial\theta^T} \left[\frac{1}{N} \left(\sum_{n=1}^{N} \log\{p(x_n \mid \hat{\theta})\} \right) \right] \\ &\overset{N \to \inf}{\to} G(\hat{\theta}) \end{aligned}$$

En effet, la matrice Σ coïncide avec la définition finie de la matrice d'information de Fisher. Par simplification, le cadre asymptotique est considéré (menant à la définition continue de la matrice d'information de Fisher).

Asymptotiquement l'expression vaut :

$$\lambda_{\text{MLE}} \simeq arg\max_{\lambda} (2\pi)^{d/2} \left(\prod_{n=1}^{N} p(x_n \mid \hat{\theta}) \right) p(\hat{\theta} \mid \lambda)$$

alors :

$$\lambda_{\text{MLE}} \simeq arg\max_{\lambda} \left[\frac{d}{2} \log\{2\pi\} + \left(\sum_{n=1}^{N} \log\{p(x_n \mid \hat{\theta})\} \right) + \log\{p(\hat{\theta} \mid \lambda)\} \right]$$

En ne considérant que les termes dépendants de λ

$$\lambda_{\text{MLE}} \simeq arg\max_{\lambda} \Big[\log\{p(\hat{\theta} \mid \lambda)\} \Big]$$

Au moyen des calculs précédents, l'estimation des hyper paramètres λ_{MLE} qui maximise la vraisemblance avec les coefficients d'ondelettes coïncide asymptotiquement avec l'estimation au sens du maximum de vraisemblance sur la variété Θ. Du point de vue de l'implémentation, cette équivalence permet d'estimer les hyper paramètres après l'extraction des descripteurs paramétriques. Du point de vue théorique,

les hyper paramètres sont reliés aux coefficients de décomposition.

Le résultat obtenu montre aussi que du point de vue de la décision, le test d'appartenance à une classe de la donnée x revient à tester l'appartenance de ce paramètre associé à la classe via un test avec les hyper paramètres de la classe.

Estimation des hyper paramètres λ. Ce paragraphe revient sur l'estimation des paramètres de la gaussienne concentrée définie précédemment. L'estimation au sens du maximum de vraisemblance de la position revient à une minimisation d'une somme de divergences, autrement dit l'estimation de la position est invariante par rapport au paramètre de variance.

Proposition 3. *Soient $\mathcal{P}$ un modèle paramétrique sur les données x, Θ l'ensemble de tous les vecteurs de paramètres de $\mathcal{P}$, d la dimension de l'espace Θ, $p(x \mid \theta)$ une vraisemblance suivant $\mathcal{P}$ et $G(\theta)$ la matrice d'information de Fisher associée. Soient $\bar{\theta}$ le barycentre, σ^2 la variance et $\lambda = (\bar{\theta}, \sigma^2)$ le vecteur d'hyper paramètres. Soient K vecteurs de paramètres $\theta_k \in \Theta$ pour $k = 1, \ldots, K$. Soit $p(\theta \mid \lambda)$ une loi* a priori *intrinsèque, alors les estimateurs du maximum de vraisemblance des hyper paramètres sont donnés par :*

$$\begin{aligned}\hat{\bar{\theta}} &= arg\min_{\bar{\theta}\in\Theta} \frac{1}{K}\sum_{k=1}^{K} J(\theta_k, \bar{\theta}) \\ \hat{\sigma}^2 &= \frac{1}{dK}\sum_{k=1}^{K} J(\theta_k, \bar{\theta})\end{aligned}$$

Démonstration. Soient $\mathcal{P}$ un modèle paramétrique sur les données x, Θ l'ensemble de tous les vecteurs de paramètres de $\mathcal{P}$, d la dimension de l'espace Θ, $p(x \mid \theta)$ une vraisemblance suivant $\mathcal{P}$ et $G(\theta)$ la matrice d'information de Fisher associée. Soient $\bar{\theta}$ le barycentre, σ^2 la variance et $\lambda = (\bar{\theta}, \sigma^2)$ le vecteur d'hyper paramètres. Soient K vecteurs de paramètres $\theta_k \in \Theta$ pour $k = 1, \ldots, K$. Soit $p(\theta \mid \lambda)$ une loi *a priori* intrinsèque.

La vraisemblance s'écrit :

$$\mathcal{L}(\lambda \mid (\theta_k)_{k=1}^K) = \prod_{k=1}^{K} p(\theta_k \mid \lambda) = \frac{1}{(2\pi)^{dK/2}\sigma^{dK}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=1}^{K} J(\theta_k, \bar{\theta})\right\}$$

conduisant à la log-vraisemblance :

$$-\log\{\mathcal{L}(\lambda \mid (\theta_k)_{k=1}^K)\} = \frac{1}{2\sigma^2}\sum_{k=1}^{K} J(\theta_k, \bar{\theta}) + dK\log\{\sigma\} + \frac{dK}{2}\log\{2\pi\}$$

Le barycentre au sens du maximum de vraisemblance n'est pas explicite :

$$\begin{aligned}\hat{\bar{\theta}} &= \underset{\bar{\theta}\in\Theta}{arg\,\min} - \log\{\mathcal{L}(\lambda \mid (\theta_k)_{k=1}^K)\} \\ &= \underset{\bar{\theta}\in\Theta}{arg\,\min} \frac{1}{K}\sum_{k=1}^{K} J(\theta_k, \bar{\theta})\end{aligned}$$

Il reste maintenant à dériver la log-vraisemblance selon l'écart-type σ :

$$\frac{\partial}{\partial\sigma}\left[-\log\{\mathcal{L}(\lambda \mid (\theta_k)_{k=1}^K)\}\right] = \frac{dK}{\sigma} - \frac{1}{\sigma^3}\sum_{k=1}^{K} J(\theta_k, \bar{\theta})$$

et annuler cette dérivée conduit à écrire

$$\hat{\sigma}^2 = \frac{1}{dK}\sum_{k=1}^{K} J(\theta_k, \bar{\theta})$$

□

La discussion pour l'existence et l'unicité de ces deux hyper paramètres est conditionnelle au choix du modèle paramétrique effectué au départ. De plus les solutions pour l'estimation du paramètre de position nécessitent l'attention particulière du chapitre suivant. Dans cette section nous venons de présenter la distribution gaussienne concentrée sur Θ. Nous pouvons évoquer les travaux de Schutz et al. sur le sujet [84] et les travaux de Said et al. sur la construction de distributions gaussiennes sur la variété riemannienne des paramètres de gaussienne avec la distance riemannienne [92]. Nous allons maintenant nous intéresser au cas du mélange de gaussiennes concentrées.

4.3.3 Mélange de gaussiennes sur variétés

Pour la classification d'images texturées avec descripteurs paramétriques, nous proposons une approche bayésienne. Nous proposons la loi mélange de gaussiennes concentrées comme loi *a priori*.

Densité de probabilité. Soit N_{cl} le nombre de clusters. Par définition une loi mélange se formule à partir d'un jeu de densités de probabilités p_i et de poids de la densité w_i pour tout $i = 1, \ldots, N_{cl}$.

$$p(\theta \mid w_i, p_i, i = 1, \ldots, N_{cl}) = \sum_{i=1}^{N_{cl}} w_i p_i(\theta)$$

Sans perdre en généralité, les densités de probabilités marginales p_i sont supposées indépendantes, de distribution gaussienne concentrée p_Θ sur la variété stochastique Θ. Les hyper paramètres utilisés par le cluster numéro i sont notés $\bar{\theta}_i$ pour le paramètre de position et σ_i^2 pour le paramètre de variance. Voici la définition de la loi mélange de gaussiennes concentrées :

$$p(\theta \mid \left(\bar{\theta}_i, \sigma_i^2\right)_{i=1}^{N_{cl}}) = \sum_{i=1}^{N_{cl}} w_i p_\Theta(\theta \mid \bar{\theta}_i, \sigma_i^2) \quad \text{(III.30)}$$

$$= \sum_{i=1}^{N_{cl}} w_i \frac{1}{\sigma_i^d (2\pi)^{d/2}} \exp\left\{-\frac{1}{2\sigma_i^2} J(\theta, \bar{\theta}_i)\right\}$$

$$= \frac{1}{N_{cl}(2\pi)^{d/2}} \sum_{i=1}^{N_{cl}} \frac{1}{\sigma_i^d} \exp\left\{-\frac{1}{2\sigma_i^2} J(\theta, \bar{\theta}_i)\right\} \quad \text{(III.31)}$$

Estimation des hyper paramètres L'estimation des paramètres d'une loi mélange est généralement fait au moyen d'un algorithme « Espérance-Maximisation » (EM). L'algorithme EM est conçu pour maximiser la vraisemblance complète attendue pour la loi mélange. Cet algorithme est principalement utilisé pour estimer les paramètres d'un loi mélange de gaussiennes et quelques variantes ont été développées pour des lois mélanges de distributions plus originales comme les gaussiennes généralisées [47, 57] ou les distributions de Laplace [93]. Une extension de l'algorithme EM permet d'utiliser toutes les distributions issue d'une famille exponentielle pour la loi mélange : l'algorithme de clustering selon Bregman [94]. Plus récemment, l'estimation au sens du maximum des k vraisemblances (K-MLE) est proposée par Nielsen [95] : cet algorithme repose sur l'estimation au sens du maximum de vraisemblance de la distribution issue de la loi mélange. De plus une mise à jour similaire à l'algorithme de type K-moyennes est utilisé pour maximiser la vraisemblance complète. Cet algorithme est conçu pour des lois mélanges de distributions issues de famille exponentielle dont la distribution gaussienne concentrée ne fait pas partie.

Dans ce qui suit, nous allons étendre l'algorithme k-MLE au cas de la loi mélange de gaussiennes concentrées. Étant donné les poids w_i de la loi mélange, l'algorithme de clustering dur nommé K-MLE assigne de manière itérative chacun des descripteurs locaux θ_k au cluster le plus vraisemblable puis met à jour les composantes du modèle avec l'estimateur au sens du maximum de vraisemblance. Après cette étape, similaire à l'algorithme des k-moyennes, les poids w_i sont mis à jour. L'algorithme boucle jusqu'à

obtenir le maximum de la log-vraisemblance complète [95].

$$\begin{aligned}\mathcal{L}(\{\theta_k, i_k\}_{k=1}^{K} \ldots | w, \lambda) &= \frac{1}{K}\sum_{k=1}^{K} \log \prod_{i=1}^{N_{cl}} (w_i p_\Theta(\theta_k|\lambda_i))^{\delta_i(i_k)} \\ &= \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{N_{cl}} \delta_i(i_k)(\log p_\Theta(\theta_k|\lambda_i) + \log w_i) \\ &= \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{N_{cl}} \delta_i(i_k)\left(\frac{1}{2\sigma^2} J(\theta_k, \bar{\theta}_i) + d\log\{\sigma_i\} + \frac{d}{2}\log\{2\pi\} + \log w_i\right)\end{aligned}$$

avec $w_i \geq 0$ le poids de la i-ème composante ($\sum w_i = 1$, le label i_k est un entier associé au descripteur local θ_k et représente le cluster et $\delta_i(i_k) = 1$ si et seulement si θ_k est un échantillon issu du cluster numéro i, et 0 sinon.

Algorithme III.5 : Pseudo-code pour l'estimation des paramètres du modèle de mélange de gaussiennes concentrées

Données : Un ensemble de K descripteurs locaux $(\theta_k)_{k=1}^{K}$ et N_{cl} le nombre de clusters
Résultat : Un ensemble de N_{cl} paramètres de position $\left(\bar{\theta}_i\right)_{i=1}^{N_{cl}}$ et paramètres de variance $\left(\sigma_i^2\right)_{i=1}^{N_{cl}}$

Initialisation des N_{cl} paramètres $\left(\bar{\theta}_i, \sigma_i^2\right)_{i=1}^{N_{cl}}$
$I_{\text{intra}} \leftarrow \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{N_{cl}} \delta_i(i_k)\left(\frac{1}{2\sigma_i^2} J(\theta_k, \bar{\theta}_i) + d\log\{\sigma_i\}\right)$;
répéter
  répéter
    $I_{\text{old}} \leftarrow I_{\text{intra}}$;
    pour chaque *descripteur local k ;* `/* Assignation */`
    faire
      $i_k \leftarrow \underset{i}{arg\,\max} \log\left\{w_i p_\Theta(\theta_k|\bar{\theta}_i, \sigma_i^2)\right\}$;
    fin
    pour chaque *cluster i ;* `/* Assignation */`
    faire
      $N_i \leftarrow \sum_{k=1}^{K} \delta_i(i_k)$;
      $\sigma_i^2 \leftarrow \frac{1}{dN_i}\sum_{k=1}^{K} \delta_i(i_k) J(\theta_k, \bar{\theta}_i)$;
    fin
  jusqu'à *convergence locale*;
  pour chaque *cluster i ;* `/* Mise à jour du paramètre de position */`
  faire
    $N_i \leftarrow \sum_{k=1}^{K} \delta_i(i_k)$;
    $\bar{\theta}_i \leftarrow \underset{\bar{\theta}\in\Theta}{arg\,\min} \frac{1}{N_i}\sum_{k=1}^{K} \delta_i(i_k) J(\theta_k, \bar{\theta})$;
  fin
  $I_{\text{intra}} \leftarrow \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{N_{cl}} \delta_i(i_k)\left(\frac{1}{2\sigma^2} J(\theta_k, \bar{\theta}_i) + d\log\{\sigma_i\}\right)$;
jusqu'à $I_{old} \geq I_{intra}$;

Ainsi la log-vraisemblance complète peut être écrite comme :

$$\mathcal{L}(\{\theta_k, i_k\}_{k=1}^K \dots |\lambda) = \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{N_{cl}} \delta_i(i_k)\left(\frac{1}{2\sigma^2}J(\theta_k, \bar{\theta}_i) + d\log\{\sigma_i\}\right) + \frac{d}{2}\log\{2\pi\} - \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{N_{cl}}\delta_i(i_k)\log\{w_i\}$$

Comme le maximum de la log-vraisemblance en λ est recherché, la formule est soustraite de son contenu indépendant de λ :

$$\mathcal{L}(\{\theta_k, i_k\}_{k=1}^K \dots |\lambda) \simeq \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{N_{cl}} \delta_i(i_k)\left(\frac{1}{2\sigma^2}J(\theta_k, \bar{\theta}_i) + d\log\{\sigma_i\}\right)$$

L'algorithme III.5 présente l'estimation des paramètres du mélange de gaussiennes concentrées. Il est composé de deux boucles imbriquées. La boucle intérieure assigne chaque descripteur local à un cluster si et seulement si les paramètres de position $\bar{\theta}_i$ et de variance σ_i^2 du cluster numéro i maximisent la vraisemblance. Puis le paramètre de variance σ_i^2 est estimé de nouveau. La boucle extérieure appelle en premier la boucle intérieure, par conséquent le paramètre de variance σ_i^2 est optimal pour le paramètre de position $\bar{\theta}_i$ précédent. Ensuite la boucle extérieure estime de nouveau le paramètre de position $\bar{\theta}_i$ en fonction des clusters existants. Cette boucle se termine lorsque la log-vraisemblance complète de la loi mélange est minimale.

5 Conclusion

Dans ce chapitre, nous nous somme intéressés à la méthode de « Sac de mots visuels » comme une technique usuelle de classification. Cette méthode s'adapte aux descripteurs paramétriques qui sont souvent utilisés dans la communauté de classification d'images texturées. Ce chapitre aborde trois problèmes engendrés par l'utilisation de descripteurs paramétriques. Premièrement, nous privilégions une mesure de dissimilarité intrinsèque à l'espace des descripteurs paramétriques – ou variété riemannienne – Θ. La distance euclidienne est une mesure extrinsèque à la variété tandis que la divergence de Jeffrey est intrinsèque à la variété. Deuxièmement, les images naturelles présentent des différences au niveau descripteurs même si elles représentent le même objet, ce que nous appelons la diversité intra-classe. Cette diversité a pour effet de former des clusters de descripteurs paramétriques dans la variété Θ. Troisièmement, nous introduisons la notion de clustering, pour estimer la position de ces clusters. Nous définissons la

distribution gaussienne concentrée sur la variété Θ, conjointement à la mesure intrinsèque choisie. Par conséquent, l'algorithme de clustering que nous proposons est une loi mélange de gaussiennes concentrées sur Θ. Cela signifie que les clusters sont paramétriques, caractérisés par un paramètre de position $\bar{\theta}$ et un paramètre de variance σ^2. L'estimation de $\bar{\theta}$ est présenté avec plus de détails dans le chapitre suivant.

Chapitre IV

Estimation du Barycentre

Contenu du chapitre

1 Introduction

Dans ce document, nous développons une méthode de classification dite de « sac de mots visuels » (SMV) basée sur des descripteurs paramétriques. Le choix des vecteurs paramétriques a conduit à utiliser des mesures de dissimilarité spécifiques telles que les divergences de Kullback-Leibler et de Jeffrey. Dans le cadre de la distribution gaussienne généralisée (GGD) (exemple de modèle univarié), ces mesures ont une forme explicite (cf équation (III.6)). Dans le ;cadre multivarié, la mesure entre deux modèles SIRV revient à une somme de mesures entre les multiplieurs d'une part et les matrices de covariance d'autre part (cf équation (III.19)). En effectuant les hypothèses d'indépendances intra-bande et inter-bandes, et en ignorant la sous-bande d'approximation, la mesure de dissimilarité entre deux images texturées revient au calcul de la somme des mesures de dissimilarités entre chacune des sous-bandes (cf equation (III.2)). La section III.4 présente la construction du dictionnaire par l'estimation des statistiques d'ordre 1 et 2 sur un cluster d'images texturées. Comme la mesure est calculée séparément sur chaque sous-bande, le barycentre (statistique d'ordre 1) est calculé sur chaque sous-bande séparément, sans perte de généralité. L'estimateur de la variance autour du barycentre étant donné dans la proposition 3 page 72, il ne sera pas abordé dans ce chapitre. Ce chapitre présente l'estimation d'une distribution barycentrique au sens de la divergence de Jeffrey. Il s'agit, plus précisément, de l'estimation d'un barycentre sur une variété riemannienne constituée par l'ensemble des vecteurs paramétriques possibles.

Dans ce chapitre, le calcul du barycentre au sens de la divergence de Jeffrey est présenté. La première section présente l'état de l'art sur l'estimation de barycentre sur une variété riemannienne. Nous présentons alors la famille exponentielle. Un modèle issu d'une famille exponentielle a comme propriété d'avoir des formes explicites pour le barycentre. Puis nous nous intéressons à des modèles paramétriques qui ne sont pas extraits des familles exponentielles. C'est alors que nous présentons les méthodes de recherche linéaire pour estimer le barycentre. Dans une seconde partie nous abordons l'estimation du barycentre pour une distribution univariée, pour laquelle la mesure de dissimilarité est explicite. Nous nous penchons, sans perte de généralité, sur l'exemple de la distribution gaussienne généralisée (GGD). Une troisième partie présente l'estimation d'un barycentre pour une distribution multivariée. Ici, les modèles Spherically Invariant Random Vector (SIRV) sont étudiés à titre d'exemple, sachant que la mesure de dissimilarité est obtenue sur un vecteur aléatoire joint. Dans une quatrième partie, nous nous intéressons à étudier les propriétés de ces barycentres dans ce contexte de géométrie non euclidienne, notamment à travers de questionnements sur leur existence et leur unicité.

2 Estimation des hyper-paramètres

Cette première section est dédiée à définir les notations qui seront utilisées dans la suite de ce chapitre. La première sous-section explicite le problème d'estimation qui est relié à la définition d'une distribution gaussienne concentrée sur une variété riemannienne. La deuxième sous-section montre que pour un modèle paramétrique issu d'une famille exponentielle, le barycentre admet une formulation explicite. Pour cette famille de modèles, nous rappelons brièvement les algorithmes pour estimer les barycentres au sens de la divergence de Jeffrey. Cependant, comme nous l'avons abordé dans le chapitre 1 de cette thèse, les modèles paramétriques classiquement utilisés en analyse d'images texturées ne sont généralement pas issus de la famille exponentielle. Dans ce cas, il est nécessaire de revenir à la définition de barycentre afin de proposer un algorithme d'estimation, ce qui est l'objet de la troisième sous-section.

2.1 Estimation du barycentre

Le problème posé est ambigu,

— d'un côté la notion de barycentre est associée à une notion physique, une notion de distance. Dans le chapitre 1, nous avons montré que ni la divergence de Kullback-Leibler ni la divergence de Jeffrey ne sont des distances mais les divergences s'accordent avec la géométrie de l'espace riemannien. Sans distance, il n'est plus possible de parler de barycentre au sens conventionnel du terme.

— De l'autre côté, l'élément $\bar{\theta}$ est un estimateur de la position d'une distribution gaussienne concentrée sur l'espace Θ, soit un estimateur d'une statistique de premier ordre. Pour une distribution gaussienne, l'estimateur au sens du maximum de vraisemblance est une moyenne arithmétique qui coïncide avec le barycentre au sens de la distance euclidienne.

Dans l'ensemble de ce document, le vecteur $\bar{\theta}$ est nommé barycentre. Cette section va donc revenir en détail sur ce qu'est un barycentre et définira le vecteur $\bar{\theta}$.

2.1.1 Barycentre dans l'espace euclidien

Le barycentre est défini physiquement par la mécanique statique. Soit I un plan. Soit $(i_1, \ldots, i_N)$ une collection de N points de I, i_0 représente le point d'appui de l'espace I. Soient $(\vec{F}_1, \ldots, \vec{F}_N)$ les forces appliquées respectivement en $(i_1, \ldots, i_N)$ supposées verticales, et $\vec{F}$ la force appliquée au point d'appui i_0 supposé. Soient $(w_1, \ldots, w_N)$ les longueurs respectivement des forces $(\vec{F}_1, \ldots, \vec{F}_N)$, et $\bar{w}$ la longueur de la force $\vec{F}$. Soit $\overrightarrow{i_0 i_n}$ le vecteur correspondant à $\overrightarrow{i_0 i_n} = i_n - i_0$ pour tout $n = 1, \ldots, N$. Soit $\sqrt{\overrightarrow{i_0 i_n}^T \overrightarrow{i_0 i_n}}$ la distance euclidienne entre i_0 et i_n pour tout $n = 1, \ldots, N$. Le principe fondamental de la statique donne

alors deux équations :

$$\sum_{n=1}^{N} w_n \sqrt{\overrightarrow{i_0 i_n}^{\,T} \overrightarrow{i_0 i_n}} = 0, \qquad \sum_{n=1}^{N} w_n = 1$$

Le point d'appui i_0 correspond à la définition physique du barycentre connu, également comme le centre des masses.

2.1.2 Barycentre sur variété

En 1977, Karcher [96] étend le concept de barycentre pour des variétés riemanniennes, autrement dit les points sont sur un plan courbé. Soit Θ l'espace des paramètres, vu comme une variété riemannienne. Soient $(\theta_k)_{k=1}^{K}$ un jeu de K points de Θ et θ_0 un point quelconque de Θ. Ici $\overrightarrow{\theta_0 \theta_k}$ correspond à la dérivée en θ_0 de la géodésique reliant θ_0 à θ_k. L'extension de la notion de barycentre sur une variété est donnée par la moyenne de Karcher [96, 97] θ_0 qui vérifie :

$$\sum_{k=1}^{K} w_k \sqrt{\overrightarrow{\theta_0 \theta_k}^{\,T} G(\theta_0) \overrightarrow{\theta_0 \theta_k}} = 0, \qquad \sum_{k=1}^{K} w_k = 1$$

avec $G(\theta_0)$ la matrice d'information de Fisher [79, 82] calculée en θ_0. Autrement dit, le calcul de la moyenne de Karcher utilise la géométrie de la variété. La distance de Rao, notée $\mathrm{GD}(\theta_0, \theta_k)$, entre deux points est la longueur de la géodésique reliant θ_0 à θ_k, de plus le carré de la norme du vecteur $\sqrt{\overrightarrow{\theta_0 \theta_k}^{\,T} G(\theta_0) \overrightarrow{\theta_0 \theta_k}}$ égale le carré de la distance de Rao $\mathrm{GD}(\theta_0, \theta_k)$. Par conséquent, la moyenne de Karcher θ_0 vérifie :

$$\sum_{k=1}^{K} w_k \sqrt{\mathrm{GD}(\theta_0, \theta_k)} = 0, \qquad \sum_{k=1}^{K} w_k = 1$$

A titre d'exemple, la moyenne de Karcher a été utilisée pour l'estimation de barycentre sur l'espace des matrices de covariance [98](matrices semi-définies positives) avec des applications en élasticité [99], traitement de signaux radar [100, 101], imagerie médicale [102–104] et traitement des images [105].

2.1.3 Barycentre suivant une divergence

Pour les modèles paramétriques utilisés dans cette thèse (GGD, GΓD, modèle SIRV), la distance géodésique n'est pas connue explicitement. Néanmoins comme cette dernière admet une équivalence locale avec la divergence de Kullback-Leibler (voir équation (III.23) au chapitre 1), il est possible de définir des

barycentres orientés à gauche $\bar{\theta}^L$ et à droite $\bar{\theta}^R$ définis de la façon suivante :

$$\begin{aligned}\bar{\theta}^L &= \underset{\theta\in\Theta}{arg\,min} \sum_{k=1}^{K} w_k \mathrm{KL}(\theta \parallel \theta_k) \\ \bar{\theta}^R &= \underset{\theta\in\Theta}{arg\,min} \sum_{k=1}^{K} w_k \mathrm{KL}(\theta_k \parallel \theta)\end{aligned}$$

La communauté de classification d'images texturées incite à utiliser des vraisemblances qui ne suivent pas une famille exponentielle [46, 48, 57], tout en privilégiant la divergence de Jeffrey [44, 48, 57]. C'est dans cette optique que le barycentre au sens de la divergence de Jeffrey $\bar{\theta}$ est défini comme la solution minimisant la fonction de coût l, i.e. :

$$\bar{\theta} = \underset{\theta\in\Theta}{arg\,min}\ l(\theta) = \underset{\theta\in\Theta}{arg\,min} \sum_{k=1}^{K} w_k J(\theta, \theta_k)$$

Pour résumer, la littérature fournie un ensemble de définitions géométriques du barycentre que ce soit sur un espace euclidien ou sur un espace riemannien. Le point important étant de comprendre le rôle de chaque élément dans la formule finale par rapport à un exemple simple. Néanmoins le barycentre $\hat{\theta}$ qui est recherché dans le chapitre précédent est un estimateur qu'il ne faut pas confondre avec la définition géométrique du barycentre. Dans la section suivante, nous revenons sur l'estimateur pour mettre en avant les liens existants entre les formules de l'estimateur d'un côté et du barycentre de l'autre.

2.1.4 Estimateurs du maximum de vraisemblance

Le système de clustering utilisé par la méthode SMV est un mélange de gaussiennes. Si les variances des distribution gaussiennes sont toutes supposées égales, l'estimation des paramètres de mélange de gaussiennes coïncide avec un algorithme de type K-moyennes. Une spécialisation, au cas des vecteurs paramétriques, de la distribution gaussienne est proposée dans le chapitre précédent : la distribution gaussienne concentrée. Cette dernière est paramétrée par un barycentre $\bar{\theta}$ et une variance σ^2.

La proposition 3 vue au chapitre III, nous donne l'estimateur de $\hat{\theta}$:

$$\hat{\bar{\theta}} = \underset{\bar{\theta}\in\Theta}{arg\,min} \frac{1}{K} \sum_{k=1}^{K} J(\theta_k, \bar{\theta}).$$

Afin de simplifier les notations, considérons la définition :

Définition 7. Nous définissons la fonction de coût par :

$$l(\bar{\theta}) = \frac{1}{dK} \sum_{k=1}^{K} J(\theta_k, \bar{\theta})$$

Par exemple, une fonction de coût l_{GD} peut être définie au moyen d'une autre mesure de dissimilarité comme une distance géodésique :

$$l_{\mathrm{GD}}(\bar{\theta}) = \frac{1}{dK} \sum_{k=1}^{K} \mathrm{GD}^2(\theta_k, \bar{\theta})$$

Il est donc important de préciser que la fonction de coût définie avec la divergence de Jeffrey sera notée l. Pour éviter toute confusion, le symbole sera indenté différemment lorsque la mesure change, comme pour la fonction de coût l_{GD}. La définition de la fonction de coût l permet de réécrire les estimateurs au sens du maximum de vraisemblance

Lemme 7. *Soit l une fonction de coût, alors les estimateurs du maximum de vraisemblance des hyper paramètres sont donnés par :*

$$\begin{aligned} \hat{\bar{\theta}} &= arg\min_{\bar{\theta} \in \Theta} l(\bar{\theta}); \\ \hat{\sigma}^2 &= l(\hat{\bar{\theta}}). \end{aligned}$$

L'estimateur au sens du maximum de vraisemblance de l'écart-type $\hat{\sigma}$ est explicite dès que le barycentre est estimé. La difficulté de l'estimation est donc concentrée sur la minimisation de la fonction de coût l sur la variété riemannienne Θ. La section III.4.2 indique que Θ est muni d'une distance riemannienne qui est paramétrée par la matrice d'information de Fisher. Cette matrice est donnée par le modèle *a priori*. La section suivante montre un exemple simple de modèle *a priori* : la famille exponentielle pour laquelle les barycentres $\bar{\theta}^L$ et $\bar{\theta}^R$ admettent une forme explicite.

2.2 Famille exponentielle : formule explicite du barycentre

L'ensemble des familles exponentielles est défini comme une partition de l'espace des modèles paramétriques. Suite à la définition, les propriétés de la famille sont utilisées pour donner l'estimateur explicite des barycentres au sens de la divergence de Kullback-Leibler.

2.2.1 Définition

Définition 8. La densité de probabilité d'une famille exponentielle est donnée par [106] :

$$p(x \mid \lambda) = \exp\{< \lambda, t(x) > -A(\lambda) + C(x)\} \tag{IV.1}$$

avec $< \cdot, \cdot >$ un produit scalaire, $t(x)$ les statistiques suffisantes, λ un vecteur de paramètres naturels, A la fonction log-normalisante, enfin $C(x)$ est un terme de normalisation de la densité. De plus, la fonction log-normalisante A est utilisée pour caractériser la famille exponentielle.

2.2.2 Invariance à la paramétrisation

Paramétrisation. La définition précédente n'utilise pas la notation usuelle θ pour un vecteur de paramètres. En effet, le vecteur de paramètres naturels $\lambda = \phi(\theta)$ est l'image par ϕ, une fonction de changement de variables, du vecteur de paramètres θ dit « source ». Soit $\theta = (\theta_k)_{k=1}^K$ une collection de K paramètres sources. Soit $\lambda = (\lambda_k)_{k=1}^K$ une collection de K paramètres naturels, définis comme $\lambda_k = \phi(\theta_k)$.

Mesures de dissimilarité. La divergence de Kullback-Leibler entre deux distributions suivant un modèle avec paramètres naturels de la définition est notée KL_Λ :

$$\mathrm{KL}_\Lambda(\lambda_1 \| \lambda_2) = \mathrm{KLD}(p(x \mid \lambda_1) \| p(x \mid \lambda_2))$$

à l'opposé de la divergence de Kullback-Leibler entre deux distributions suivant un modèle avec paramètres source :

$$\mathrm{KL}(\theta_1 \| \theta_2) = \mathrm{KLD}(p(x \mid \theta_1) \| p(x \mid \theta_2)).$$

De part leurs définitions respectives, il est possible d'écrire une égalité entre les deux quantités :

$$\mathrm{KL}_\Lambda(\lambda_1 \| \lambda_2) = \mathrm{KL}_\Lambda(\phi(\theta_1) \| \phi(\theta_2)) = \mathrm{KL}(\theta_1 \| \theta_2) \tag{IV.2}$$

L'égalité (IV.2) montre que la dissimilarité calculée entre deux distributions ne dépend pas des paramètres sources ou naturels. La divergence de Kullback-Leibler est invariante à la paramétrisation.

2.2.3 Estimation

Le barycentre à droite Le barycentre $\bar{\theta}$ est défini comme le minimum de la fonction de coût l. Mais ici, deux divergences sont disponibles, une pour la paramétrisation θ et une pour λ. Comme la divergence de Kullback-Leibler n'est pas symétrique, soient $\bar{\theta}^R$ le barycentre au sens de la divergence de Kullback-Leibler à droite et $\bar{\theta}^L$ le barycentre au sens de la divergence de Kullback-Leibler à gauche (respectivement $\bar{\lambda}^R$ et $\bar{\lambda}^L$ pour les barycentres avec la paramétrisation naturelle). L'égalité (IV.2) peut être utilisée sur la formule de la fonction de coût :

$$\bar{\theta}^R = \underset{\bar{\theta}\in\Theta}{arg\,min} \frac{1}{K}\sum_{k=1}^{K} \mathrm{KL}_\Lambda(\phi(\theta_k)\|\phi(\bar{\theta})). \tag{IV.3}$$

La fonction de coût peut aussi être écrite pour la distribution paramétrique de paramètres naturels λ

$$\bar{\lambda}^R = \underset{\bar{\lambda}\in\Lambda}{arg\,min} \frac{1}{K}\sum_{k=1}^{K} \mathrm{KL}_\Lambda(\lambda_k\|\bar{\lambda}). \tag{IV.4}$$

Supposons maintenant que ϕ est un homéomorphisme entre Θ et Λ, autrement dit pour tous paramètres naturels $\lambda_k \in \Lambda$ il existe des paramètres sources $\theta_k \in \Theta$ tel que $\lambda_k = \phi(\theta_k)$. Alors, par identification entre les égalités (IV.3) et (IV.4), les barycentres sont reliés $\bar{\lambda}^R = \phi(\bar{\theta}^R)$. Au moyen de la fonction de changement de variable ϕ (et sa fonction réciproque ϕ^{-1}), il suffit de calculer une fois le barycentre $\bar{\lambda}^R$ pour avoir le barycentre $\bar{\theta}^R$ recherché.

Le barycentre à gauche. De manière similaire, le barycentre à gauche $\bar{\theta}^L = \phi^{-1}(\bar{\lambda}^L)$ est défini comme solution de :

$$\bar{\lambda}^L = \underset{\bar{\lambda}\in\Lambda}{arg\,min} \frac{1}{K}\sum_{k=1}^{K} \mathrm{KL}_\Lambda(\bar{\lambda}\|\lambda_k). \tag{IV.5}$$

La suite de cette section donne les formules explicites pour les deux barycentres orientés $\bar{\lambda}^L$ et $\bar{\lambda}^R$.

Formules explicites. Le problème d'estimation des barycentres à gauche et à droite est un problème dual. Soit H l'espace espéré, le barycentre à gauche $\bar{\lambda}^L$ dans Λ devient un barycentre à droite $\bar{\nu}^R$ dans H. En bref, la théorie indique que seul le barycentre à droite admet une formule explicite quel que soit son espace. La formule explicite pour le barycentre à gauche $\bar{\lambda}^L$ est solution du problème dual. Le gradient ∇A de la fonction log-normalisante A définit un changement de variable depuis l'espace naturel Λ vers l'espace espéré H. La transformation de Legendre [106] du gradient ∇A constitue un autre changement de variables partant de l'espace espéré H vers l'espace naturel Λ. Les deux théorèmes de Banerjee [94]

explicitent les barycentres issus de la divergence de Kullback-Leibler.

Théorème 8. *Soient $\mathcal{P}$ un modèle paramétrique sur les données x, Θ l'ensemble de tous les vecteurs de paramètres de $\mathcal{P}$, d la dimension de l'espace Θ, $p(x \mid \theta)$ une vraisemblance suivant $\mathcal{P}$. Soient K vecteurs de paramètres $\theta_k \in \Theta$ pour $k = 1, \ldots, K$.*

Supposons que le modèle paramétrique est issu d'une famille exponentielle. Soit ϕ la fonction de changement de variable entre les paramètres sources θ_k et les paramètres naturels. Soient Λ l'espace des paramètres naturels, $\lambda_k = \phi(\theta_k)$ les paramètres naturels associés à chaque paramètre source θ_k. Posons A la fonction log-normalisante définie sur Λ, ∇A la dérivée de la fonction A et $(\nabla A)^{-1}$ la fonction réciproque de ∇A.

Le barycentre orienté à gauche s'exprime :

$$\bar{\lambda}^L = (\nabla A)^{-1}\left(\frac{1}{dK}\sum_{k=1}^{K}\nabla A(\lambda_k)\right) \tag{IV.6}$$

et le barycentre orienté à droite s'exprime :

$$\bar{\lambda}^R = \frac{1}{dK}\sum_{k=1}^{K}\lambda_k \tag{IV.7}$$

Remarque : Soient $\bar{\theta}^L$ et $\bar{\theta}^R$ les barycentres orientés à gauche et à droite respectivement. Par invariance à la paramétrisation, nous avons l'égalité $\bar{\lambda}^L = \phi(\bar{\theta}^L)$ (respectivement $\bar{\lambda}^R = \phi(\bar{\theta}^R)$) pour le barycentre à gauche (respectivement à droite) dans l'espace naturel.

Le barycentre symétrique. Le barycentre $\bar{\theta}$ qui est estimé est solution de la fonction de coût l construite avec la divergence de Jeffrey. Cette divergence est obtenue comme somme de deux divergences de Kullback-Leibler opposées. Par conséquent, il existe une relation entre les barycentres orientés ($\bar{\theta}^L$ et $\bar{\theta}^R$) et le barycentre recherché $\bar{\theta}$. Comme décrit précédemment, la fonction de changement de variable ϕ présente le barycentre $\bar{\theta} = \phi(\bar{\lambda})$ comme l'image du barycentre $\bar{\lambda}$ au sens de la divergence de Jeffrey. Nielsen et Nock [106] proposent un algorithme d'estimation numérique du barycentre $\bar{\lambda}$ au sens de la divergence de Jeffrey au moyen d'une dichotomie entre le barycentre à gauche $\bar{\lambda}^L$ et le barycentre à droite $\bar{\lambda}^R$.

Par conséquent, seul le barycentre à droite $\bar{\theta}^R$ admet une formule explicite pour une distribution issue d'une famille exponentielle. La compréhension du caractère dual de l'espace espéré H permet de déduire une formule explicite pour le barycentre à gauche $\bar{\theta}^L$. Nielsen et Nock ne présentent pas de formule

Algorithme IV.1 : Pseudo-code de la dichotomie entre les barycentres orientés à gauche et à droite

Données : Le barycentre à gauche $\bar{\lambda}^L$, le barycentre à droite $\bar{\lambda}^R$
Résultat : Le barycentre symétrique $\bar{\lambda}$

$t_m \leftarrow 0, t_M \leftarrow 1$; /* Nous supposons que $t \in [t_m, t_M]$ */
tant que $t_M - t_m \geqslant \epsilon$; /* ϵ est la précision machine */
faire
  % 1. Déplacement géodésique.
  $t_h \leftarrow t_m/2 + t_M/2$; /* Calcule la moitié de l'intervalle */
  $\lambda_h \leftarrow (\nabla A)^{-1}\left((1-t_h)\nabla A(\bar{\lambda}^R) + t_h \nabla A(\bar{\lambda}^L)\right)$; /* Calcule le vecteur correspondant */
  % 2. Bisecteur de type mélange
  s_h signe de $\mathrm{KL}_\Lambda(\lambda_h \| \bar{\lambda}^R) - \mathrm{KL}_\Lambda(\bar{\lambda}^L \| \lambda_h)$
  % 3. Dichotomie
  si $s_h < 0$ **alors**
    $t_m \leftarrow t_h$
  sinon
    $t_M \leftarrow t_h$
  fin
fin
$\bar{\lambda} \leftarrow \lambda_h$

explicite pour le barycentre symétrique $\bar{\theta}$. Dans un cas favorable, utilisant une famille exponentielle, le barycentre au sens de la divergence de Jeffrey n'admet pas de formule explicite. Dans la suite de ce mémoire, nous aborderons le cas plus large des modèles paramétriques utilisés dans la classification d'images texturées. Pour rappel, nous discutons de la manière d'obtenir le barycentre $\hat{\theta}$.

2.3 Modèles paramétriques pour la classification d'images texturées

Le calcul de barycentre symétrique est obtenu simplement pour tout modèle issu d'une famille exponentielle. Nielsen [107] propose un document applicatif autour des familles exponentielles. Ce document contient une liste des distributions usuelles parmi lesquelles se trouve la distribution gaussienne univariée, la distribution gaussienne multivariée, la distribution de Poisson, la distribution de Laplace centrée, la distribution de Bernoulli, la distribution binomiale, la distribution multinomiale, la distribution de Rayleigh, la distribution Gamma. Dans le cadre de la classification d'images texturées, les modèles utilisés diffèrent. Unser [26] utilise une loi gaussienne multivariée (exemple de famille exponentielle) comme modèle paramétrique. Il obtient des gains en performances de classifications en utilisant les statistiques d'ordre 3 et 4 (respectivement le coefficient asymétrique et le coefficient de kurtosis) en plus des deux paramètres de position et de variance-covariance. La GGD, la GΓD, les modèles SIRV, les modèles à copules gaussienne ou de Student ne font, malheureusement, pas partie intégrante de la famille exponentielle. Cette question a déclenché une collaboration avec messieurs Schwander et Nielsen de l'école polytechnique et a

notamment donné lieu à une publication dans la conférence ICPR 2012 [62]. Si le paramètre de forme β est supposé fixe, alors nous pouvons démontrer que la GGD est une famille exponentielle.

Ce bref état de l'art sur le barycentre $\bar{\theta}$ est insuffisant pour une application numérique des mélanges de gaussiennes concentrées. Pour le moment, le problème de l'estimation du barycentre $\bar{\theta}$ coïncide avec la recherche du minimum de la fonction de coût l. Et c'est dans cette direction que se poursuit le mémoire : minimiser une fonction de coût.

3 Estimation du barycentre avec une méthode d'optimisation

Le premier chapitre a présenté la méthode de sac de mots visuels. Au cours de la phase d'apprentissage, des barycentres sont calculés sur les clusters afin de représenter la classe d'images. Cette méthode est indépendante des descripteurs choisis. L'étude des descripteurs univariés est séparée de l'étude des descripteurs mutivariés pour, d'une part, décrire le processus en détails sur un exemple simplifié et, d'autre part, valider les descripteurs multivariés par des performances de classification. Sans perte de généralité, la distribution gaussienne généralisée est utilisée comme exemple de modèle univarié. Ce modèle a notamment été validé par Do et Vetterli [4] pour la classification d'images texturées et présente seulement deux paramètres à estimer (un paramètre d'échelle α et un paramètre de forme β).

3.1 Etat de l'art de l'optimisation

En dehors des modèles issus de la famille exponentielle, la littérature [44] sur les barycentres est plus concise. Par conséquent, il est nécessaire d'étendre ces travaux en revenant sur le problème de la minimisation de la fonction de coût l.

3.1.1 Optimisation sur un espace euclidien

« Mathématiquement parlant, l'optimisation est la minimisation ou la maximisation d'une fonction sous contraintes sur ces variables. » Nocedal et Wright [108]. Cet état de l'art s'inspire des travaux de Nocedal et Wright pour donner un point de vue général sur la méthode d'optimisation d'une part et sur les problèmes inhérents à l'optimisation d'autre part. Les notations usuelles et la terminologie associée sont données dans la liste suivante :

— θ un vecteur de *paramètres*, aussi appelé *variables* ou *inconnues* ;

— l la *fonction de coût* dépendant de θ dont nous cherchons un extremum ;

— c un vecteur de *contraintes* que les paramètres doivent vérifier. C'est également une fonction de variable θ. Le nombre de composantes de c est le nombre de restrictions individuelles qui sont posées sur les variables.

Prenons l'exemple de la distribution gaussienne généralisée (GGD), le problème d'optimisation s'écrit :

$$\min_{\theta \in \Theta} l(\theta) \quad \text{avec } \theta = (\alpha, \beta) \qquad \text{sous contraintes} \begin{cases} \alpha > 0 \\ \beta > 0 \end{cases} \tag{IV.8}$$

D'autres notations sont utilisées dans cette section, soient ∇l le gradient de la fonction de coût l et $\mathcal{H}l$ la hessienne de la fonction de coût l.

Optimisation locale ou globale. Cet état de l'art commence par une clarification du vocabulaire de l'optimisation. Les algorithmes ayant la complexité calculatoire la moins élevée sont ceux recherchant une solution locale, un point de l'espace où la fonction de coût est minimale sur un voisinage plus ou moins grand. Ils ne cherchent en général pas la meilleure des solutions, le minimum global. Le minimum global est nécessaire (voire très recherché) pour certaines applications. Mais le minimum global est généralement très difficile à identifier (comparaison exhaustive de tous les minima locaux) et il est même plus difficile à positionner (comme un extremum de la fonction : la dérivée de la fonction de coût l est nulle). Lorsque la fonction de coût est convexe, alors tout minimum local est un minimum global. Néanmoins, les problèmes non-linéaires usuels, qu'ils soient sous contraintes ou non, peuvent posséder des solutions locales qui ne sont pas le minimum global.

La fin de ce chapitre présente une discussion autour de l'existence et l'unicité du vecteur $\bar{\theta}$, qui est reliée à cette différence entre solution locale et solution globale. Après ce court précis sur le vocabulaire, continuons sur le fonctionnement d'un algorithme d'optimisation.

Les algorithmes d'optimisation. Les algorithmes d'optimisation sont itératifs. Ils sont initialisés à partir d'une première approche de la valeur optimale $\bar{\theta}$. Ils génèrent ensuite une suite de vecteurs qui réduisent la fonction de coût par rapport au précédent vecteur, jusqu'à atteindre le minimum local. La stratégie adoptée pour passer d'un estimé à l'itération i vers l'estimé à l'itération $i+1$ est la principale différence entre les algorithmes d'optimisation. La plupart des stratégies utilisent au mieux les valeurs prises par la fonction de coût l, les contraintes c voire les dérivées d'ordre 1 et 2 de ses fonctions. Certains algorithmes accumulent de l'information apprise aux itérations précédentes, là où d'autres algorithmes n'utilisent que l'information locale autour de l'estimé courant. La liste suivante présente les propriétés

communes aux algorithmes efficaces :

— Robustesse. Ils doivent avoir de bonnes performances sur un large panel de problèmes dans leur classe, pour tout choix raisonnable du vecteur d'initialisation.

— Efficacité. Ils ne doivent pas nécessiter une grande quantité de mémoire ou une grande complexité calculatoire.

— Précision. Ils doivent être capables de localiser un minimum local avec précision, sans être sensibles outre mesure aux erreurs dans les données ou aux erreurs dans l'approximation dues aux arrondis qui apparaissent lorsque l'algorithme est implémenté sur une machine.

Il peut arriver que certains objectifs soient contradictoires. Par exemple, une méthode de programmation non linéaire qui converge rapidement peut utiliser une trop grosse quantité de mémoire pour des problèmes plus larges. D'un autre côté, une méthode robuste peut être la plus lente. De ce fait, le problème de l'optimisation réside dans les compromis à faire entre précision et rapidité, complexité calculatoire et espace mémoire nécessaire, etc...

Pour chaque itération, la méthode de recherche linéaire estime la direction de recherche p_j et décide de la distance parcourue suivant p_j. Une itération est donnée par :

$$\theta_{j+1} = \theta_j + p_j \tag{IV.9}$$

La réussite d'une méthode de recherche linéaire dépend des choix effectifs suivant la direction p_j.

La plupart des méthodes de recherche linéaire nécessitent que p_j soit la « direction de descente » - celle pour laquelle $p_j \nabla l(\theta_j) < 0$ - parce que cette propriété garantit que la fonction l peut être réduite suivant cette direction. De plus, les directions de recherche sont généralement de la forme :

$$p_j = -B_j^{-1} \nabla l(\theta_j)$$

avec B_j une matrice semi-définie positive. Pour la méthode de descente de gradient, B_j est la matrice identité I_d, alors que pour la méthode de Newton B_j est la hessienne calculée $\mathcal{H}l(\theta_j)$.

Pseudo-code associés aux différentes méthodes. La description précédente du fonctionnement peut être illustrée au moyen d'un pseudo-code. L'algorithme IV.2 présente la méthode de descente de gradient à pas fixe. Cet algorithme est initialisé avec la moyenne arithmétique $\theta_{0,\text{mo}}$ sans perte de généralité. À chaque itération de l'algorithme, une vérification sur la norme de la mise à jour est effectuée

afin de savoir si l'algorithme a convergé. S'il n'a pas encore convergé, l'équation (IV.9) est utilisée pour calculer l'élément suivant dans la suite.

L'algorithme IV.3 présente la méthode de Newton-Raphson à pas fixe. C'est une méthode de recherche linéaire comme la méthode de descente de gradient, ce qui explique le grand nombre de similitudes entre les deux algorithmes. La différence se situe dans l'estimation de la direction de descente. La direction de descente est « corrigée » par la hessienne $\mathcal{H}l$ de la fonction de coût. La hessienne peut être assimilée à une indication sur la courbure de la fonction de coût, ce qui est une information importante pour accélérer la convergence de la méthode.

Algorithme IV.2 : Pseudo-code de la méthode de descente de gradient

Données : Un ensemble de K réalisations $(\theta_k)_{k=1}^K$, ∇l le gradient de la fonction de coût
Résultat : Le vecteur θ estimé de $\bar{\theta}$

$\theta \leftarrow \frac{1}{K}\sum_{k=1}^K \theta_k$;
tant que $\|\nabla l(\theta)\| \geqslant \epsilon$; `/* ϵ est la précision machine */`
faire
  $\theta \leftarrow \theta - \nabla l(\theta)$;
fin

Algorithme IV.3 : Pseudo-code de la méthode de Newton-Raphson

Données : Un ensemble de K réalisations $(\theta_k)_{k=1}^K$, ∇l le gradient et $\mathcal{H}l$ la hessienne de la fonction de coût
Résultat : Le vecteur θ estimé de $\bar{\theta}$

$\theta \leftarrow \frac{1}{K}\sum_{k=1}^K \theta_k$;
tant que $\|(\mathcal{H}l(\theta))^{-1}\nabla l(\theta)\| \geqslant \epsilon$; `/* ϵ est la précision machine */`
faire
  $\theta = \theta - (\mathcal{H}l(\theta))^{-1}\nabla l(\theta)$;
fin

Le fonctionnement général d'un algorithme d'optimisation est donc simple. Mais il ne prend pas en compte les contraintes du problème qui est posé. Avant de présenter le problème plus complet de l'optimisation sous contraintes, présentons un des avantages de la fonction de coût l qui doit être minimisée. Les méthodes de descente de gradient sont invariantes à la paramétrisation de l'espace pour minimiser la fonction l. Cela permet de choisir librement la paramétrisation de l'espace, en privilégiant celle qui facilite les calculs par exemple.

Proposition 4. *Soit ϕ une formule de changement de coordonnées sans courbure (dérivée seconde nulle). Soit $\lambda_k = \phi(\theta_k)$ le vecteur de nouveaux paramètres associé à la distribution de paramètres θ_k. Soit Λ*

l'image par ϕ de l'espace Θ. Soit $l_{\phi^{-1}}$ la fonction de coût définie sur la nouvelle paramétrisation :

$$l_{\phi^{-1}}(\lambda_k) = l(\phi^{-1}(\lambda_k)).$$

Soit l_Λ la fonction de coût définie sous les paramètres Λ :

$$l_\Lambda(\lambda) = \sum_{k=1}^{K} JD(p(x \mid \lambda), p(x \mid \lambda_k))$$

alors l'invariance à la paramétrisation de la divergence de Jeffrey permet d'écrire l'égalité $l_{\phi^{-1}} = l_\Lambda$.

Soit $(\theta_j)_{j=1}^J$ la suite d'itérations obtenues avec la méthode de Newton-Raphson sur la fonction de coût l (respectivement $(\lambda_j)_{j=1}^J$ la suite d'itérations obtenues avec la méthode de Newton-Raphson sur la fonction de coût l_Λ). Si les initialisations θ_0 et λ_0 vérifient l'équation $\lambda_0 = \phi(\theta_0)$ alors $\lambda_j = \phi(\theta_j)$ pour tout entier j.

Démonstration. Soit ϕ une formule de changement de coordonnées sans courbure. Soient i un entier strictement positif et inférieur à la dimension de Λ, ϕ_i la formule envoyant la coordonnée λ_i du vecteur λ. Par développement limité, cette fonction s'écrit :

$$\phi_i(\theta) = \phi_i(\bar{\theta}) + \sum_{k=1}^{\infty} \frac{1}{(k-1)!} D^k \phi_i(\bar{\theta})(\theta - \bar{\theta})^k$$

La seconde dérivée vérifie :

$$D^2\phi_i(\theta) = \sum_{k=2}^{\infty} \frac{1}{(k-1)!2!} D^k \phi_i(\bar{\theta})(\theta - \bar{\theta})^{k-2}$$

Comme la seconde dérivée est nulle pour tout θ, nous pouvons affirmer que $D^k\phi_i(\bar{\theta}) = 0$ pour tous $k > 1$. Autrement dit, ϕ est un changement de coordonnées affine.

Le gradient de la fonction de coût $l_{\phi^{-1}}$ vérifie :

$$\nabla l_{\phi^{-1}}(\lambda) = J_{\phi^{-1}}(\lambda) \nabla l(\phi^{-1}(\lambda))$$

avec $J_{\phi^{-1}}$ la matrice jacobienne associée au changement de coordonnées ϕ^{-1}. Ensuite la hessienne de la fonction de coût $l_{\phi^{-1}}$ vérifie :

$$\mathcal{H}\, l_{\phi^{-1}}(\lambda) = J_{\phi^{-1}}(\lambda) \mathcal{H}\, l(\phi^{-1}(\lambda)) J_{\phi^{-1}}(\lambda)^T + \nabla l(\phi^{-1}(\lambda) \nabla (J_{\phi^{-1}})(\lambda)$$

Puis, par la non courbure du changement de variables,

$$\mathcal{H}\, l_{\phi^{-1}}(\lambda) = J_{\phi^{-1}}(\lambda)\mathcal{H}\, l(\phi^{-1}(\lambda))J_{\phi^{-1}}(\lambda)^T$$

Ceci permet de formuler la direction de descente :

$$\begin{aligned} p_j &= -\left(J_{\phi^{-1}}(\lambda_j)\mathcal{H}\, l(\phi^{-1}(\lambda_j))J_{\phi^{-1}}(\lambda_j)^T\right)^{-1} J_{\phi^{-1}}(\lambda_j)\nabla l(\phi^{-1}(\lambda_j)) \\ &= -(J_{\phi^{-1}}(\lambda_j)^T)^{-1}(\mathcal{H}\, l(\phi^{-1}(\lambda_j)))^{-1}\nabla l(\phi^{-1}(\lambda_j)) \end{aligned}$$

□

Optimisation sous contraintes. Dans un cadre plus complet, il est possible de prendre en compte les contraintes sur les paramètres. Le paragraphe suivant présente les équations associées à la formulation sous contraintes qui peut être envisagée afin d'améliorer les performances. Supposons maintenant que notre problème soit soumis à un ensemble de 2 contraintes $c_1(\theta)$ et $c_2(\theta)$ sur les paramètres θ. Par exemple, pour la GGD avec la formule (IV.8), les contraintes sont des contraintes d'inégalité sur les paramètres $\alpha > 0$ et $\beta > 0$. L'équation (IV.8) est alors formulée au moyen d'une fonction lagrangienne. La fonction de coût est contrainte en lui soustrayant $\lambda_i c_i(\theta)$, pour tout $i = 1, 2$. λ_1 et λ_2 sont deux réels strictement positifs, nommés « multiplieurs de Lagrange », qui composent le vecteur $\lambda = (\lambda_1, \lambda_2)$.

$$\mathcal{L}(\theta, \lambda) = l(\theta) - \lambda_1 c_1(\theta) - \lambda_2 c_2(\theta) \tag{IV.10}$$

Lorsqu'il n'existe aucune direction de descente au point $\bar{\theta}$, l'équation de l'équilibre s'écrit :

$$\nabla_{\bar{\theta}}\mathcal{L}(\bar{\theta}, \bar{\lambda}), \qquad \text{pour un certain } \bar{\lambda} \geqslant 0 \tag{IV.11}$$

sous contrainte que :

$$\bar{\lambda}_1 c_1(\bar{\theta}) = 0, \quad \bar{\lambda}_2 c_2(\bar{\theta}) = 0 \tag{IV.12}$$

Le vecteur $\bar{\theta}$ qui minimise la fonction lagrangienne est défini pour minimiser également la fonction de coût. Minimiser la fonction lagrangienne est une condition suffisante pour l'obtention du minimum de la fonction de coût.

L'état de l'art ainsi présenté est valable sur tout espace euclidien. Or l'espace des vecteurs paramétriques est une variété riemannienne qui n'est pas forcément plate. Le paragraphe suivant montre que des

algorithmes d'optimisation ont été étendus au cas des variétés riemanniennes.

3.1.2 Optimisation sur une variété riemannienne

Pour bien comprendre le fonctionnement d'une méthode de recherche linéaire, il faut revenir à la notion fondamentale de la distance. Lorsque les points A et B font partie de la variété Θ, la distance est alors la longueur de la géodésique, courbe la plus courte reliant A à B. Amari et Douglas [109] proposent d'utiliser cette connaissance de la géométrie de la variété pour faire avancer l'algorithme de recherche linéaire.

Algorithme IV.4 : Pseudo-code de la méthode d'adaptation du gradient naturel

Données : Un ensemble de K réalisations $(\theta_k)_{k=1}^K$, ∇l le gradient de la fonction de coût et G la matrice d'information de Fisher de Θ
Résultat : Le vecteur θ estimé de $\bar{\theta}$

$\theta \leftarrow \frac{1}{K}\sum_{k=1}^K \theta_k$;
tant que $\|(G(\theta))^{-1}\nabla l(\theta)\| \geqslant \epsilon$; `/* ϵ est la précision machine */`
faire
  $\theta = \theta + (G(\theta))^{-1}\nabla l(\theta)$;
fin

En pratique, la fonction coût l à minimiser n'est pas euclidienne, l'espace des paramètres sous-jacent n'est pas euclidien mais courbé et tordu, autrement dit riemannien. Par conséquent, l'opposé du gradient $-\nabla l(\theta)$ ne représente pas la direction de descente de la fonction de coût l dans l'espace des paramètres, et alors la méthode de descente de gradient n'est plus appropriée. Pour écrire un algorithme avec des propriétés de convergence utiles, la méthode de descente de gradient doit s'adapter à la courbure locale de l'espace des paramètres. Cette méthode est nommée « gradient naturel ». La direction de descente se formule comme suit :

$$p_j = -G(\theta_j)^{-1}\nabla l(\theta_j)$$

avec $G(\theta)$ la matrice d'information de Fisher au point θ de la variété.

Le gradient naturel possède de nombreuses propriétés [109], la liste suivante évoque trois de ses propriétés utiles pour mettre en avant la méthode. La méthode d'adaptation du gradient naturel

— diffère de l'algorithme de Newton en général,

— propose une mise à jour des paramètres non linéaire et

— peut être plus simple à mettre en place que les méthodes d'approximations usuelles.

Cette méthode ne peut donc pas être confondue avec la méthode de Newton-Raphson, et propose une mise en place plus simple puisque le calcul de la hessienne $\mathcal{H}l$ de la fonction de coût l n'est pas nécessaire. La

propriété de mise à jour non linéaire est également intéressante. Par exemple, pour la GGD la variabilité des paramètres de forme β_k est moins importante que la variabilité des paramètres d'échelle α_k, ce qui est présenté plus en détails dans la suite de ce chapitre dédié à l'estimation du vecteur $\bar{\theta}$ minimum de la fonction de coût l.

3.2 Problèmes algorithmiques

Pour minimiser une fonction de coût l, la théorie de l'optimisation préconise l'utilisation d'un algorithme de descente. Le fonctionnement de cet algorithme a été présenté puis adapté au contexte spécifique des variétés riemanniennes. L'algorithme ainsi défini n'est pas encore utilisable sur un ordinateur car certaines variables n'ont pas été choisies. Donc cette section présente une discussion autour du critère d'arrêt et de l'initialisation de l'algorithme d'optimisation.

Une fois les variables fixées, une application numérique de l'algorithme d'optimisation est réalisé. L'objectif de cette application numérique est de justifier l'utilisation de la méthode d'adaptation du gradient naturel. C'est une méthode moins explicite qu'une descente de gradient mais avec le même résultat : l'obtention du minimum de la fonction de coût l. Les différentes implémentations seront comparées par rapport à leur vitesse de convergence qui joue sur le nombre de boucles pour atteindre le critère d'arrêt.

3.2.1 Choix théorique : le critère d'arrêt

En pratique, la véritable valeur $\bar{\theta}$ n'est pas connue, expliquant l'utilisation d'un algorithme d'optimisation pour l'estimer. Cela signifie que le critère d'arrêt de l'algorithme ne peut pas être l'évaluation d'une distance entre l'itération θ_j et la véritable valeur $\bar{\theta}$.

Du fait de la convergence de la suite $(\theta_j)_{j=1}^{\infty}$ vers le vecteur $\bar{\theta}$, l'écart entre le vecteur à l'itération i et le vecteur à l'itération $i+1$ tend à décroître. Le critère d'arrêt de l'algorithme est donc fixé autour d'une vérification de l'écart entre θ_{j+1} et θ_j, par rapport à une valeur fixée de ϵ

$$\exists N_j \in \mathbb{R}, \text{tel que } \forall j > N_j, \|\theta_{j+1} - \theta_j\|_2 \leqslant \epsilon.$$

Qui devient une inégalité sur la direction de descente, en utilisant l'égalité (IV.9)

$$\exists N_j \in \mathbb{R}, \text{tel que } \forall j > N_j, \|\theta_{j+1} - \theta_j\|_2 = \|\theta_j + p_j - \theta_j\|_2 = \|p_j\|_2 \leqslant \epsilon.$$

En pratique, le réel ϵ est fixé à la valeur machine du plus petit nombre non nul.

3.2.2 Choix théorique : l'initialisation

L'initialisation θ_0 d'une méthode de recherche linéaire doit être acceptable au sens de l'application. Revenons un instant sur la définition de $\bar{\theta}$ comme minimum de la fonction de coût l. Minimiser la fonction de coût l, revient à minimiser une somme de mesures de dissimilarités avec les vecteurs θ_k, pour tout $k = 1, \ldots, K$. Le caractère optimal du vecteur $\bar{\theta}$ correspond aux conditions asymptotiques à la limite avec toutes les distributions très proches. Dire que deux distributions sont proches revient à dire que la dissimilarité est faible entre elles. Par la propriété de la divergence, lorsque la dissimilarité est faible, la mesure de dissimilarité est égale à la distance géodésique. Asymptotiquement, la position espérée de $\bar{\theta}$ est la même que le barycentre $\bar{\theta}_{\mathrm{GD}}$ au sens de la distance géodésique. Par similitude avec la fonction de coût l, soit la fonction de coût basée sur une distance géodésique l_{GD} :

$$l_{\mathrm{GD}}(\bar{\theta}) = \frac{1}{dK} \sum_{k=1}^{K} \mathrm{GD}^2(\theta_k, \bar{\theta})$$

Proposition 5. *Soient $\mathcal{P}$ un modèle paramétrique sur les données x, Θ l'ensemble de tous les vecteurs de paramètres de $\mathcal{P}$, d la dimension de l'espace Θ, $p(x \mid \theta)$ une vraisemblance suivant $\mathcal{P}$. Soient K vecteurs de paramètres $\theta_k \in \Theta$ pour $k = 1, \ldots, K$. Soit l_{GD} la fonction de coût définie suivant la distance géodésique et $(\theta_k)_{k=1}^K$.*

L'élément $\bar{\theta}_{GD}$ qui minimise l_{GD} est positionné dans un espace convexe de Θ contenant la collection de vecteurs $(\theta_k)_{k=1}^K$.

Définition 9. $A \subset \Theta$ est un sous-ensemble convexe de Θ si pour tous vecteurs paramétriques θ_1 et $\theta_2 \in A$ nous avons $\forall t \in [0, 1]$:

$$\theta = \gamma_{1\to 2}(t) \in A$$

avec $\gamma_{1\to 2}$ une paramétrisation de la géodésique partant de $\theta_1 = \gamma_{1\to 2}(0)$ vers $\theta_2 = \gamma_{1\to 2}(1)$.

Démonstration. Cela se montre par récurrence sur K :

1. Cas où $K = 1$, la fonction de coût coïncide avec une mesure de dissimilarité entre θ_1 et $\bar{\theta}$. Le minimum d'une mesure est atteinte lorsque $\bar{\theta} = \theta_1$ qui est le plus petit espace convexe de Θ.
2. Cas de $K = 2$, pour une mesure de dissimilarité symétrique, nous avons l'égalité : $l_{\mathrm{GD}}(\theta_1) = l_{\mathrm{GD}}(\theta_2)$. Par le théorème des valeurs intermédiaires, il existe c éléments de la géodésique $\gamma_{1\to 2}$ reliant θ_1 à θ_2 pour lesquels $\nabla l_{\mathrm{GD}}(c) = 0$, condition suffisante pour dire que $c = \bar{\theta}$. $\bar{\theta}$ appartient à un espace convexe contenant θ_1 et θ_2 au sens géodésique.

3. Soit une collection de $K-1$ vecteurs $(\theta_k)_{k=1}^{K-1}$ vérifiant que $\bar{\theta}$ appartienne à un convexe contenant les vecteurs $(\theta_k)_{k=1}^{K-1}$. Posons $\bar{\theta}_{K-1}$ l'élément $\bar{\theta}$ calculé sur les vecteurs $(\theta_k)_{k=1}^{K-1}$. La fonction de coût l_{GD} est dérivable au point $\bar{\theta}_{K-1}$ et prend une valeur nulle ou non nulle. Si $\nabla l_{\mathrm{GD}}(\bar{\theta}_{K-1}) = 0$ alors la condition est suffisante pour faire de $\bar{\theta}_{K-1}$ un minimum de l_{GD}. Sinon $l_{\mathrm{GD}}(\bar{\theta}_{K-1}) = l_{\mathrm{GD}}(\theta_K)$ et, par le théorème des valeurs intermédiaires, nous avons une condition suffisante pour que $\bar{\theta}$ appartienne à une géodésique reliant $\bar{\theta}_{K-1}$ et θ_K. Nous savons donc que le barycentre appartient encore à un convexe au sens géodésique, contenant les vecteurs $(\theta_k)_{k=1}^{K}$.

Le barycentre $\bar{\theta}_{\mathrm{GD}}$ au sens géodésique se trouve dans un convexe contenant la collection. □

Asymptotiquement, le vecteur $\bar{\theta}$ est le barycentre au sens de la distance géodésique $\bar{\theta}_{\mathrm{GD}}$. Soit une famille exponentielle, dans ce cas, les barycentres orientés à gauche et à droite sont deux barycentres égaux asymptotiquement. Alors que la distribution associée à chaque barycentre est fortement différente [106]. Ce qu'il faut comprendre c'est que le vecteur $\bar{\theta}$ n'a pas de position privilégiée en dehors de ce cadre asymptotique. Par conséquent, le vecteur θ_0 est supposé appartenir à l'espace convexe de Θ contenant la collection de vecteurs $(\theta_k)_{k=1}^{K}$.

La liste suivante présente des exemples d'initialisation pour l'algorithme

— $\theta_{0,\mathrm{al}}$ est une sélection aléatoire d'un vecteur de la collection $(\theta_k)_{k=1}^{K}$;
— $\theta_{0,\mathrm{mi}}$ est le vecteur de la collection $(\theta_k)_{k=1}^{K}$ tel que $l(\theta_0) \leqslant l(\theta_k)$
— $\theta_{0,\mathrm{mo}} = \frac{1}{K}\sum_{k=1}^{K}\theta_k$ est la moyenne arithmétique empirique de la collection de vecteurs $(\theta_k)_{k=1}^{K}$

Dans la suite de ce paragraphe, nous allons montrer que l'initialisation vérifie bien la condition d'être dans l'espace convexe. $\theta_{0,\mathrm{mi}}$ et $\theta_{0,\mathrm{al}}$ sont deux vecteurs de la collection $(\theta_k)_{k=1}^{K}$ qui est incluse dans l'espace convexe, alors ce sont deux vecteurs qui font partie de l'espace convexe. Par récurrence, il est possible de montrer que la moyenne arithmétique $\theta_{0,\mathrm{mo}}$ appartient au plus petit espace convexe contenant les points de la la collection de vecteurs $(\theta_k)_{k=1}^{K}$.

Par exemple, pour le modèle GGD. L'espace convexe de Θ contenant la collection de vecteurs $(\theta_k)_{k=1}^{K}$ sera le plus petit volume défini par $[\alpha_{\min};\alpha_{\max}]\times[\beta_{\min};\beta_{\max}]$. Avec $\alpha_{\min}$ le plus petit paramètre d'échelle α_k de la collection de vecteurs $(\theta_k)_{k=1}^{K}$ (respectivement $\alpha_{\max}$ est le plus grand paramètre d'échelle α_k,

$\beta_{\min}$ est le plus petit et $\beta_{\max}$ est le plus grand paramètre de forme β_k)

$$\begin{cases} \alpha_{\min} \leqslant \alpha_k \\ \alpha_{\max} \geqslant \alpha_k \\ \beta_{\min} \leqslant \beta_k \\ \beta_{\max} \geqslant \beta_k \end{cases} \qquad \forall k = 1, \ldots, K$$

$\theta_{0,\text{mo}}$ est un vecteur à deux éléments notés α_{mo} et β_{mo}. Par identification, il est possible d'exprimer ces deux éléments à partir des paramètres d'échelle α_k et de forme β_k :

$$\begin{cases} \alpha_{\text{mo}} = \frac{1}{K} \sum_{k=1}^{K} \alpha_k \\ \beta_{\text{mo}} = \frac{1}{K} \sum_{k=1}^{K} \beta_k \end{cases}$$

Ensuite, la définition des valeurs $\alpha_{\min}, \alpha_{\max}, \beta_{\min}$ et $\beta_{\max}$ donne les inégalités

$$\begin{cases} \alpha_{\min} = \frac{1}{K} \sum_{k=1}^{K} \alpha_{\min} \leqslant \alpha_{\text{mo}} \leqslant \frac{1}{K} \sum_{k=1}^{K} \alpha_{\max} = \alpha_{\max} \\ \beta_{\min} = \frac{1}{K} \sum_{k=1}^{K} \beta_{\min} \leqslant \beta_{\text{mo}} \leqslant \frac{1}{K} \sum_{k=1}^{K} \beta_{\max} = \beta_{\max} \end{cases}$$

ce qui indique que $\theta_{0,\text{mo}} \in [\alpha_{\min}; \alpha_{\max}] \times [\beta_{\min}; \beta_{\max}]$ appartient à l'espace convexe.

Les trois initialisations proposées vérifient la position demandée par rapport aux vecteurs de paramètres θ_k estimés au sens du maximum de vraisemblance. Chacune des initialisations devrait permettre à la méthode de recherche linéaire de converger vers le même minimum local. Mais ces trois initialisations ne sont pas égales. Il reste donc à faire un choix parmi ces trois initialisations, elles sont comparées suivant leur complexité calculatoire.

L'initialisation $\theta_{0,\text{mi}}$ nécessite d'ordonner les vecteurs θ_k afin de trouver celui qui minimise la fonction de coût. L'impact est plus fort lorsque le nombre K de vecteurs est élevé ou que la fonction de coût l a une complexité calculatoire importante. La moyenne arithmétique $\theta_{0,\text{mo}}$ propose une complexité calculatoire plus faible que l'initialisation $\theta_{0,\text{mi}}$, en effet seules des additions et une division par K sont requises. Enfin l'initialisation $\theta_{0,\text{al}}$ présente la complexité calculatoire la plus faible car ce n'est qu'un problème de sélection aléatoire d'index (voire utiliser systématiquement θ_1).

Considérons maintenant la complexité calculatoire de la méthode de recherche linéaire. La définition de $\theta_{0,\text{mi}}$ permet d'ordonner deux initialisations :

$$l(\theta_{0,\text{mi}}) \leqslant l(\theta_{0,\text{al}})$$

Si la fonction de coût l admet une valeur plus élevée, l'hypothèse de continuité de l sur Θ indique que la distance entre $\theta_{0,\text{al}}$ et $\bar{\theta}$ est plus élevée qu'entre $\theta_{0,\text{mi}}$ et $\bar{\theta}$. Comme le vecteur $\bar{\theta}$ est le minimum de la fonction l, $\theta_{0,\text{mi}}$ est une meilleure initialisation que $\theta_{0,\text{al}}$. La définition de la moyenne arithmétique $\theta_{0,\text{mo}}$ ne permet pas d'établir si la fonction de coût l admet une distance inférieure à la fonction de coût l évaluée en $\theta_{0,\text{mi}}$. Le paragraphe suivant présente l'initialisation $\theta_{0,\text{mo}}$.

La moyenne arithmétique $\theta_{0,\text{mo}}$ des vecteurs θ_k, est l'initialisation la moins cohérente du point de vue de l'application. Les vecteurs θ_k sont des estimateurs au sens du maximum de vraisemblance sur des variables aléatoires x_k qui suivent un modèle paramétrique. Comme le vecteur $\theta_{0,\text{mo}}$ n'est pas associé à la distribution d'une variable aléatoire x, alors la divergence de Jeffrey n'est pas définie entre $\theta_{0,\text{mo}}$ et un vecteur θ_k. C'est pour cette raison que la moyenne arithmétique pourrait ne pas convenir comme initialisation de la méthode de recherche linéaire car elle ne correspond pas au problème de la classification. Soit ϕ un changement de coordonnées affine entre Θ et Λ. Dès lors la moyenne arithmétique vérifie

$$\lambda_{0,\text{mo}} = \frac{1}{K}\sum_{k=1}^{K}\lambda_k = \frac{1}{K}\sum_{k=1}^{K}\phi(\theta_k) = \frac{1}{K}\sum_{k=1}^{K}(A\theta_k + B)$$

et, par propriété affine de la fonction de changement de coordonnées ϕ,

$$\lambda_{0,\text{mo}} = A\left(\frac{1}{K}\sum_{k=1}^{K}\theta_k\right) + B = \phi\left(\frac{1}{K}\sum_{k=1}^{K}\theta_k\right) = \phi(\theta_{0,\text{mo}})$$

avec A une matrice de passage entre les espaces Θ et Λ, B le vecteur reliant le 0 de Θ au 0 de Λ. L'équation précédente démontre simplement que la moyenne arithmétique $\theta_{0,\text{mo}}$ est aussi invariante à la paramétrisation. Néanmoins les valeurs prises par les éléments qui constituent $\theta_{0,\text{mo}}$ vérifient les contraintes existantes sur les paramètres, autrement dit, les calculs des mesures de dissimilarité peuvent être menés.

Chacune des initialisations proposées a un inconvénient. Celui de la moyenne arithmétique $\theta_{0,\text{mo}}$ par exemple ne permet pas de comparer la complexité calculatoire de la recherche linéaire qui serait obtenue. Validant ou invalidant l'initialisation $\theta_{0,\text{mo}}$ comme la plus mauvaise pour la complexité calculatoire de l'optimisation. Du point de vue complexité calculatoire de l'initialisation, la moyenne arithmétique est un bon compromis. Les deux autres initialisations représentent deux exemples extrêmes possibles pour la complexité calculatoire que ce soit pour l'initialisation ou pour la méthode, la moyenne arithmétique $\theta_{0,\text{mo}}$ va être utilisé comme initialisation.

Les méthodes de recherche linéaire (la méthode de descente de gradient, la méthode de Newton-Raphson et la méthode d'adaptation du gradient naturel) sont choisies pour estimer le vecteur $\bar{\theta}$ défini

comme le minimum de la fonction de coût l. Le problème est résumé sous forme d'optimisation avec l'équation (IV.8) présentée plus tôt dans ce chapitre. Ce qu'il faut noter dans cette équation, c'est la présence de contraintes sur les paramètres. Principalement, si les contraintes ne sont pas vérifiées, la fonction de coût ne peut pas être calculée. Dans la suite de cette section, nous allons présenter notre approche des contraintes et des études comparatives entre les trois méthodes de recherche linéaire.

3.2.3 Application numérique : la vitesse de convergence

L'estimation de la direction de descente p_j est sujette aux variations de la fonction de coût :

$$p_j = -B_j^{-1} \nabla l(\theta_j)$$

et cela peut impacter l'itération de façon à ce que θ_{j+1} ne vérifie plus les contraintes alors que θ_j le vérifiait. Par conséquent, une maitrise de la direction de descente p_j pour que la mise à jour vérifie les contraintes est développée. Par mesure de simplicité la formule de la direction de descente est conservée. La modification proposée a lieu sur la formule de l'itération :

$$\theta_{j+1} = \theta_j + q_j p_j \tag{IV.13}$$

avec q_j un réel strictement positif. Là où p_j est la direction de descente, q_j est la vitesse de descente. Une valeur supérieure à 1 permet d'avancer plus dans la direction définie alors qu'une valeur inférieure à 1 permet de faire une itération plus proche. Une fois introduite la notion de vitesse de convergence q_j, l'objectif qui est fixé est, sans perte de généralité, de donner une valeur maximale ϵ_J à la dissimilarité entre un vecteur θ_j et le vecteur issue de l'itération θ_{j+1}. Comme la variété est courbée, il est important d'utiliser une géométrie adaptée. La distance géodésique n'est pas utilisée en pratique mais il reste possible d'utiliser la mesure de dissimilarité de la fonction de coût l : la divergence de Jeffrey. Une mise à jour n'est possible que si l'inégalité :

$$J(\theta_j, \theta_{j+1}) \leqslant \epsilon_J$$

est vérifiée. Explicitons l'inégalité précédente au moyen de la formule de l'itération (IV.13) :

$$J(\theta_j, \theta_j + q_j p_j) \leqslant \epsilon_J$$

dans laquelle θ_j et p_j sont fixes, laissant juste le réel q_j comme variable.

Ce paragraphe montre quelle est la relation existant entre le produit de la vitesse de convergence q_j et de la direction de descente p_j avec la valeur de la divergence. Un développement limité à l'ordre 2 de la divergence de Jeffrey donne :

$$J(\theta_j, \theta_{j+1}) = (\theta_{j+1} - \theta_j)^T G(\theta_j)(\theta_{j+1} - \theta_j) + \mathcal{O}((\theta_{j+1} - \theta_j)^3).$$

Explicitons la formule au moyen de l'itération (IV.13)

$$J(\theta_j, \theta_j + q_j p_j) = (q_j p_j)^T G(\theta_j)(q_j p_j) + \mathcal{O}((q_j p_j)^3) = q_j^2 p_j^T G(\theta_j) p_j + \mathcal{O}(q_j^3 p_j^3).$$

La divergence de Jeffrey entre les itérations θ_{j+1} et θ_j dépend de la norme riemannienne de la direction de descente $p_j^T G(\theta_j) p_j$ et du carré de la vitesse de descente q_j. Ce développement limité ne reste valable que pour de très petites itérations et ne peut pas être utilisé pour évaluer la valeur numérique du déplacement entre l'itération θ_i et l'itération θ_{i+1}. Pour la suite de ce mémoire, nous présentons l'utilisation de la divergence $J(\theta_j, \theta_{j+1})$ pour évaluer le critère d'arrêt. Ce processus conduit à une complexité calculatoire plus élevée pour le calcul de la mise à jour que l'utilisation de la norme euclidienne.

Dans ce paragraphe est présenté un exemple concret. Soit une classe d'images texturées acquise depuis la base de donnée VisTex. L'ensemble des 16 images d'une même classe est utilisé. La GGD est utilisée pour la modélisation de la distribution des coefficients de la sous-bande verticale de la première échelle d'une décomposition en ondelettes stationnaires 2-D. θ_k est le vecteur de paramètres estimé au sens du maximum de vraisemblance pour l'image $k = 1, \ldots, 16$. Posons $\epsilon_J = 10^{-2}$. $\epsilon_J = 10^{-2}$ est une condition plus restrictive que $\epsilon_J = 10^{-1}$, utilisée pour mieux apprécier les différences entre trois méthodes de recherche linéaire aux performances proches. L'exemple est utilisé pour comparer les trois méthodes de recherche linéaire, suffisamment freinées pour apprécier les différences de comportement lors des itérations.

L'algorithme IV.5 présente la méthode de descente de gradient à pas fixe. Elle adapte l'algorithme d'origine (voir algorithme IV.2) en y ajoutant quelques notations. Le seuil ϵ_J est la valeur maximale admise pour une mise à jour. Elle est fixée à la valeur faible 10^{-2} afin de mieux apprécier la descente de gradient. Deux vecteurs θ_j et θ_{j+1} sont utilisés à la place d'un seul vecteur θ, en effet la mesure de dissimilarité de manière générale ne peut être réduite à la norme de direction de descente $\|p_j\|_2$. Conserver en mémoire le vecteur θ_j précédent l'itération est donc nécessaire pour évaluer le critère d'arrêt de la boucle itérative. Au sein de la boucle itérative, le réel q_j est mis à la valeur 10^{-2}. Encore une fois, cette valeur de départ est faible pour ne pas rajouter de vérifications supplémentaires sur les calcul de la vitesse

Algorithme IV.5 : Pseudo-code de la méthode de descente de gradient à pas fixe

Données : Un ensemble de K réalisations $(\theta_k)_{k=1}^K$, ∇l le gradient de la fonction de coût
Résultat : Le vecteur θ estimé de $\bar{\theta}$

1 $\epsilon_J \leftarrow 10^{-2}$;
2 $\theta_{j+1}, \theta_j \leftarrow \frac{1}{K}\sum_{k=1}^K \theta_k$;
3 **répéter**
4 $\quad \theta_j \leftarrow \theta_{j+1}$;
5 $\quad q_j \leftarrow 10^{-2}$;
6 $\quad$ **répéter**
7 $\quad\quad q_j \leftarrow \frac{2q_j}{3}$;
8 $\quad\quad \theta_{j+1} \leftarrow \theta_j - q_j \nabla l(\theta_j)$;
9 $\quad$ **jusqu'à** $J(\theta_j, \theta_{j+1}) \geqslant \epsilon_J$;
10 **jusqu'à** $J(\theta_j, \theta_{j+1}) \geqslant \epsilon$;
11 $\theta \leftarrow \theta_{j+1}$;

q_j (par exemple $\beta_{j+1} > 0$). De là nous commençons une seconde boucle itérative dédiée à réduire la valeur de la vitesse q_j afin d'obtenir une valeur acceptable pour l'itération θ_{i+1}. À la fin de la seconde boucle itérative, l'itération θ_{i+1} présente une dissimilarité faible avec l'itération θ_i. La première boucle itérative se poursuit jusqu'à ce que l'itération θ_{i+1} présente une évolution négligeable par rapport à l'itération θ_i.

Algorithme IV.6 : Pseudo-code de la méthode de Newton-Raphson à pas fixe

Données : Un ensemble de K réalisations $(\theta_k)_{k=1}^K$, ∇l le gradient et $\mathcal{H}\, l(\theta_j)$ la hessienne de la fonction de coût
Résultat : Le vecteur θ estimé de $\bar{\theta}$

1 $\epsilon_J \leftarrow 10^{-2}$;
2 $\theta_{j+1}, \theta_j \leftarrow \frac{1}{K}\sum_{k=1}^K \theta_k$;
3 **répéter**
4 $\quad \theta_j \leftarrow \theta_{j+1}$;
5 $\quad q_j \leftarrow 10^{-2}$;
6 $\quad$ **répéter**
7 $\quad\quad q_j \leftarrow 2/3 q_j$;
8 $\quad\quad \theta_{j+1} \leftarrow \theta_j - q_j (\mathcal{H}\, l(\theta_j))^{-1} \nabla l(\theta_j)$;
9 $\quad$ **jusqu'à** $J(\theta_j, \theta_{j+1}) \geqslant \epsilon_J$;
10 **jusqu'à** $J(\theta_j, \theta_{j+1}) \geqslant \epsilon$;
11 $\theta \leftarrow \theta_{j+1}$;

L'algorithme IV.6 présente la méthode de Newton-Raphson à pas fixe. L'algorithme est similaire à l'algorithme IV.5 de la méthode de descente de gradient à pas fixe en dehors de la ligne 8. Sur cette ligne 8 se trouve la formule d'itération θ_{i+1} par le calcul de la direction de descente p_j. Par extension, l'algorithme de la méthode d'adaptation du gradient naturel est similaire en dehors de la ligne 8. Cette ligne 8 se formule alors :

$$\theta_{j+1} \leftarrow \theta_j + q_j G(\theta_j)^{-1} \nabla l(\theta_j)$$

Les trois méthodes ayant une définition similaire, nous les formulerons maintenant à partir d'une matrice $B(\theta_j)$: 1) pour une descente de gradient, B est une matrice identité ; 2) pour une méthode de Newton-Raphson, $B(\theta_j) = \mathcal{H}l(\theta_j)$; 3) pour une adaptation du gradient naturel, $B(\theta_j) = -G(\theta_j)$.

À chaque itération j de l'algorithme, il est possible d'évaluer la fonction de coût $l(\theta_j)$. La figure IV.1 (b) montre la valeur de la fonction de coût par rapport au nombre d'itérations. Les courbes verte, rouge et bleue montrent la décroissance de la fonction de coût pour une suite construite avec, respectivement, la descente de gradient, la méthode de Newton-Raphson et l'adaptation de gradient naturel à pas fixe. Soient θ_j^{∇} l'élément de la suite respectant la descente de gradient à pas fixe, θ_j^{H} l'élément de la suite respectant la méthode de Newton-Raphson à pas fixe, θ_j^{G} l'élément de la suite respectant l'adaptation de gradient naturel à pas fixe. Pour chaque itération j il est possible de montrer que $l(\theta_j^{\nabla}) \geqslant l(\theta_j^{H}) \geqslant l(\theta_j^{G})$. Cela montre que l'adaptation de gradient naturel offre une convergence plus rapide que les deux méthodes précédentes. L'ordonnée du graphique affiche la valeur de la fonction de coût que nous cherchons à minimiser, c'est la dérivée de la fonction de coût qui est annulée. Il faut également noter une différence de comportement de la suite construite par descente de gradient par rapport aux deux autres méthodes. Après avoir ralenti sa convergence, l'algorithme accélère de nouveau jusqu'à atteindre la valeur de l vers laquelle ont convergé les deux autres méthodes.

La figure IV.1 (a) montre dans le plan (α, β) le chemin suivi par chacune des trois méthodes d'optimisation. Les courbes verte, rouge et bleue correspondent aux courbes contenant respectivement les suites $(\theta_j^{\nabla})_{j=1}^{\infty}$, $(\theta_j^{H})_{j=1}^{\infty}$ et $(\theta_j^{G})_{j=1}^{\infty}$ construites avec, respectivement, la descente de gradient, la méthode de Newton-Raphson et l'adaptation de gradient naturel à pas fixe. Les courbes relient la moyenne arithmétique $\theta_0 = \frac{1}{K}\sum_{k=1}^{K} \hat{\theta}_{k,i}$ utilisée comme initialisation de chaque algorithme à un certain point $\theta_{10^5}^{\nabla}$, $\theta_{10^5}^{H}$ et $\theta_{10^5}^{G}$. Il n'y a, visuellement, aucune différence entre $\theta_{10^5}^{\nabla}$, $\theta_{10^5}^{H}$ et $\theta_{10^5}^{G}$ puisque les trois courbes semblent relier les même deux points. La courbe bleue dessine une droite entre le point de départ et le point d'arrivée, la courbe rouge présente une légère courbure de cette droite. La courbe verte (construite par descente de gradient) présente deux modes de fonctionnement, dans le premier mode, la position en α est fixe et la valeur du β diminue. Dans le second mode, le couple de paramètres (α, β) converge linéairement vers le point d'arrivée.

La première application numérique vient d'illustrer l'intérêt particulier à l'égard de l'adaptation du gradient naturel. La suite $(\theta_j)_j$ ainsi construite converge plus rapidement et vers le même estimé $\bar{\theta}$ que les deux autres implémentations. La complexité calculatoire de la méthode d'adaptation du gradient naturel devrait être équivalente à celle de la méthode de Newton-Raphson. En effet, les deux algorithmes

utilisent une matrice $B(\theta_j)$ non égale à l'identité. Pour la suite de ce mémoire, la méthode d'adaptation du gradient naturel est utilisé afin d'assurer une convergence rapide vers l'estimé $\bar{\theta}$ du minimum de la fonction de coût l.

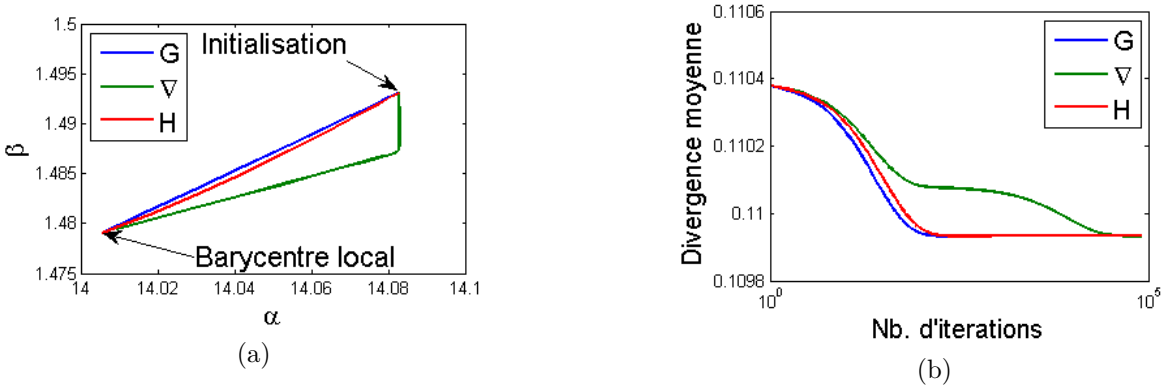
3.2.4 Application numérique : l'initialisation

L'estimé du minimum de la fonction de coût l est obtenu localement au moyen d'un algorithme d'optimisation. Pour converger vers cet estimé, il est donc important de choisir une initialisation qui soit proche de l'estimé. Une nouvelle application numérique est mise en place pour montrer que la notion de proximité de cette initialisation est très légère. Cela peut remettre en question le caractère local de l'estimé $\bar{\theta}$ du minimum de la fonction de coût l.

Cette section illustre l'importance de l'initialisation dans la méthode de recherche linéaire. Par exemple, considérons la méthode d'adaptation du gradient naturel. Le problème (IV.8) est solutionné avec 7 initialisations différentes $\theta_{0,A} = (0.7; 1.8), \theta_{0,B} = (0.7; 1.2), \theta_{0,C} = (0.4; 1.2), \theta_{0,D} = (0.4; 1.8), \theta_{0,E} = (0.5235; 1.35), \theta_{0,F} = (0.6136; 1.493)$ et $\theta_{0,O} = \theta_{0,\text{mo}}$ la moyenne arithmétique. Les initialisations sont de 2 natures, soit positionnées en périphérie du nuage $(\theta_k)_{k=1}^K$ pour $\theta_{0,E}$ et $\theta_{0,F}$, soit extérieures au nuage pour $\theta_{0,A}, \theta_{0,B}, \theta_{0,C}$ et $\theta_{0,D}$. Dans le second cas, les initialisations ne sont pas de bonnes approximations du barycentre $\bar{\theta}$ dont la position est supposée intérieure au nuage. De plus, les initialisations en périphéries sont positionnées le long de l'axe principal du nuage.

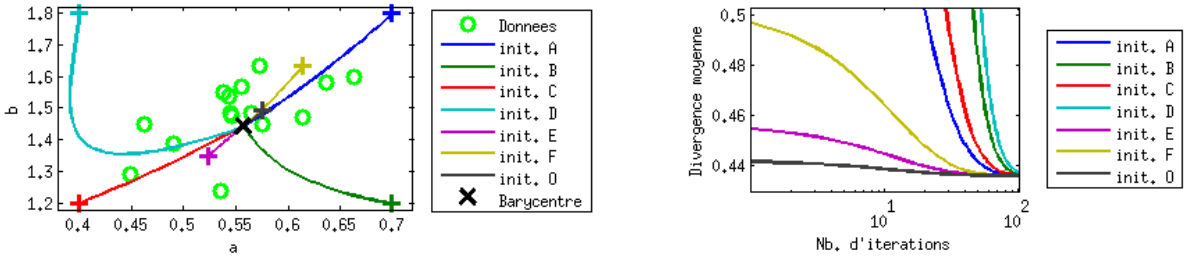
La figure IV.2.(a) présente dans le plan $(\alpha; \beta)$ les vecteurs paramétriques $(\theta_k)_{k=1}^K$ avec des cercles verts clair, les vecteurs utilisés pour l'initialisation au moyen de plus colorés, les itérations pour chacune des initialisations par une ligne continue colorée et le barycentre $\bar{\theta}$ résultant de la méthode d'adaptation du gradient naturel avec une croix noire. La figure IV.2.(b) présente une courbe colorée représentant l'évolution de la fonction de coût $l(\theta_j)$ suivant les itérations j pour une certaine initialisation θ_0. Le code couleur utilisé dans les figures IV.2.(a) et IV.2.(b) est donné par les initialisations : $\theta_{0,A}$ pour la couleur bleue, $\theta_{0,B}$ pour la couleur verte, $\theta_{0,C}$ pour la couleur rouge, $\theta_{0,D}$ pour la couleur cyan, $\theta_{0,E}$ pour la couleur magenta, $\theta_{0,F}$ pour la couleur jaune et $\theta_{0,O}$ pour la couleur noire.

La courbe noire dans la figure IV.2.(b) présente l'évolution de la fonction de coût l avec la moyenne arithmétique $\theta_{0,\text{mo}}$ comme initialisation. Cette implémentation est similaire à celles vues dans les figures IV.1.(b) et IV.4.(a), autrement dit il s'agit de la courbe témoin. Premier constat, pour toute les autres initialisations, la fonction de coût l admet une valeur supérieure à $l(\theta_{0,\text{mo}})$. Il est alors compréhensible que la convergence de l'algorithme soit plus lente. Parmi les courbes tracées, seules les courbes magenta,



(a) (b)

Figure IV.1 – Étude comparative entre trois méthodes d'optimisation. Initialisation avec le barycentre arithmétique empirique $\theta_0 = \frac{1}{K}\sum_{k=1}^{K}\theta_{k,i}$. Pas de mise a jour fixé à $\epsilon_0 = 0,02$. Le code couleur choisi est bleu pour l'adaptation du gradient naturel, vert pour la descente de gradient et rouge pour la méthode de Newton-Raphson. La figure (a) montre la position géométrique des mises à jour successives θ_j, alors que la figure (b) montre la fonction de coût suivant le nombre d'itérations j de l'algorithme.



(a) (b)

Figure IV.2 – Méthode d'adaptation du gradient naturel, étude comparative pour 7 initialisations. L'initialisation avec la moyenne arithmétique empirique $\theta_0 = \frac{1}{K}\sum_{k=1}^{K}\theta_{k,i}$ est noté O puis associé à la couleur noire (respectivement $\theta_0 = (0.7; 1.8)$ pour A bleu, $\theta_0 = (0.7; 1.2)$ pour B vert, $\theta_0 = (0.4; 1.2)$ pour C rouge, $\theta_0 = (0.4; 1.8)$ pour D cyan , $\theta_0 = (0.5235; 1.35)$ pour E magenta, $\theta_0 = (0.6136; 1.493)$ pour F jaune). Pas de mise à jour adaptatif avec $\epsilon_J = 0,02$. La figure (a) montre la position géométrique des mises à jour successives θ_j, alors que la figure (b) montre la fonction de coût suivant le nombre d'itérations j de l'algorithme.

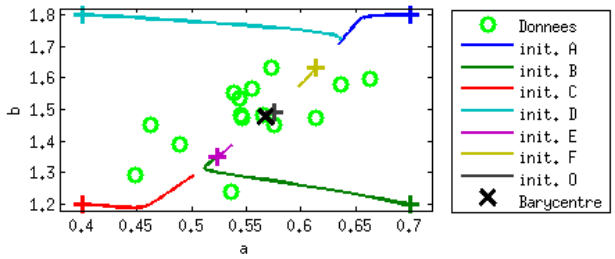


Figure IV.3 – Méthode de descente du gradient, étude comparative pour 7 initialisations. L'initialisation avec la moyenne arithmétique empirique $\theta_0 = \frac{1}{K}\sum_{k=1}^{K}\theta_{k,i}$ est noté O puis associé à la couleur noire (respectivement $\theta_0 = (0.7; 1.8)$ pour A bleu, $\theta_0 = (0.7; 1.2)$ pour B vert, $\theta_0 = (0.4; 1.2)$ pour C rouge, $\theta_0 = (0.4; 1.8)$ pour D cyan , $\theta_0 = (0.5235; 1.35)$ pour E magenta, $\theta_0 = (0.6136; 1.493)$ pour F jaune). Pas de mise à jour adaptatif avec $\epsilon_J = 0,02$. La figure montre la position géométrique des 30 mises à jour successives θ_j.

jaune, et noire sont dans le cadre défini. La courbe noire est associée à l'initialisation $\theta_{0,\text{mo}}$ comme dit précédemment, les deux autres courbes correspondent respectivement avec l'initialisation $\theta_{0,E}$ et $\theta_{0,F}$ qui sont les deux initialisations périphériques au nuage. Pour les initialisations externes au nuages, la méthode d'adaptation du gradient naturel utilise plus de 20 itérations avant de vérifier $l(\theta_j) \leqslant 0.5$, ce qui indique que l'algorithme fait décroitre la fonction de coût l.

La figure IV.3 contient les même informations que la figure IV.2.(a), la différence se fait au niveau de la méthode de recherche linéaire utilisée : respectivement la méthode de descente du gradient et la méthode d'adaptation du gradient naturel. Amari et Douglas [109] proposent d'utiliser la géométrie de la variété afin d'augmenter la vitesse de convergence. Prenons exemple sur l'initialisation $\theta_{0,D}$ de couleur cyan. La direction de descente empruntée par la méthode de descente de gradient $-\nabla l(\theta_{0,D})$ est perpendiculaire à la direction de descente empruntée par la méthode d'adaptation du gradient naturel $G(\theta_{0,D})^{-1}\nabla l(\theta_{0,D})$. Néanmoins, aucune des directions utilisées n'est dirigée vers le barycentre $\bar{\theta}$ représenté par une croix noire, conduisant une évolution très courbée de la suite de vecteurs (θ_j). Inversement, pour les initialisations $\theta_{0,B}, \theta_{0,E}$ et $\theta_{0,F}$ aucune différence dans la direction de descente à l'origine n'est à constater. La méthode d'adaptation du gradient naturel se dirige vers le barycentre de manière plus directe qu'avec la méthode de descente du gradient.

Cette section montre que des choix de variables devaient être réalisés afin d'envisager une mise en œuvre d'un algorithme d'optimisation. Après la discussion sur ses deux variables, cette section présente des applications numériques pour valider l'algorithme d'optimisation ainsi créé. L'algorithme doit utiliser l'adaptation du gradient naturel avec comme initialisation la moyenne géométrique $\theta_{0,\text{mo}}$ des vecteurs paramétriques. Cette section a clarifié le propos autour de l'algorithme d'optimisation et montré des résultats, dans la section suivante nous innovons sur cette base saine d'algorithme d'optimisation pour convenir au cadre applicatif des images texturées.

3.3 Estimation du barycentre : La descente de gradient projeté

La méthode proposée consiste à maitriser les itérations pour que la dissimilarité entre l'itération θ_{j+1} et l'itération θ_j décroisse. Pour ce faire, l'équation (IV.13) est proposée pour ne pas modifier la valeur de la direction de descente calculée mais pour modifier la vitesse de descente q_j. La recherche de cette vitesse optimale ne peut donc ce faire que pour des itérations respectant les contraintes c. Les algorithmes précédents n'ont pas présenté de vérification des contraintes en fixant une vitesse q_j relativement faible pour l'initialisation.

Dans le cadre applicatif, un temps d'apprentissage fini est recherché pour une base de données d'apprentissage volumineuse. Sans remettre en cause la section précédente, proposer de plus grandes directions de descente dans la méthode de recherche linéaire, permettrait de converger plus rapidement. Utiliser de plus grandes directions de descente influe alors sur le respect des contraintes. Jusqu'à maintenant les contraintes étaient respectées par les paramètres restrictifs de l'algorithme. Apporter de la souplesse à ces mêmes paramètres est possible seulement si le respect des contraintes est directement restreint à chaque itération.

L'itération θ_{i+1} doit vérifier 2 restrictions. La première de ces restrictions est le respect des contraintes c du problème défini par l'équation (IV.8). Les algorithmes de recherche linéaire présentés ne prennent pas en compte ces contraintes et une vérification interne à la méthode est requise. La seconde restriction est le respect de l'inégalité sur la fonction de coût l :

$$l(\theta_{i+1}) \leqslant l(\theta_i)$$

La seconde restriction assure une convergence de l'algorithme, mais peut aussi le faire converger plus rapidement. De plus, la première restriction permet de vérifier la seconde restriction puisque la fonction de coût l n'est pas défini au delà des contraintes c.

Pour vérifier la première restriction, l'itération donnée par l'équation (IV.13) est suffisante. Il suffit que les contraintes c sont vérifiées par l'itération θ_{i+1} en plus de la valeur maximale pour la dissimilarité ϵ_J. Supposons que les contraintes soient vérifiées pour θ_j, alors, par continuité de ces contraintes c, il existe un voisinage V_j de θ_j dans lequel tout vecteur θ vérifie les contraintes. Cela indique qu'il existe une vitesse de descente q_j maximum.

Pour vérifier la seconde restriction, il est encore possible de réduire la vitesse jusqu'à obtenir l'inégalité sur la fonction de coût. Ne pas vérifier la seconde restriction signifie que l'itération θ_{i+1} a une dissimilarité acceptable par rapport à l'itération θ_j mais augmente la fonction de coût. La proposition est alors de réduire la valeur de dissimilarité maximale ϵ_J de moitié.

L'algorithme IV.7 montre la méthode de recherche linéaire à pas adaptatif. Cette méthode prend une entrée supplémentaires par rapport à l'algorithme à pas fixe IV.6. En lieu et place de la matrice hessienne $\mathcal{H}\, l$ de la fonction de coût, c'est une fonction B qui associe une matrice à un vecteur θ_j. La fonction B renvoie une matrice identité (respectivement la matrice hessienne $\mathcal{H}\, l$ de la fonction de coût et l'opposé de la matrice d'information de Fisher $-G$) pour une méthode de descente de gradient (respectivement pour une méthode de Newton-Raphson et une méthode d'adaptation du gradient naturel). La fonction

Algorithme IV.7 : Pseudo-code de la méthode de recherche linéaire à pas adaptatif

Données : Un ensemble de K réalisations $(\theta_k)_{k=1}^K$, ∇l le gradient de la fonction de coût, B une fonction renvoyant la matrice pour la méthode de recherche linéaire et ϵ_J la dissimilarité maximum entre deux itérations

Résultat : Le vecteur θ estimé de $\bar{\theta}$

$\theta_{j+1}, \theta_j \leftarrow \frac{1}{K}\sum_{k=1}^{K}\theta_k$;
répéter
  $\theta_j \leftarrow \theta_{j+1}$;
  $q_j \leftarrow \frac{3}{2}$; /* Distance parcourue seuillée */
  répéter
    répéter
      $q_j \leftarrow \frac{2q_j}{3}$;
      $\theta_{j+1} \leftarrow \theta_j - q_j B(\theta_j)^{-1}\nabla l(\theta_j)$; /* 1ière restriction */
    jusqu'à $\exists$ *une contrainte* $c(\theta_{j+1})$ *non vérifiée ou* $J(\theta_j, \theta_{j+1}) \geqslant \epsilon_J$;
    si $l(\theta_{j+1}) > l(\theta_j)$ **alors**
      $\epsilon_J \leftarrow \epsilon_J/2$; /* 2nde restriction */
    fin
  jusqu'à $l(\theta_{j+1}) > l(\theta_j)$;
jusqu'à $J(\theta_j, \theta_{j+1}) \geqslant \epsilon$;
$\theta \leftarrow \theta_{j+1}$;

B « fluidifie » la présentation des algorithmes et propose une solution applicative pour un programme qui sait s'adapter à la méthode de recherche. La nouvelle entrée est la valeur maximale admise pour la dissimilarité ϵ_J. À partir des connaissances *a priori* sur la fonction de coût l il est possible de fixer une valeur ϵ_J assurant la plus petite vitesse de convergence.

L'algorithme IV.7 présente 3 boucles itératives imbriquées. La première boucle itérative vérifie la condition de sortie qui est une dissimilarité insuffisante entre les itérations θ_{i+1} et θ_i. Cette boucle itérative vient directement de l'algorithme IV.2, qui est commune à toute méthode de recherche linéaire. La deuxième boucle itérative assure que la seconde restriction est vérifiée, autrement dit que la fonction de coût l est décroissante avec l'itération θ_{i+1}. Enfin la troisième boucle itérative est similaire à la deuxième boucle itérative de l'algorithme IV.6. Comme évoqué précédemment, au cours de cette boucle, la vitesse q_j est fixée de façon à avoir une itération θ_{j+1} suffisamment similaire à θ_j. En sortie des trois boucles itératives se trouve un vecteur θ qui est similaire à $\bar{\theta}$ comme élément de la suite d'itérations $(\theta_j)_{j=1}$.

L'algorithme IV.7 proposé présente une boucle de plus par rapport à l'algorithme IV.6, ainsi qu'un plus grand nombre de comparaisons. Le fait de converger plus rapidement en laissant l'algorithme faire de plus larges pas influe sur la complexité calculatoire obtenue. La suite de cette section montre la vitesse de convergence de l'algorithme, par rapport à la première boucle itérative.

Les figures IV.4.(a) et IV.4.(b) page 120 présentent une comparaison entre 4 implémentations d'une

méthode d'adaptation du gradient naturel. La méthode d'adaptation du gradient naturel est utilisée pour trouver un minimum local de la fonction de coût l. Les 4 implémentations ont pour initialisation la moyenne arithmétique. La première implémentation considérée suit l'algorithme IV.6 à pas fixe. La courbe de couleur bleue représente dans la figure IV.4.(a) l'évolution de la valeur $l(\theta_j)$ en fonction de l'entier j pour l'implémentation à pas fixe. De même, la courbe bleue dans la figure IV.4.(b) représente l'évolution de l'écart relatif :

$$\frac{|l(\theta_{j+1}) - l(\theta_j)|}{|l(\theta_j)|}$$

en fonction de l'itération j pour cette même implémentation à pas fixe. L'écart relatif fait ressortir les différences entre la valeur de la fonction de coût évaluée en l'itération θ_j et la valeur évaluée en l'itération θ_{j+1}. En considérant un développement limité de la fonction de coût l nous écrivons :

$$\frac{|l(\theta_{j+1}) - l(\theta_j)|}{|l(\theta_j)|} = \frac{|(\theta_{j+1} - \theta_j)l'(\theta_j) + \mathcal{O}((\theta_{j+1} - \theta_j)^2)|}{|l(\theta_j)|}.$$

Trois implémentations de l'algorithme IV.7 à pas adaptatif sont proposées avec une valeur de dissimilarité maximum croissante : $\epsilon_J \in \{1, 2, 9\}$. Dans la figure IV.4.(a), une courbe verte (respectivement rouge et cyan) représente l'évolution de la valeur prise par la fonction de coût $l(\theta_j)$ en fonction de l'itération j pour une implémentation à pas adaptatif avec $\epsilon_J = 1$ (respectivement $\epsilon_J = 2$ et $\epsilon_J = 9$). Une implémentation à pas adaptatif avec $\epsilon_J = 1$ ou 2, montre que la fonction de coût l atteint son minimum en moins de 4 itérations. Dans les mêmes conditions, l'algorithme à pas fixe a requis 200 itérations avant d'atteindre le minimum de la fonction de coût l. La courbe cyan met près d'une vingtaine d'itérations à atteindre le minimum. Cette courbe correspond à un algorithme à pas adaptatif avec $\epsilon_J = 9$ qui est une condition encore plus lâche qu'avec $\epsilon_J \in \{1, 2\}$. La divergence de Jeffrey entre deux itérations θ_{j+1} et θ_j est quadratique en la vitesse q_j. Par conséquent, laisser possible une dissimilarité plus large entre deux itérations ne permet pas d'assurer une convergence en un nombre fini d'itérations. Les raisons de cette instabilité sont présentées en annexe 3, en résumé la fonction de coût l n'est pas convexe en le paramètre de forme β. Dans la figure IV.4.(b), une courbe verte (respectivement rouge et cyan) représente l'évolution de l'écart relatif :

$$\frac{|l(\theta_{j+1}) - l(\theta_j)|}{|l(\theta_j)|}$$

en fonction de l'itération j pour une implémentation à pas adaptatif avec $\epsilon_J = 1$ (respectivement $\epsilon_J = 2$ et $\epsilon_J = 9$). Lorsque $\theta_2 = \theta_1$, la fonction de coût l prend une valeur égale et l'écart relatif est nul. Dans une représentation logarithmique comme dans la figure IV.4 la valeur 0 est non définie et donc non affichée.

Les courbes rouge, verte et cyan ne sont donc pas tracées lorsqu'elles ont convergé. Comme vu dans la figure IV.4.(a), les courbes rouge et verte présentent une grande vitesse de convergence comparées aux courbes associées aux 2 autres implémentations. L'implémentation à pas fixe présente une vitesse de convergence croissante au fil des itérations, et donc du gain de similarité avec $\bar{\theta}$.

Nous avons montré les différences en complexité calculatoire dans notre publication de 2012 [63]. ∎

3.4 Exemple : La distribution GGD

La gaussienne généralisée (GGD) est définie par un paramètre d'échelle α strictement positif et un paramètre de forme β strictement positif. La densité de probabilité explicite d'une GGD s'écrit :

$$p(x|\alpha,\beta) = \frac{1}{2\alpha\Gamma(1/\beta+1)} \exp\left\{-\left(\frac{|x|}{\alpha}\right)^{\beta}\right\}$$

avec Γ la fonction Gamma définie par la formule $\Gamma(z) = \int_{\mathbb{R}^+} t^{z-1}e^{-t}.dt$.

Choy et Tong [44] se sont intéressés à la classification d'images texturées. Ils proposent d'adapter le travail précédent de Do et Vetterli[4] sur la recherche d'image basée sur le contenu (CBIR) à une classification basée distribution caractéristique (1-CB). Les deux applications sont très différentes : d'un côté le CBIR renvoie, pour une image requête, l'ensemble des N_I images d'apprentissage les plus similaires ; d'un autre côté le 1-CB renvoie, pour une image requête, la classe de texture la plus vraisemblable. Choy et Tong [44] proposent d'estimer la distribution caractéristique de chaque texture et de l'utiliser en classification. Par mesure de simplicité, le modèle paramétrique de la distribution caractéristique est celui utilisé pour la distribution de la variable aléatoire x. Comme décrit au premier chapitre, une GGD est représentée dans l'espace de paramètres par un vecteur $\bar{\theta}$. La distribution caractéristique est la GGD associée au vecteur $\bar{\theta}^R$ minimisant la divergence de Kullback-Leibler à droite avec les vecteurs θ_k. Par rapport aux notations définies dans ce document, le vecteur $\bar{\theta}^R$ est un barycentre orienté à droite selon la divergence de Kullback-Leibler. De plus, Choy et Tong [44] démontrent l'existence d'une telle distribution caractéristique, sans s'attarder sur son unicité.

Au début de ce chapitre nous avons défini la fonction de coût l. Le vecteur $\bar{\theta}$ qui minimise l représente la distribution caractéristique d'une texture qui doit être estimée. Pour ce faire, une méthode de recherche linéaire est proposée, ce qui nécessite de calculer les dérivées d'ordre 1 et 2 de la fonction de coût l. Par

Posons	$A_{L,k} = \left(\frac{\alpha}{\alpha_k}\right)^{\beta_k} \frac{\Gamma((\beta_k+1)/\beta)}{\Gamma(1/\beta)}$ $A_{R,k} = \left(\frac{\alpha_k}{\alpha}\right)^{\beta} \frac{\Gamma((\beta+1)/\beta_k)}{\Gamma(1/\beta_k)}$
Divergence	$J(\theta, \theta_k) = A_{L,k} + A_{R,k} - \frac{1}{\beta} - \frac{1}{\beta_k}$
Dérivée en α	$\frac{\partial J(\theta,\theta_k)}{\partial \alpha} = \frac{1}{\alpha}\left(\beta_k A_{L,k} + \beta A_{R,k}\right)$
Dérivée en β	$\frac{\partial J(\theta,\theta_k)}{\partial \beta} = -\frac{A_{L,k}}{\beta^2}\left[(\beta_k+1)\Psi\left(\frac{\beta_k+1}{\beta}\right) - \Psi\left(\frac{1}{\beta}\right)\right] - \frac{1}{\beta^2} +$ $A_{R,k}\left[\frac{1}{\beta_k}\Psi\left(\frac{\beta+1}{\beta_k}\right) + \log\left\{\frac{\alpha_k}{\alpha}\right\}\right]$
Fonction digamma	$\Psi(z) = \frac{\Gamma'(z)}{\Gamma(z)} = D\log\{\Gamma(z)\}$

Tableau IV.1 – Divergence de Jeffrey et dérivées pour une vraisemblance GGD

exemple, la dérivée partielle de la fonction de coût l suivant le paramètre d'échelle α est donné par :

$$\frac{\partial l(\bar{\theta})}{\partial \bar{\alpha}} = \sum_{k=1}^{K} \frac{\partial J(\theta_k, \bar{\theta})}{\partial \bar{\alpha}} = \frac{1}{\bar{\alpha}} \sum_{k=1}^{K} \left(\beta_k \left(\frac{\bar{\alpha}}{\alpha_k} \right)^{\beta_k} \frac{\Gamma((\beta_k+1)/\bar{\beta})}{\Gamma(1/\bar{\beta})} + \bar{\beta} \left(\frac{\alpha_k}{\bar{\alpha}} \right)^{\bar{\beta}} \frac{\Gamma((\bar{\beta}+1)/\beta_k)}{\Gamma(1/\beta_k)} \right) \tag{IV.14}$$

Plus généralement, la fonction de coût l est une somme de divergences de Jeffrey entre les vecteurs $\bar{\theta}$ et θ_k. L'opération de calcul de dérivée est linéaire, autrement dit la dérivée d'une somme de fonctions égale la somme des dérivées. Le calcul de la dérivée est alors réalisé sur la divergence de Jeffrey entre les vecteurs $\bar{\theta}$ et θ_k indépendamment des autres divergences. Le tableau IV.1 donne la formule des deux dérivées partielles de la divergence de Jeffrey, la dérivée partielle de la fonction de coût l en $\bar{\beta}$ s'exprime :

$$\frac{\partial l(\bar{\theta})}{\partial \bar{\beta}} = \sum_{k=1}^{K} \frac{\partial J(\theta_k, \bar{\theta})}{\partial \bar{\beta}}$$

Une condition nécessaire de premier ordre pour que le vecteur $\bar{\theta}$ minimise la fonction de coût l est que le gradient ∇l de la fonction s'annule en $\bar{\theta}$. La dérivée partielle en le paramètre de forme β de la fonction de coût l est complexe. L'estimation du paramètre de forme moyen $\bar{\beta}$ ne peut pas passer par une formulation explicite. Ceci appuie l'intérêt qui est porté à la méthode de recherche linéaire du minimum de la fonction de coût l.

Parmi les méthodes de recherche linéaire présentées dans la section 3.1, les méthodes de Newton-Raphson et d'approximation du gradient naturel se calculent à partir d'une matrice. Théoriquement, chacune de ces méthodes propose une vitesse de convergence supérieure à la méthode de descente de gradient.

Premièrement la méthode de Newton-Raphson nécessite de calculer la matrice hessienne $\mathcal{H}l$ de la fonction de coût l. La dérivée est linéaire, par conséquent la matrice hessienne est définie comme la somme des matrices hessiennes des fonctions $\theta \to J(\theta, \theta_k)$, pour tout $k = 1, \ldots, K$. Le tableau IV.2 présente

l'ensemble des 3 formules explicites utilisées pour calculer la matrice hessienne. La matrice hessienne de la fonction de coût l pour une distribution GGD est une matrice 2x2 symétrique, les éléments en dehors de la diagonale sont égaux :

$$\frac{\partial^2 J(p(\cdot \mid \theta), p(\cdot \mid \theta_k))}{\partial\alpha\partial\beta} = \frac{\partial^2 J(p(\cdot \mid \theta), p(\cdot \mid \theta_k))}{\partial\beta\partial\alpha}.$$

Deuxièmement, la méthode d'adaptation du gradient naturel nécessite de calculer la matrice d'information de Fisher associée au modèle GGD. De part sa définition, la matrice d'information de Fisher est indépendante de la collection de vecteurs $(\theta_k)_{k=1}^K$. Le tableau IV.3 donne explicitement la forme de la matrice d'information de Fisher.

La GGD est un exemple de modèle univarié. Dans cet exemple, le gradient ∇l et la hessienne $\mathcal{H}l$ de la fonction de coût l ont été explicités. De plus, la formule explicite de la matrice d'information de Fisher pour la GGD a été donnée. Pour rappel, il s'agit de la matrice qui représente la géométrie riemannienne de la variété paramétrique.

Avec l'ensemble de ces formules, il est possible de mettre en place une méthode de recherche linéaire. Nous utilisons l'exemple de la GGD dans différentes publications [63, 65, 66]. La section suivante évoque ▮ présente un exemple avec un modèle stochastique multivarié.

3.5 Exemple : le modèle SIRV

Un modèle multivarié peut être utilisé en classification d'images texturées. Ce dernier est utilisé pour modéliser les dépendances spatiales, couleur ou dépendances inter-bandes. Un modèle univarié, ne modélisant pas ce type de dépendances, ne permet pas d'obtenir des performances de classifications équivalentes à l'utilisation de modèle multivarié. Dans ce document, le modèle multivarié utilisé est un modèle Spherically Invariant Random Vector (SIRV) [56] (voir page 41) avec une distribution Weibull pour le multiplieur.

La divergence de Kullback-Leibler ne peut pas être définie entre deux SIRV. C'est pour cette raison que la loi jointe $\vec{y} = (\tau, \vec{g})$ est introduite. Le multiplieur τ et la distribution gaussienne multivariée $\vec{g}$ sont supposées indépendantes, la densité de y s'exprime simplement comme :

$$Y(\vec{y} \mid \theta) = \phi(\vec{g} \mid M) p_\tau(\tau \mid a, b)$$

avec $\theta = (\Sigma, a, b)$ le vecteur de paramètres de la loi jointe y. La divergence de Kullback-Leibler est

Matrice hessienne	$\mathcal{H}J(p(\cdot \mid \theta), p(\cdot \mid \theta_k)) = \begin{pmatrix} \partial_{\alpha^2} J & \partial_{\alpha,\beta} J \\ \partial_{\alpha,\beta} J & \partial_{\beta^2} J \end{pmatrix}$
Dérivée en α^2	$\frac{\partial^2 J(p(\cdot\|\theta),p(\cdot\|\theta_k))}{\partial \alpha^2} = \frac{1}{\alpha^2}\left((\beta_k^2 - \beta_k)A_{L,k} + (\beta^2+\beta)A_{R,k}\right)$
Dérivée en $\alpha\beta$	$\frac{\partial^2 J(p(\cdot\|\theta),p(\cdot\|\theta_k))}{\partial \alpha \partial \beta} = \frac{A_{L,k}\beta_k}{\alpha\beta^2}\left[-(\beta_k+1)\Psi\left(\frac{\beta_k+1}{\beta}\right) + \Psi\left(\frac{1}{\beta}\right)\right] + \frac{A_{R,k}}{\alpha\beta_k}\left[-\beta\Psi\left(\frac{\beta+1}{\beta_k}\right) + \beta_k\left(\beta\log\left\{\frac{\alpha_k}{\alpha}\right\} - 1\right)\right]$
Dérivée en β^2	$\frac{\partial^2 J(p(\cdot\|\theta),p(\cdot\|\theta_k))}{\partial \beta^2} = \frac{A_{L,k}}{\beta^4}\left[(\beta_k+1)^2\left(\Psi_1\left(\frac{\beta_k+1}{\beta}\right) + \left(\Psi\left(\frac{\beta_k+1}{\beta}\right)\right)^2\right) + 2(\beta_k+1)\left(\beta - \Psi\left(\frac{1}{\beta}\right)\right)\Psi\left(\frac{\beta_k+1}{\beta}\right) + \left(\Psi\left(\frac{1}{\beta}\right)\right)^2 - \Psi_1\left(\frac{1}{\beta}\right) - 2\beta\Psi\left(\frac{1}{\beta}\right)\right] + \frac{A_{R,k}}{\beta_k^2}\left[\Psi_1\left(\frac{\beta+1}{\beta_k}\right) + \left(\Psi\left(\frac{\beta+1}{\beta_k}\right)\right)^2 + \left(\beta_k\log\left\{\frac{\alpha_k}{\alpha}\right\}\right)^2 + 2\beta_k\log\left\{\frac{\alpha_k}{\alpha}\right\}\Psi\left(\frac{\beta+1}{\beta_k}\right)\right]$
Fonction trigamma	$\Psi_1(z) = D\Psi(z) = D^2\log\{\Gamma(z)\}$

Tableau IV.2 – Dérivées d'ordre deux de J pour une vraisemblance GGD. Les variables ($A_{L,k}$ et $A_{R,k}$) manquantes ici sont données dans le tableau IV.1

Matrice d'information de Fisher	$G(\theta) = \begin{pmatrix} g_{\alpha,\alpha}(\theta) & g_{\alpha,\beta}(\theta) \\ g_{\alpha,\beta}(\theta) & g_{\beta,\beta}(\theta) \end{pmatrix}$
Suivant α, α	$g_{\alpha,\alpha}(\theta) = \frac{\beta}{\alpha^2}$
Suivant α, β	$g_{\alpha,\beta} = -\frac{1}{\alpha\beta}\left[\Psi\left(\frac{1}{\beta}\right) + \beta + 1\right]$
Suivant β, β	$g_{\beta,\beta}(\theta) = \frac{\beta+1}{\beta^4}\Psi_1\left(\frac{\beta+1}{\beta}\right) + \frac{1}{\beta^2} + \frac{1}{\beta^3}\left[\left(\Psi\left(\frac{1}{\beta}\right)\right)^2 + 2(\beta+1)\Psi\left(\frac{1}{\beta}\right)\right]$

Tableau IV.3 – Matrice d'information de Fisher pour une vraisemblance GGD

séparable, l'indépendance de τ et de $\vec{g}$ conduit à écrire :

$$\mathrm{KL}(\theta \parallel \theta') = \mathrm{KL}_\phi(M \parallel M') + \mathrm{KL}_\tau(a, b \parallel a', b') \tag{IV.15}$$

Remarque : l'équation (IV.15) reprend l'équation (III.19) avec

$$\begin{aligned} \mathrm{KL}_\tau(a, b \parallel a', b') &= \mathrm{KLD}(p_\tau(\tau|a, b) \parallel p_\tau(\tau|a', b')) \\ \mathrm{KL}_\phi(M \parallel M') &= \mathrm{KLD}(\phi(\vec{g}|M) \parallel \phi(\vec{g}|M')) \end{aligned}$$

L'estimation du barycentre est conduite de manière séparée du fait de la définition séparée de la divergence. Elle se constitue d'un barycentre d'une distribution gaussienne multivariée (qui est une famille exponentielle) et d'un barycentre de Weibull (pour la partie multiplieur). L'estimation du barycentre pour un modèle de dépendances spatiale exploite conjointement l'état de l'art pour les familles exponentielles ainsi que les barycentres pour des familles non exponentielles.

La suite de cette section est découpée en deux parties. Dans une première partie nous présentons une modification apportée aux paramètres afin d'assurer leur indépendance dans y. Dans une seconde partie, l'indépendance est utilisée conjointement à la séparabilité de la divergence de Jeffrey pour estimer le barycentre $\bar{\theta}$.

3.5.1 Indépendance

Théoriquement, l'indépendance informationnelle entre la distribution gaussienne multivariée $\vec{g}$ et le multiplieur τ est obtenue en fixant $\mathbb{E}[\tau] = 1$ ce qui permet aussi de réduire le nombre de paramètres utilisés pour la modélisation du multiplieur. Alternativement, la littérature propose de forcer la trace de la matrice de covariance de $\vec{g}$ à valoir 1 pour atteindre cette même indépendance informationnelle entre la distribution gaussienne multivariée $\vec{g}$ et le multiplieur τ.

Le multiplieur τ est une variable aléatoire modélisée par une loi Weibull (voir équation (III.14)) avec a et b les paramètres, respectivement, de forme et d'échelle. De plus, comme le modèle SIRV dispose d'une distribution gaussienne multivariée $\vec{g}$ modélisée par une matrice de covariance M, le multiplieur τ est normalisé afin d'obtenir une moyenne de 1. Autrement dit $\mathbb{E}\{\tau\} = 1$, ce qui donne $b = \left(\Gamma\left(1/a + 1\right)\right)^{-1}$. Pour la suite de cette section, la loi Weibull est uniquement caractérisée par le paramètre de forme $\theta_\tau = a$.

3.5.2 Séparabilité

Soit $\vec{y}$ la distribution jointe $(\tau, \vec{g})$, alors l'estimation du barycentre $\bar{\theta}$ au sens de la divergence de Jeffrey devient, par séparabilité de J, une estimation indépendante de deux barycentres $\bar{\theta} = (\bar{a}, \bar{M})$:

$$J(\theta, \theta') = J_\phi(M, M') + J_\tau(a, a') \tag{IV.16}$$

La séparabilité de J donne que la fonction de coût l est séparable. Le tableau IV.4 présente la formulation de la fonction de coût l comme somme d'une fonction de coût l_ϕ sur la partie gaussienne et d'une fonction de coût l_τ sur le multiplieur. Le minimum de la fonction de coût l_ϕ est estimé par l'approche famille exponentielle alors que la fonction de coût l_τ nécessite une méthode de recherche linéaire.

Fonction de coût	$l(\bar{\theta}) = l_\phi(\bar{M}) + l_\tau(\bar{a})$
Sur $\vec{g}$	$l_\phi(\bar{M}) = \sum_{k=1}^{K} J_\phi(M_k, \bar{M}))$
Sur τ	$l_\tau(\bar{a}) = \sum_{k=1}^{K} J_\tau(a_k, \bar{a})$
Densité τ	$p_\tau(\tau \mid a) = a\tau^{a-1} \left(\Gamma\left(1/a+1\right)\right)^a e^{-(\tau(\Gamma(1/a+1)))^a}$

Tableau IV.4 – Fonction de coût pour un modèle SIRV avec une distribution Weibull pour multiplieur

Barycentre $\vec{g}$. La distribution gaussienne multivariée est issue d'une famille exponentielle. L'estimation du barycentre $\bar{M}$ est, comme expliqué dans la sous-section 2.2 le résultat d'une dichotomie entre les barycentres orientés à gauche $\bar{M}_L$ et à droite $\bar{M}_R$. Nielsen [107] fourni l'ensemble des définitions nécessaires pour leur estimation. Le changement de coordonnées entre les paramètres sources M et les paramètres naturels θ_ϕ :

$$\phi(M) = \frac{1}{2} M^{-1}$$

et sa fonction réciproque :

$$\phi^{-1}(\lambda) = \frac{1}{2} \lambda^{-1}.$$

La fonction log-normalisante A a pour dérivée la changement de coordonnées entre les paramètres naturels λ et les paramètres espérés H :

$$\nabla A(\lambda) = -\frac{1}{2} \lambda^{-1}$$

et sa fonction réciproque :

$$(\nabla A)^{-1}(H) = -\frac{1}{2} H^{-1}.$$

Les barycentres orientés sont donnés explicitement par les formules (IV.6) et (IV.7) :

$$\bar{M}^L = -\frac{1}{dK}\sum_{k=1}^{K} M_k, \qquad \bar{M}^R = -dK\left(\sum_{k=1}^{K} M_k^{-1}\right)^{-1}$$

Le barycentre $\bar{M}$ est le résultat de la dichotomie entre les barycentre orientés. L'algorithme IV.1 est une implémentation simple pour obtenir le barycentre.

Barycentre τ. La distribution Weibull est, tel que la GGD, une famille non exponentielle. Cette sous-section présente les formules explicites et la méthode de recherche utilisée pour le barycentre. Supposons que $b = (\Gamma(1/a+1))^{-1}$, alors la fonction de coût l_τ s'écrit :

$$\begin{aligned} l_\tau(a) = 2K + \sum_{k=1}^{K} & -\frac{a_k}{a}\left(\frac{\Gamma(1/a_k+1)}{\Gamma(1/a+1)}\right)^{a_k}\Gamma\left(\frac{a_k}{a}\right) - \frac{a}{a_k}\left(\frac{\Gamma(1/a+1)}{\Gamma(1/a_k+1)}\right)^{a}\Gamma\left(\frac{a}{a_k}\right) - \\ & \frac{(a-a_k)^2}{aa_k}\Psi(1) + a\log\left\{\frac{\Gamma(1/a+1)}{\Gamma(1/a_k+1)}\right\} + a_k\log\left\{\frac{\Gamma(1/a_k+1)}{\Gamma(1/a+1)}\right\} \end{aligned} \tag{IV.17}$$

Cette fonction de coût l_τ n'a qu'une seule variable, cela implique que la dérivée partielle en a est égale à la dérivée de l_τ :

$$g_{5,k}(a) = \left(\frac{\Gamma(1/a_k+1)}{\Gamma(1/a+1)}\right)^{a_k}\Gamma\left(\frac{a_k}{a}+1\right) \tag{IV.18}$$

$$g_{6,k}(a) = \left(\frac{\Gamma(1/a+1)}{\Gamma(1/ap+1)}\right)^{a}\Gamma\left(\frac{a}{a_k}+1\right) \tag{IV.19}$$

$$\begin{aligned} l'_\tau(a) = & \sum_{k=1}^{K}\left(\frac{a_k}{a^2}-\frac{1}{a_k}\right)\Psi(1) - \left[\frac{a_k}{a^2}(1-g_{5,k}(a)) - \frac{1-g_{6,k}(a)}{a}\right]\Psi\left(\frac{1}{a}+1\right) - \\ & g_{5,k}(a)\frac{a_k}{a^2}\Psi\left(\frac{a_k}{a}+1\right) + g_{6,k}(a)\frac{1}{a_k}\Psi\left(\frac{a}{a_k}+1\right) + \\ & (1-g_{6,k}(a))\left(\log\left\{\Gamma\left(\frac{1}{a_k}+1\right)\right\} - \log\left\{\Gamma\left(\frac{1}{a}+1\right)\right\}\right) \end{aligned} \tag{IV.20}$$

La méthode ALGOL, de Dekker [110], permet d'obtenir rapidement la valeur de a pour laquelle la dérivée l'_τ de la fonction de coût s'annule. Brent [111] propose une amélioration de l'implémentation en terme de complexité calculatoire plus faible, nommé ALGOL 60. Cet algorithme est implémenté dans le logiciel Matlab® sous la fonction `fzero`.

Nous avons donc utilisé le modèle SIRV avec multiplieur Weibull pour la classification d'images texturée avec une supposition de dépendance spatiale limitée [64, 75].

Au cours de cette section nous avons présenté un schéma complet pour estimer le barycentre $\bar{\theta}$ pour la GGD. Du point de vue applicatif, il suffit de connaitre le gradient ∇l de la fonction de coût et la matrice d'information de Fisher G du modèle pour implémenter une méthode d'adaptation du gradient naturel. Cette méthode est comparée à deux autres méthodes de recherche linéaire afin de montrer le gain obtenu en vitesse de convergence. Nous avons ensuite présenté la vitesse de descente q_j comme un paramètre supplémentaire permettant de gagner en vitesse de convergence tout en respectant les contraintes c.

Nous avons également présenté un schéma pour estimer le barycentre pour un modèle SIRV avec multiplieur Weibull. La divergence J séparée induit une fonction de coût séparée et, par indépendance des fonctions de coût, à une estimation séparée du barycentre $\bar{\theta}$ avec, d'une part, l'estimation de la matrice de covariance $\bar{\Sigma}$ et, d'autre part, le paramètre du multiplieur $\bar{a}$. Un point commun à tous les modèles stochastiques est le paramètre d'échelle, le formalisme joint propose d'utiliser deux densités de probabilité et donc deux paramètres d'échelle sont estimés. La précision de l'estimateur d'un paramètre d'échelle dépend de la précision de l'estimateur de l'autre paramètre d'échelle, ce qui impacte la précision de tous les estimateurs. Le paramètre d'échelle est transféré du multiplieur en supposant que la moyenne de ce multiplieur égale 1, au profit de la matrice de covariance $\bar{\Sigma}$.

La section suivante présente une étude de l'existence et l'unicité du barycentre dans le cadre univarié. En effet, le modèle pour les images texturées peut être validé autrement qu'avec les résultats de classification. Résultats de classification qui sont présentés dans le prochain chapitre de ce mémoire.

4 Existence et unicité du barycentre

Les sections précédentes introduisent des méthodes d'estimation du barycentre $\bar{\theta}$ comme un minimum de la fonction de coût l. Les méthodes de recherche linéaire utilisées définies sont itératives et nécessitent l'existence d'une zone de la variété où les itérations θ_{j+1} et θ_j sont suffisamment similaires (critère d'arrêt des algorithmes). La condition d'arrêt des algorithmes est une condition suffisante pour trouver un extremum dans cette zone de la variété. Par simplicité, les méthodes de recherche nécessitent la présence d'un extremum local. Ensuite il est nécessaire de vérifier que cet extremum local soit un minimum local. Deux conditions sont suffisantes pour avoir un minimum, la première est que tout vecteur θ dans un voisinage de l'extremum local vérifie $l(\theta) \leqslant l(\bar{\theta})$; la seconde étant que la dérivée seconde de la fonction de coût l s'annule en l'extremum. Les deux conditions sont difficiles à vérifier, la première demande une recherche plus qu'exhaustive avec une fonction continue, la seconde demande que la fonction de coût l soient au moins deux fois dérivable. Généralement, une autre condition suffisante est utilisée, l'aspect

convexe de la fonction de coût l. L'aspect convexe de l nous donne l'existence du minimum local.

Supposons que les méthodes de recherche linéaire trouvent un minimum local. Une question se pose alors sur la véracité du vecteur calculé, savoir si le minimum calculé est le minimum global c'est supposer qu'il existe un autre minimum local. La littérature parle alors d'unicité du minimum local, qui devient alors le minimum global. Une condition suffisante pour avoir l'unicité du minimum est que l soit une fonction strictement convexe. A titre de contre exemple, soit l une fonction de coût à valeurs constantes. La fonction l est convexe. Néanmoins, tous les vecteurs sont des minima locaux.

Considérons maintenant la fonction de coût l construite avec le modèle univarié GGD. La section courante montre que la fonction de coût n'est pas convexe et que la recherche d'existence et d'unicité en est plus complexe. Le manque d'outils mathématiques ne permet ni de valider ni d'invalider le caractère unique du barycentre $\bar{\theta}$. Car la divergence de Jeffrey ne vérifie pas l'inégalité triangulaire, un minimum local peux ce situer au delà d'un espace convexe autour des vecteurs paramétriques θ_k, pour $k = 1, \ldots, K$.

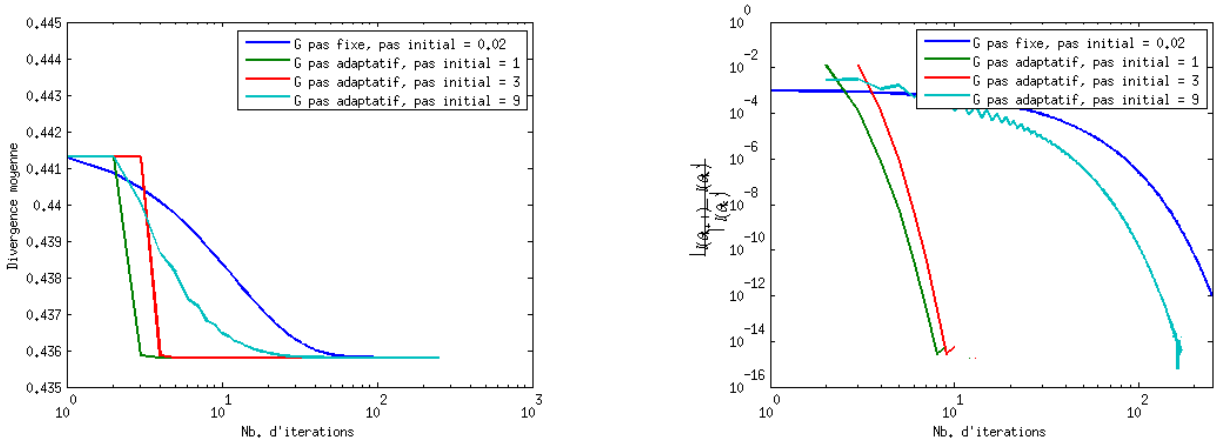
4.1 Existence du barycentre

Nous cherchons à trouver le minimum de la fonction de coût l. Le vecteur de paramètres minimisant cette fonction de coût n'est autre que l'estimateur au sens du maximum de vraisemblance du barycentre $\bar{\theta}$, alors que la valeur atteinte au minimum de cette fonction est l'estimateur au sens du maximum de vraisemblance de la variance σ^2. Nous supposons ici que la vraisemblance $p(x \mid \theta)$ suit une GGD (III.3).

En 2007, Choy et Tong [44] se sont intéressés à l'estimation du barycentre pour une loi *a priori* intrinsèque construite autour de la divergence de Kullback-Leibler à droite. En annexe de [44] ils montreront l'existence du barycentre sans pousser jusqu'à l'unicité d'un tel barycentre. La KL n'est pas symétrique, et la démonstration de Choy et Tong n'est pas valide pour montrer l'existence du barycentre pour une loi *a priori* intrinsèque construite autour de la divergence de Kullback-Leibler à gauche.

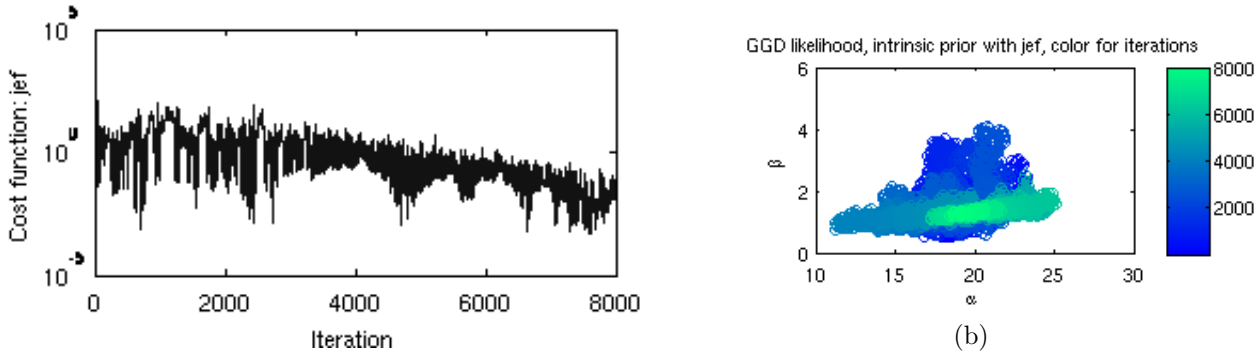
La divergence de Jeffrey est une symétrisation de la KL. Et sa définition (III.24) montre que l'existence d'un minimum pour la fonction de coût dépend fortement des preuves d'existence pour les KL orientées. Nous n'avons pas de démonstration théorique de l'existence de ce barycentre.

L'existence du barycentre $\bar{\theta}$ au sein d'un espace convexe au sens géodésique est étudié. Soit V l'espace convexe au sens géodésique. Pour rappel la géodésique est la courbe de plus courte longueur entre deux vecteurs paramétriques. Définir V comme convexe, c'est affirmer que pour tous couples de vecteurs paramétriques, la géodésique qui les relie est contenue dans V. La démonstration est effectuée par récurrence.



(a) (b)

Figure IV.4 – Étude comparative entre 3 exemples d'utilisation d'un pas adaptatif et un exemple d'utilisation d'un pas non adaptatif. Initialisation avec la moyenne arithmétique empirique $\theta_0 = \frac{1}{K}\sum_{k=1}^{K}\theta_k$. Le code couleur choisi est bleu pour l'utilisation d'un pas de mise a jour fixé à $\epsilon_0 = 0,02$. Les codes couleur vert, rouge et cyan sont utilisés avec un pas adaptatif. Ce pas adaptatif est initialisé à 1, 2 et 9 respectivement pour chaque couleur. La figure (a) montre la fonction de coût suivant le nombre d'itérations j de l'algorithme, tandis que la figure (b) montre la dérivée de la fonction de coût suivant le nombre d'itérations j.



(a) (b)

Figure IV.5 – Soit un jeu de vecteurs de paramètres $(\theta_n)_{n=1}^{N}$. La suite $(\theta_k)_{k=0}$ mise à jour avec un recuit simulé évolue dans Θ vers le barycentre global. Pour chaque mise à jour nous pouvons positionner θ_k avec une couleur basée sur k (b) et nous pouvons calculer la valeur de la fonction de coût l (a)

Soit le cas initial avec $N = 2$. L'application numérique montre alors que la fonction de coût est égale pour les deux vecteurs paramétriques $l(\theta_1) = l(\theta_2)$. Soit les vecteurs sont confondus et V_2 n'est constitué que d'une valeur qui fait office de minimum et de maximum pour la fonction de coût. Soit les vecteurs ne sont pas confondus, permettant d'appliquer le théorème des valeurs intermédiaires. Il existe un extremum r sur la géodésique reliant les deux vecteurs paramétriques θ_1 à θ_2. Deux cas simples se présentent, soit r est un minimum de la fonction de coût l alors $\bar{\theta}_2 = r$, soit r est un maximum alors le minimum est atteint en θ_1 et θ_2. L'espace convexe V_2 contient les trois vecteurs θ_1, θ_2 et r. V_2 contient au moins un minimum de la fonction de coût l.

Pour vérifier la récurrence, V_{k-1} contient un minimum $\bar{\theta}_{k-1}$ pour la fonction de coût

$$l_{k-1}(\theta) = \sum_{n=1}^{k-1} w_{k-1,n} J(\theta, \theta_n), \qquad 1 = \sum_{n=1}^{k-1} w_{k-1,n}$$

La fonction de coût l vérifie l'égalité récursive

$$l(\theta) = (1 - w_k) l_{k-1}(\theta) + w_k J(\theta, \theta_k)$$

Ce qui permet d'évaluer la fonction de coût en $\bar{\theta}_{k-1}$

$$l(\bar{\theta}_{k-1}) = (1 - w_k) l_{k-1}(\bar{\theta}_{k-1}) + w_k J(\bar{\theta}_{k-1}, \theta_k) = (1 - w_k)\sigma_{k-1}^2 + w_k J(\bar{\theta}_{k-1}, \theta_k)$$

avec σ_{k-1}^2 le minimum obtenu par la fonction de coût l_{k-1} et w_k une variable utilisée pour répartir les poids. Considérons maintenant l'opposé : évaluer la fonction de coût l au niveau du vecteur de paramètres θ_k.

$$l(\theta_k) = \sum_{n=1}^{k} w_n J(\theta_k, \theta_n) = \sum_{n=1}^{k-1} w_n J(\theta_k, \theta_n) = (1 - w_k) l_{k-1}(\theta_k)$$

Supposons d'abord que θ_k est un minimum de la fonction de coût l_{k-1}, alors $l(\theta_k) = l(\bar{\theta}_{k-1}) = (1 - w_k)\sigma_{k-1}^2$. Comme pour le cas $N = 2$ il existe un minimum $\bar{\theta}$ sur la géodésique reliant θ_k à $\bar{\theta}_{k-1}$. Il existe un minimum sur V pour la fonction de coût l. Autrement la définition du minimum $\bar{\theta}_{k-1}$ indique que

$l_{k-1}(\theta_k) > \sigma^2_{k-1}$. Par extension, nous écrivons

$$\begin{aligned} & (1-w_k)l_{k-1}(\theta_k) > (1-w_k)\sigma^2_{k-1} \\ \Leftrightarrow \quad & l(\theta_k) + w_k J(\bar{\theta}_{k-1}, \theta_k) > l(\bar{\theta}_{k-1}) \\ \Leftrightarrow \quad & w_k J(\bar{\theta}_{k-1}, \theta_k) > l(\bar{\theta}_{k-1}) - l(\theta_k) \end{aligned}$$

Cela montre que la différence des valeurs prises par la fonction est majorée. La fonction de coût l est de ce fait bornée sur la géodésique reliant θ_k à $\bar{\theta}_{k-1}$ en plus d'y être continue. Le théorème des valeurs intermédiaires nous donne l'existence d'un extremum r sur la géodésique. Deux cas de figures se présentent, le premier étant que $r = \bar{\theta}$ est le minimum de la fonction de coût. Autrement r est un maximum ou un point selle, le minimum peux alors se trouver sur une des extrémités de la géodésique, à savoir les vecteurs paramétriques $\bar{\theta}_{k-1}$ ou θ_k. Encore une fois, le minimum est élément de la géodésique reliant le précédent minimum $\bar{\theta}_{k-1}$ au nouveau vecteur paramétrique θ_k. Il existe donc un minimum $\bar{\theta}$ dans l'espace convexe V_k autour des vecteurs paramétriques.

Nous avons montré, par récurrence, qu'il existe un minimum $\bar{\theta}$ à la fonction de coût l sur un espace convexe au sens géodésique autour des vecteurs paramétriques $(\theta_k)_{k=1}^N$. La fonction de coût l n'est pas bornée sur l'ensemble de son espace de définition $\mathbb{R}^+ \times \mathbb{R}^+$, autrement dit il n'est pas possible d'étendre l'existence d'un minimum à toute la variété.

4.2 Le recuit simulé

La méthode du recuit simulé est une méthode numérique « gloutonne » qui est développée afin de trouver le maximum/minimum global d'une fonction de coût. Pour cette méthode d'optimisation globale, nous utiliserons encore une suite de vecteurs de paramètres $(\theta_k)_{k=0}^{\infty}$. La suite $(\theta_k)_{k=0}^{\infty}$ est construite pour converger vers le minimum global. La mise à jour est aléatoire autour du vecteur paramétrique θ_k qui le précède, néanmoins elle n'est acceptée que dans deux conditions soit 1) la fonction de coût décroit ; soit 2) la variable u uniforme sur $[0;1]$ est inférieure à la probabilité :

$$p(\theta_{k+1} \mid \theta_k, T_k) = \exp\left\{-\frac{1}{T_k}|l(\theta_{k+1}) - l(\theta_k)|\right\}$$

avec une baisse de température de b, $T_{k+1} = b * T_k$.

Nous pouvons jouer sur deux critères distincts pour la convergence de l'algorithme : le pas d'avancement de la suite θ_k et la force de la baisse de température b. Mal paramétré, le recuit simulé ne se déplace

pas suffisamment, remettant en cause l'aspect global du minimum estimé. La suite peux également diverger si elle est mal paramétrée et tomber sur un minimum local trop lointain du minimum global. Le critère d'arrêt choisi est basé sur le nombre d'itérations de l'algorithme. Suite au déroulement de la méthode de recuit simulé, nous proposons d'utiliser une méthode de descente de gradient pour s'approcher au mieux du barycentre global.

Les figures IV.5.(a) et IV.5.(b) présentent l'évolution d'une suite $(\theta_k)_{k=0}^{\infty}$ mise à jour par un algorithme de recuit simulé. Nous représentons la position des θ_k dans l'espace des paramètres Θ afin de montrer que l'algorithme cherche largement avant de se concentrer lorsque la température baisse. La couleur de la figure IV.5.(b) correspond à l'index k de la suite, nous voyons qu'en laissant le temps à l'algorithme ce dernier se dirige naturellement vers le nuage $(\theta_n)_{n=1}^{N}$ position probable du barycentre. La figure IV.5.(a) montre le bon déroulement de l'algorithme avec une décroissance de la valeur maximale prise par la fonction de coût l. Il est donc possible de trouver un vecteur de paramètres $\bar{\theta}$ qui possède une valeur $l(\bar{\theta})$ inférieure à d'autres $l(\theta)$. Nous avons donc l'existence numérique du barycentre.

En plus de montrer l'existence du barycentre, le recuit simulé est une méthode suffisamment efficace pour trouver le minimum global dans le cas ou la fonction de coût possède plusieurs minima locaux. Mais la recherche aléatoire du barycentre est plus complexe d'un point de vue calculatoire que de réaliser une simple descente de gradient. D'un point de vue pratique nous nous orientons vers la descente de gradient et nous souhaitons comparer les barycentres obtenus.

La figure IV.6 présente une comparaison entre le barycentre local (estimé avec une descente de gradient) et le barycentre global (estimé avec le recuit simulé). Nous utilisons le même jeu de points $(\theta_n)_{n=1}^{N}$ que dans les figures IV.5.(a) et IV.5.(b), néanmoins nous les représentons dans la figure IV.6 par des cercles verts. Les deux algorithmes mettent à jour une suite $(\theta_k)_{k=0}^{\infty}$ d'éléments de Θ et ce que nous pouvons fixer est l'initialisation des ces suites, θ_0 est représenté dans la figure IV.6 par un carré vert. Enfin nous comparons le barycentre global (plus rouge) au barycentre local (croix bleue). Nous apprécions l'égalité des deux barycentres, indiquant que les méthodes fonctionnent de manière similaire pour cette initialisation. De plus il n'y a pas de minimum local (croix bleue) entre le θ_0 (carré vert) et le barycentre global (plus rouge). Numériquement nous avons que le barycentre est unique par l'absence de minimum local autre que le barycentre.

Pour qu'une application numérique valide le caractère global du barycentre estimé avec notre méthode de descente de gradient, elle se doit d'être exhaustive. Autrement dit, lancer une étude exhaustive sur des cas de figure réels avec comparaison entre les méthodes du recuit simulé et de descente de gradient. Par

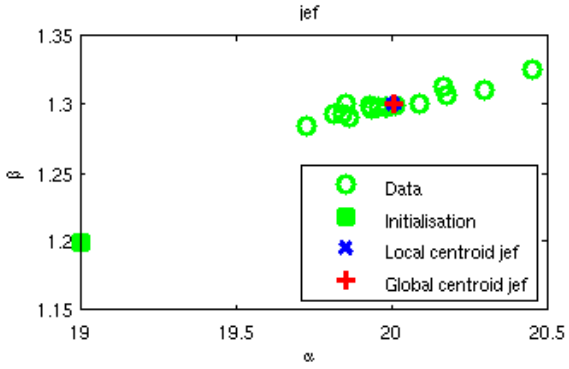
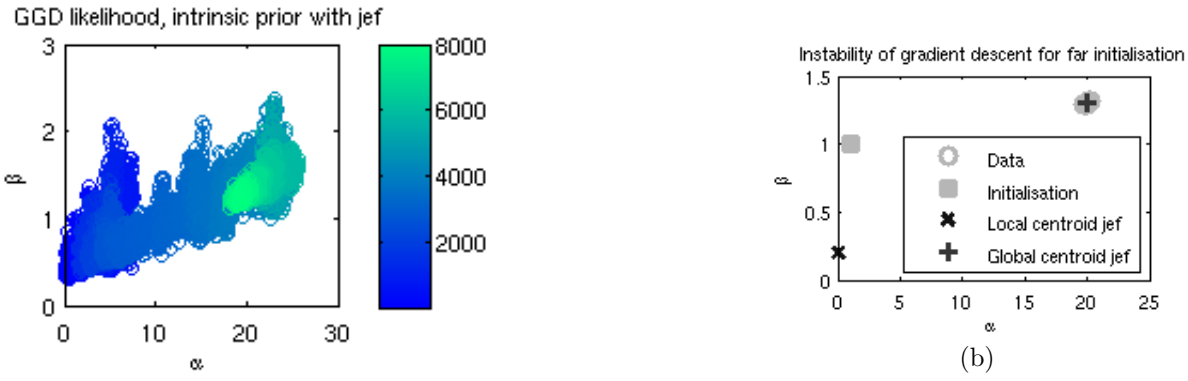


Figure IV.6 – Soit un jeu de vecteur de paramètre $(\theta_n)_{n=1}^N$ positionné par des cercles verts. Nous initialisons les méthodes de recuit simulé et de descente de gradient au niveau du même carré vert. La descente de gradient renvoie la position indiquée par la croix bleue alors que le recuit simulé renvoie la position indiquée par le plus rouge.



(a) (b)

Figure IV.7 – Soit un jeu de vecteurs de paramètre $(\theta_n)_{n=1}^N$. La suite $(\theta_k)_{k=0}$ initialisée au carré vert et mise à jour avec une méthode du recuit simulé évolue dans Θ vers le barycentre global (plus rouge). Le recuit simulé se rapproche du nuage $(\theta_n)_{n=1}^N$ (a) et le barycentre obtenu (plus rouge) se trouve dans le nuage, mais la méthode de descente de gradient diverge (croix bleue) (b)

gain de temps, il serait même recommandé de n'utiliser que la méthode de recuit simulé pour s'assurer de la pertinence du barycentre.

Considérons un nouvel exemple, la figure IV.7.(b) montre que si l'initialisation θ_0 est trop loin du nuage $(\theta_n)_{n=1}^N$ alors la méthode de descente de gradient renvoie une position limite. La méthode du recuit simulé est allé chercher le barycentre global au sein d'une enveloppe convexe autour des vecteurs paramétriques $(\theta_n)_{n=1}^N$. Et la méthode par descente de gradient a divergé, suite à un soucis au niveau de son initialisation. Autrement dit, la descente de gradient devrait être utilisée pour sa vitesse de convergence du moment que l'initialisation se trouve dans l'enveloppe convexe des vecteurs paramétriques $(\theta_n)_{n=1}^N$.

Cette section n'a pas présenté l'existence et l'unicité du barycentre $\bar{\theta}$. L'existence étant valable localement tout comme la pertinence de la méthode par descente de gradient. En annexe de ce mémoire, le problème de l'unicité du barycentre $\bar{\theta}$ est poursuivie sur différentes pistes. Par exemple, des extrema sont présents aux limites de l'espace $\mathbb{R}^+ \times \mathbb{R}^+$, néanmoins ce ne sont pas des minima. Le barycentre est peut être unique, en tous les cas il est utile pour classifier des images texturées.

5 Conclusion

La méthode de « Sac de Mots Visuels » est découpée en trois étapes. L'étape d'apprentissage, première étape, ne peut être complétée qu'avec l'estimation des hyper-paramètres de position $\hat{\bar{\theta}}$ et de variance $\hat{\sigma^2}$. Le problème est ramené à la seule estimation de la position $\hat{\bar{\theta}}$ étant donné que la variance $\hat{\sigma^2}$ est calculée avec la position. Étant le seul à être estimé, l'hyper-paramètre $\hat{\bar{\theta}}$ de position est nommé « barycentre » pour le reste de ce chapitre. L'estimation du barycentre est simple lorsque le modèle paramétrique suit une famille exponentielle, la littérature fournit l'ensemble des outils pour en faire l'estimation.

Pour classifier les images texturées, ce mémoire utilise des modèles paramétriques (GGD, GΓD et modèle SIRV) qui ne font pas partie de la famille exponentielle. Le problème d'estimation du barycentre est alors écrit sous la forme d'un problème d'optimisation : minimiser la fonction de coût l. La littérature donne alors des méthodes de recherche linéaire pour trouver le minimum d'une fonction de coût l : la méthode de descente de gradient, la méthode de Newton-Raphson et la méthode d'adaptation du gradient naturel. Ses méthodes sont similaires dans le sens ou elles utilisent le gradient ∇l de la fonction de coût l comme direction de descente p_j. Nous proposons d'améliorer ses méthodes pour respecter les contraintes c existant sur les vecteurs paramétriques. Une étude comparative est menée sur plusieurs implémentations pour montrer l'apport de l'adaptation du gradient naturel en terme de vitesse de convergence par rapport

aux deux autres méthodes de recherche linéaire. Le modèle SIRV utilisé comme modèle paramétrique apporte plus de complications car la divergence de Jeffrey n'est qu'une approximation de la véritable distance. Nous proposons alors un barycentre défini comme une collection d'un barycentre $\bar{M}$ d'une distribution gaussienne multivariée et d'un barycentre $\bar{a}$ de distribution Weibull de variance $\sigma_\tau^2 = 1$.

Enfin, nous présentons une étude sur la fonction de coût l pour le modèle GGD. Cette étude présente que le barycentre $\bar{\theta}$ existe et qu'il est, visiblement, le seul minimum de la fonction de coût l. Les outils mathématiques mis à disposition n'infirment pas l'étude visuelle conduite mais ne permettent pas de la valider. La dimension de la variété paramétrique pour les modèles GΓD et SIRV ne permet pas encore une étude visuelle et les outils mathématiques sont actuellement insuffisants. La recherche future devrait alors ce concentrer sur la validation théorique de l'existence et de l'unicité du barycentre sur une variété. Le prochain chapitre présente l'aspect applicatif de ce travail avec l'évaluation de performances de classification sur des bases de données d'images texturées.

Chapitre V

Application : Classification des images texturées

Contenu du chapitre

1 Introduction

Lors des trois chapitres précédents, nous avons présenté notre méthode pour la classification d'images texturées. Ce chapitre regroupe les expérimentations que nous avons menées afin de confronter notre solution à la vérité terrain. Commençons par un récapitulatif des différentes applications de vision par ordinateur qui utilisent l'information texture [112–115], pour l'imagerie satellitaire [7, 116–118], l'imagerie médicale [57, 119–121], l'étude de minéraux [122–125] ou l'étude de la qualité de pièces manufacturées [126–128]. Ces différents travaux constituent l'état de l'art et toutes ces solutions sont des cas particuliers de méthodes de classification de primitives.

La classification de primitives est l'étude de comment les ordinateurs perçoivent l'environnement [129], apprennent à faire la différence entre les primitives d'intérêt et le fond, en catégorisant les informations. Une des nombreuses primitives présentes dans l'image est la texture [7, 130–135]. C'est sur cette dernière que nous allons nous focaliser et nous intéresser à évaluer les performances de classification des méthodes basées sur des descripteurs paramétriques. Par exemple, la recherche d'image par contenu texturé (CBIR) utilise les modèles paramétriques depuis près de dix ans [4, 48]. Le domaine d'application du CBIR est la recherche d'images similaires à une image requête au sein de bases de données très grandes.

Certaines solutions ont été développées sur des bases de données anciennes ou sur des bases de données qui ne sont pas accessible publiquement. Une comparaison exhaustive des méthodes passe alors par l'implémentation des solutions de la littérature. Dans les bases de données actuelles accessibles la diversité naturelle est beaucoup plus présente que celles utilisées par la littérature plus ancienne. Les solutions de l'état de l'art qui ne prennent pas en compte la diversité naturelle ne peuvent pas être des concurrentes de la solution que nous proposons et qui exploite les « Sac de Mots Visuels » (SMV) avec des descripteurs paramétriques. Par contre, les méthodes plus récentes comme celle de Varma et Zisserman [10] sont des méthodes de classification invariantes à la diversité.

Concernant les aspects expérimentaux, ce chapitre est composé de trois parties. La première partie présente le problème de la classification des images texturées. Il est également l'occasion de valider les modèles paramétriques présentés dans le chapitre 1. Le deuxième partie présente une classification basée barycentre, en comparaison avec les travaux de Choy et Tong [44]. Une classe d'image texturée est représentée par la distribution GGD caractéristique des coefficients d'ondelette. Cette distribution GGD caractéristique est juste un barycentre au sens d'une divergence. Dans notre travail nous proposons

de : remplacer la distribution GGD par une distribution GΓD ou un modèle SIRV ; utiliser la variance conjointement au barycentre ; utiliser plusieurs barycentres par classe. La troisième partie présente le SMV avec « descripteurs paramétriques » que nous avons implémentés et une comparaison avec un SMV avec « descripteur patchs » donné par Varma et Zisserman [10].

2 Position du problème et validation du modèle paramétrique

Dans ce qui suit, l'image texturée appartient à une classe. La classe est un concept qui peux être commun à un ensemble d'images. Par exemple, plusieurs images de papier. La classification consiste à réaliser une décision sur la classe d'une image test. La classe de cette image test peut être connue ou pas. Il est alors question de classification supervisée ou non supervisée. Dans nos expérimentations la classification supervisée est choisie afin d'évaluer les performances de classification et des méthodes étudiées.

La caractérisation des textons d'images texturées permet, aussi bien, de faire apprendre automatiquement à l'algorithme les classes que de tester ces dernières. Il existe des collections d'images texturées accessible publiquement, elles sont appelées bases de données d'images texturées.

Cette thèse repose, entre autres, sur deux choix. Le premier concerne le modèle paramétrique (issus des choix de dépendance intra-bande, dépendance inter-bandes et séparabilité de la mesure de dissimilarité) et le second le type de modélisation pour le mélange de gaussiennes concentrées. Pour le premier, il est question de valider expérimentalement les modèles univariés utilisés. Pour le second, il s'agit de valider une représentation des descripteurs paramétriques.

Cette section est découpée en 4 sous-sections. La première présente les bases de données utilisées dans ce document. La seconde sous-section présente la validation des modèles paramétriques. La troisième sous-section discute de la représentation des descripteurs paramétriques. La quatrième sous-section présente le protocole expérimental utilisé pour l'évaluation des performances ainsi que les résultats obtenus.

2.1 Bases de données d'images texturées

Une base de données d'images texturées est une collection d'images. Dans celles présentées ici, les images sont associées à des classes *a priori*. Cela permet d'automatiser l'association entre une image d'apprentissage et la classe correspondante, puis d'évaluer les performances de classification. Le tableau I.1 présenté dans l'introduction de ce document présentait une liste non exhaustive des bases de données

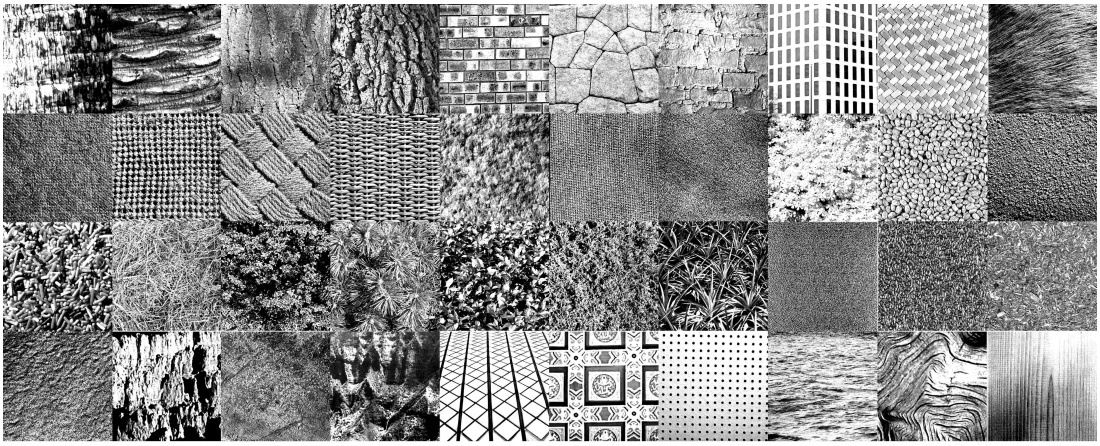


Figure V.1 – Ensemble de 40 images texturées de la base de données VisTex. Parcours lexicographique (de gauche à droite et de haut en bas) : 'Bark.0000', 'Bark.0006', 'Bark.0008', 'Bark.0009', 'Brick.0001', 'Brick.0004', 'Brick.0005', 'Buildings.0009', 'Fabric.0000', 'Fabric.0004', 'Fabric.0007', 'Fabric.0009', 'Fabric.0011', 'Fabric.0014', 'Fabric.0015', 'Fabric.0017', 'Fabric.0018', 'Flowers.0005', 'Food.0000', 'Food.0005', 'Food.0008', 'Grass.0001', 'Leaves.0008', 'Leaves.0010', 'Leaves.0011', 'Leaves.0012', 'Leaves.0016', 'Metal.0000', 'Metal.0002', 'Misc.0002', 'Sand.0000', 'Stone.0001', 'Stone.0004', 'Terrain.0010', 'Tile.0001', 'Tile.0004', 'Tile.0007', 'Water.0005','Wood.0001','Wood.0002'

utilisées usuellement dans la « littérature » et disponibles sur internet. Cette sous-section va présenter avec plus de détails quelque unes de ces bases de données.

2.1.1 Base de données de texture visuelles (VisTex)

La base de données de référence pour ce document sera la base de données VisTex [136]. La base de données VisTex est constituée de près de 167 images texturées en couleurs de taille 512x512 au format sans perte `ppm`. Un sous-ensemble de 40 classes de cette base de données est utilisé par différents auteurs [4, 45, 50–52, 54, 137] (détaillée sur la figure V.1).

Cette base de données ne présente qu'une faible variabilité en homogénéité et éclairage. Le gain en performances des approches invariantes par rotations ne présentent pas de gains significatifs [50].

2.1.2 Livre de Philip Brodatz (BrodatzR)

Dans son livre [138], Brodatz présente 112 images texturées mais seule 13 images de taille 512×512 pixels seront sélectionnées et numérisées par l'institut de traitement du signal et d'image de l'université de Californie du sud. La qualité des images numérisées est basse comme l'illustre l'image incomplète représentant la texture de cuir de porc (voir figure V.2). Néanmoins ils créent une base d'images texturées présentant des images tournées de 7 angles différents avant la numérisation. Cette base de données est utilisée dans quelques travaux sur l'invariance [32, 50].

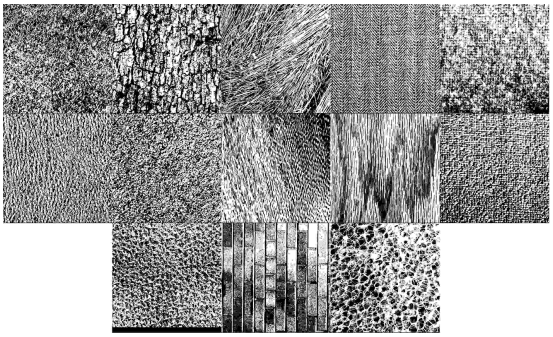


Figure V.2 – Ensemble de 13 images texturées de la base de données Brodatz proposée par sipi.usc.edu.

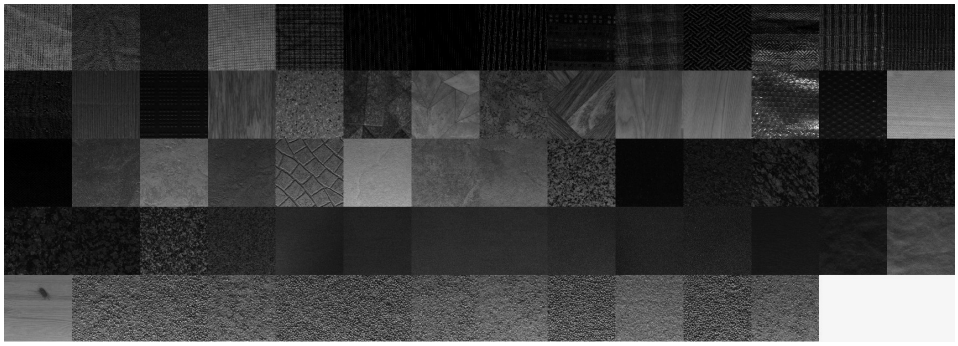


Figure V.3 – Ensemble de 68 images texturées de la base de données `Outex_TC_0013`.

La base de données Brodatz la plus complète est fournie par l'université de Stavanger (Norvège) même si une texture est manquante ramenant à 111 le nombre d'images texturées [12, 139]. Notez bien que la littérature [12, 26, 44, 46, 140–142] recèle de sélections personnelles d'images de la base de données Brodatz. La variabilité de ces bases de données repose sur les différentes qualités d'acquisition. Les différentes qualités sont spécifiques à cette base de données qui est issue d'un support imprimé.

2.1.3 Université de Oulu (OuTex)

OuTex est le nom d'une collection de bases de données de l'université de Oulu (Finlande) [141]. Différentes bases de données sont proposées avec des conditions d'éclairage et de prise de vue variables. Considérons la base de données nommée `Outex_TC_0013` constituée de 68 classes d'images texturées, chaque classe est composée de 10 images de dimensions 128×128 pixels. La figure V.3 présente les 68 textures, le nombre est suffisamment élevé pour avoir des performances comparables avec la base de données VisTex mais la diversité inter classes est faible.

2.1.4 Columbia/Utrech texture database (CUReT)

CUReT [143] désigne la base de données du Computer Vision Laboratory de l'université de Columbia. Elle contient 61 classes et est disponible en niveaux de gris ou en couleurs. Les images sont de dimensions 200×200 pixels. Le protocole d'imagerie prend en compte 7 positions de caméra autour de l'objet et un total de 205 illuminations différentes (sur les 7 positions de la caméra). La littérature [10–12] l'utilise comme base de données présentant un fort niveau de diversité intra-classe. C'est le candidat désigné pour tester les algorithmes SMV.

Dans ce mémoire, la qualité des bases de données est privilégiée par rapport à un test sur un grande quantité de bases de données différentes. La sous-section suivante montre la pertinence des modèles stochastiques utilisés.

2.2 Validation des modèles paramétriques

Le modèle stochastique est utilisé dans ce mémoire pour modéliser la distribution empirique des coefficients de décomposition (voir figure III.2, page 31). Cette sous-section montre séquentiellement les notations mathématiques pour le cas des modèles univariés puis pour celui des modèles multivariés.

Supposons que $(x_n)_{n=1}^N$ soit une collection de variables aléatoires issue de la sous-bande d'une image. Soient $(I_k)_{k=1}^K$ une partition sans recouvrement d'un fermé de $\mathbb{R}$ contenant la collection $(x_n)_{n=1}^N$. La distribution empirique la répartition de la collection des valeurs ou histogramme h_k :

$$h_k = \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}_{I_k}(x_n)$$

Chaque variable aléatoire est supposée suivre une distribution paramétrique $p(x \mid \hat{\theta})$. Soit $\hat{\theta}$ le vecteur paramétrique estimé par maximum de vraisemblance sur la collection $(x_n)_{n=1}^N$.

Afin d'évaluer la pertinence de la famille paramétrique considérée, nous comparons la distribution empirique h_n avec la distribution paramétrique $p(x \mid \hat{\theta})$. Un test d'adéquation de Kolmogorov-Smirnov :

$$\mathrm{KST}(H_k, P(x \mid \hat{\theta})) = \max_{n=1,\ldots,N} |H(x_n) - P(x_n \mid \hat{\theta})|.$$

est mis en place. Nous considérons H_k l'histogramme des fréquences cumulées :

$$\begin{aligned} H_k &= \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}_{\cap_{i=1}^{k} I_i}(x_n) = \sum_{i=1}^{k} h_i \\ H(x) &= \{H_k \mid x \in I_k\} \end{aligned}$$

et la fonction de répartition $P(x \mid \hat{\theta})$:

$$P(x \mid \hat{\theta}) = \int_{-\infty}^{x} p(t \mid \hat{\theta}).dt$$

Supposons de plus que $K = N$, alors I_k ne contient qu'une variable aléatoire x_n et l'index k correspond à l'ordre des variables x_n rangées par ordre croissant. Dans ce cas, $H(x_n) = k/N$.

La statistique du test de Kolmogorov-Smirnov est d'autant plus faible que la fonction de répartition et proche de la fréquence cumulée H_k.

2.2.1 Modèles stochastiques univariés

Considérons, tout d'abord, des distributions univariées. L'utilisation d'une distribution univariée dépend de l'indépendance inter-bandes et intra-bande. En plus des deux distributions univariées présentées dans le premier chapitre (la distribution gaussienne généralisée (GGD) et la distribution Gamma généralisée (GΓD)), la distribution Weibull [144] et la distribution Gamma [145] sont utilisées pour modéliser la distribution empirique.

La figure V.4 présente le résultat moyen du test de Kolmogorov-Smirnov calculé sur la base de données VisTex. La version de la base de donnée Vistex couramment utilisée dispose de 40 classes. Pour une classe, il existe 16 images et chaque image est décomposée en 6 sous-bandes. Le test de Kolmogorov-Smirnov est effectué sur une sous-bande pour les 4 modèles univariés choisis. Pour une classe et un modèle paramétrique, ce sont donc 96 valeurs qui sont moyennées avant d'être utilisées comme valeur des bins de la figure V.4. La couleur du bin - blanc, gris clair, gris foncé et noir - est associé à un modèle paramétrique - respectivement la distribution Gamma, la distribution Weibull, la GGD et la GΓD. Deux sont des modèles utilisés pour des applications impliquant la modélisation des images texturées : la GGD et la GΓD (cf. chapitre III). La distribution Weibull (cf. section 2.6) et la distribution Gamma font partie de la famille exponentielle d'une part et de la GΓD d'autre part (cf. section 2.2).

La GΓD montre une valeur de test moyen plus faible sur chacune des classes de la base de données par rapport aux trois autres modèles paramétriques. Les classes Bark 08 et Bark 10 (entre autres) montrent

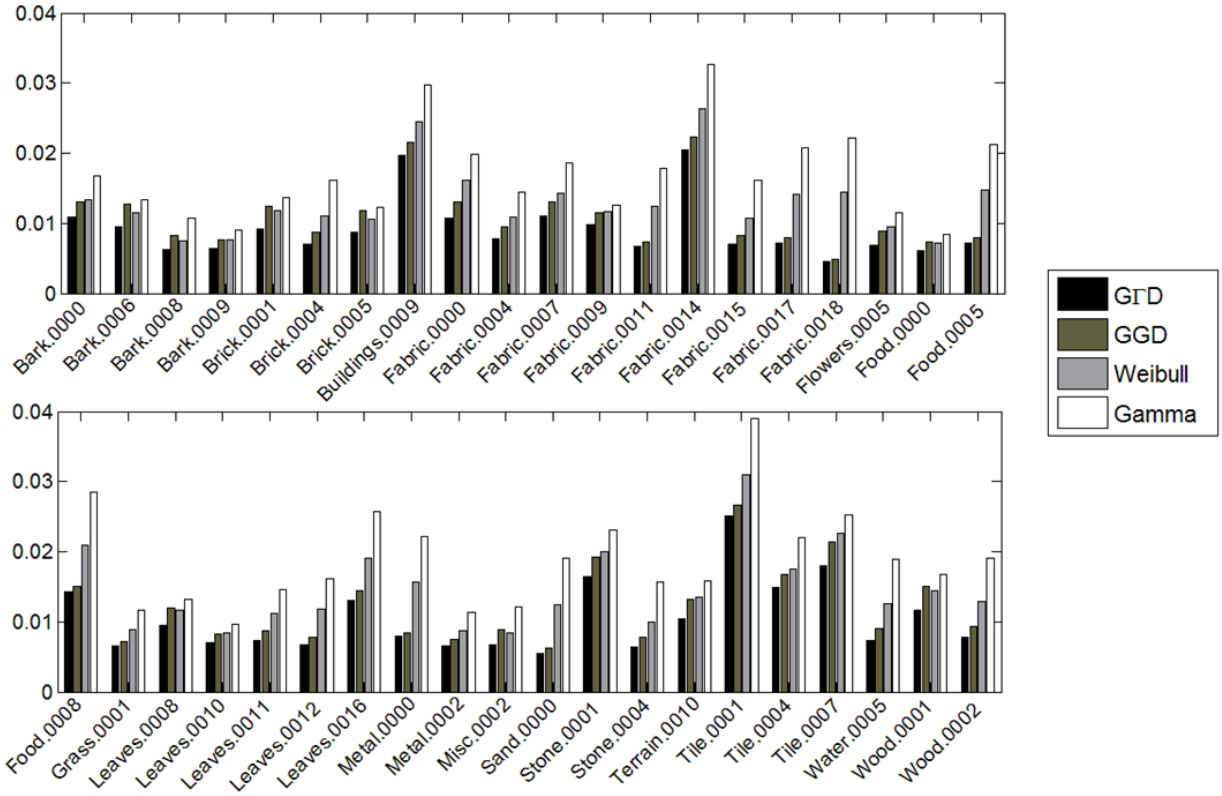


Figure V.4 – Pour chaque classe de la base de données VisTex, le résultat du test de Kolmogorov-Smirnov moyen sur les images pour 4 modèles paramétriques univariés. Les barres noires représentent le test effectué avec une distribution Gamma généralisée (respectivement gris foncé pour une distribution gaussienne généralisée, gris clair pour une distribution Weibull et blanches pour une distribution Gamma).

que la distribution Weibull à une valeur de test équivalent à la GGD, alors que les classes Fabric 14 et Fabric 17 montrent que la valeur du test pour la distribution Weibull est supérieure à la valeur du test pour la GGD. Les distributions GGD et GΓD s'adaptent plus facilement qu'une distribution Weibull ou une distribution Gamma. Les classes Food 00 et Leaves 10 montrent que les valeurs du test de Kolmogorov-Smirnov sont acceptables pour la distribution Gamma, et la famille exponentielle. De ce fait, une famille exponentielle peut être le modèle le plus adéquat pour un certain type de texture et pas pour un autre. Néanmoins, toutes les distributions montrent une dynamique similaire.

2.2.2 Modèles stochastiques multivariés

En supposant une dépendance inter-bandes, intra-bande ou couleur, le modèle paramétrique associé est une loi multivariée. Dans son travail de thèse [48] Nour-Eddine Lasmar présente une étude des distributions paramétriques multivariées. Le test de Kolmogorov-Smirnov n'est pas utilisé car la partition I_k de l'espace de grande dimension est difficilement manipulable.

La figure V.5 est extraite de la thèse de Nour-Eddine Lasmar. Elle montre visuellement les différences entre le modèle paramétrique issu des modèles SIRV et la distribution empirique. Sur la colonne de gauche sont présentés des comparaisons entre les distributions des multiplieurs. Dans la colonne de droite sont présentées les distributions jointes. Pour le modèle du multiplieur, deux modèles paramétriques sont proposés, la distribution Weibull et la distribution Gamma.

La modélisation du multiplieur est précise là où la modélisation des distributions jointes sont très régulières contrairement à la distribution empirique. Nous montrons dans la suite de ce mémoire que le modèle SIRV permet d'obtenir des performances de classification surpassant la modélisation univariée. Ce qui fait de ce modèle une solution technique même avec une modélisation non fine.

Cette section montre que les modèles paramétriques ne suivent pas parfaitement les distributions empiriques. La distribution GΓD présente de meilleures statistiques au test Kolmogorov-Smirnov que la distribution GGD. Utiliser la distribution GΓD est donc le meilleur choix pour modéliser la distribution univariée des coefficients d'ondelettes. Les modèles SIRV et la GΓD sont des apports récents à la classification d'images texturées et l'avenir présentera de nouvelles distributions utilisables dans ce même domaine. Le but de ce document est de fournir les outils nécessaires pour réaliser une classification d'images texturées avec ces nouveaux modèles paramétriques.

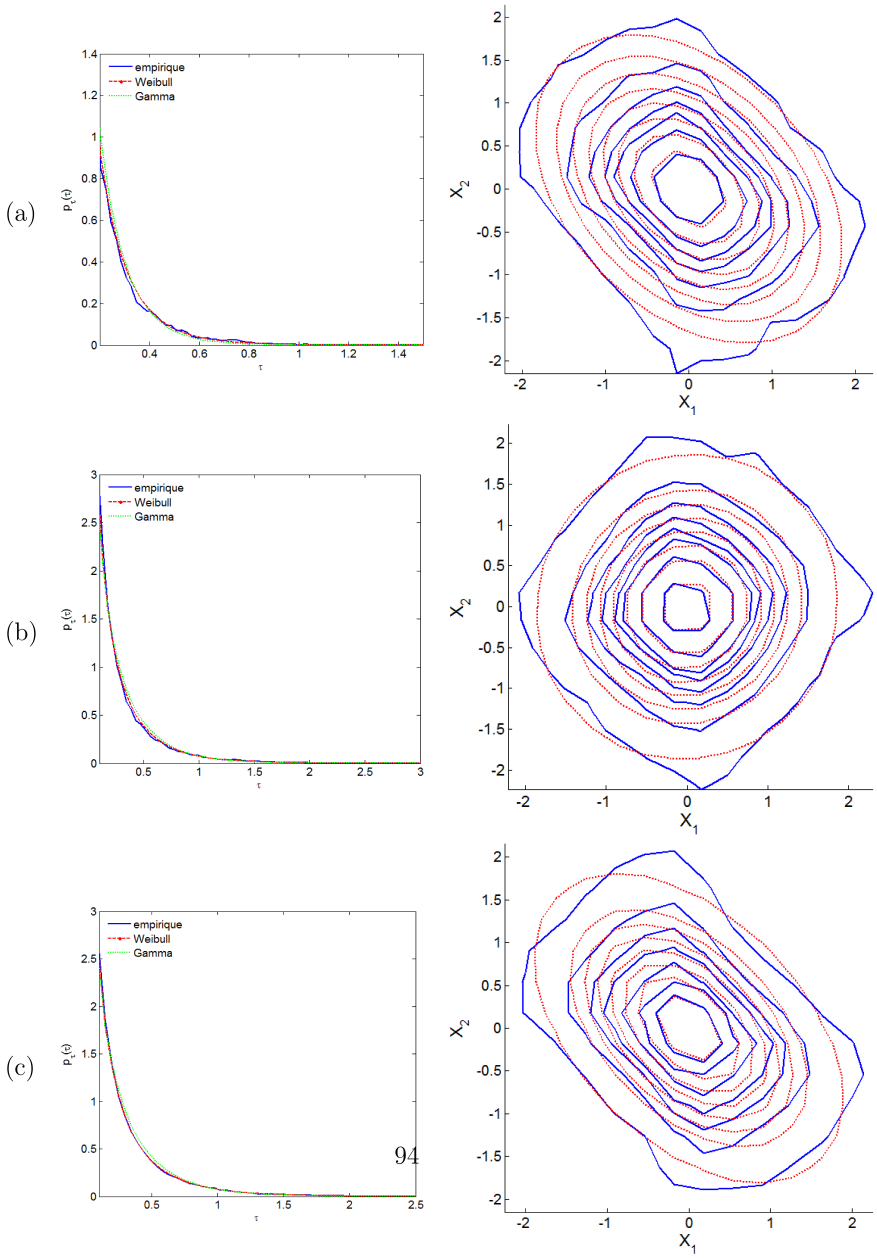


Figure V.5 – Première colonne : ajustement de la loi caractéristique de la densité caractéristique du modèle SIRV $p_\tau(\tau)$ par la distribution Weibull et Gamma. Deuxième colonne : ajustement de la partie gaussienne par une distribution gaussienne multivariée. Thèse de Nour-Eddine Lasmar

2.3 Représentation de la variété Θ

Cette section montre une validation du modèle hiérarchique développé dans le chapitre 2. Un cluster est caractérisée par un barycentre $\bar{\theta}$ et d'une variance σ^2. Comme une coupe transversale d'une distribution gaussienne est une ellipse, le cluster est supposé avoir une forme elliptique. Tous le problème repose alors sur la représentation de ces descripteurs paramétriques qui sont de grande dimension. La section suivante s'intéresse à une représentation de l'espace des descripteurs paramétriques pour valider la modélisation hiérarchique.

Une variété est décrite (cf. section 4.2) comme un espace courbe munie d'une géométrie qui lui est propre : la géométrie riemannienne. Au moyen des principes de la géométrie de l'information, l'estimateur des hyper paramètres $\bar{\theta}$ et σ^2 ont été développés. La section IV.3.1 présente une discussion sur la position *a priori* du barycentre $\bar{\theta}$ par rapport à la collection de K vecteurs paramétriques $(\theta_k)_{k=1}^K$. Le barycentre θ_0 est le point de la variété qui est élément du plan tangent. Il est alors vu comme relatif à la position du plan tangent. Néanmoins la géométrie de l'information [83] indique qu'il existe des fonctions (les connexions affines) reliant le plan tangent à l'origine avec notre plan tangent.

Dans la suite de cette section nous montrerons deux représentations possibles de l'espace Θ. Dans un premier temps, nous présenterons une projection de l'espace Θ de dimension 2 dans le plan. Dans un second temps, nous présenterons une représentation basée sur les dissimilarités existantes entre plusieurs vecteurs paramétriques (valable pour une dimension supérieure à 2).

2.3.1 Représentation par sous-bande d'une image

Soient $(x_n)_{n=1}^N$ une collection de variables aléatoires indépendantes et identiquement distribuées telles que : elles vérifient l'indépendance inter-bandes ; elles suivent un modèle paramétrique dont le vecteur paramétrique θ est de dimension 2. Dans cette thèse, la GGD est un modèle paramétrique remplissant ces deux critères et sera utilisée à titre d'exemple.

Soit $\theta = (\theta_1, \theta_2)$ le vecteur paramétrique associé au GGD où $\theta = (\alpha, \beta)$ avec α le paramètre d'échelle et β le paramètre de forme. La projection de la distribution paramétrique dans le plan consiste à utiliser les valeurs θ_1 et θ_2 comme coordonnées dans le plan euclidien. Dans le cas de la GGD, nous parlons de plan (α, β).

Ce type de représentation permet d'évaluer la position dans l'espace des vecteurs de descripteurs. Ensuite il est possible d'apprécier la position des barycentres orientés et symétriques.

Soit l'exemple de la GGD, la figure V.6 présente des vecteurs paramétriques $(\theta_k)_{k=1}^K$ estimés au sens

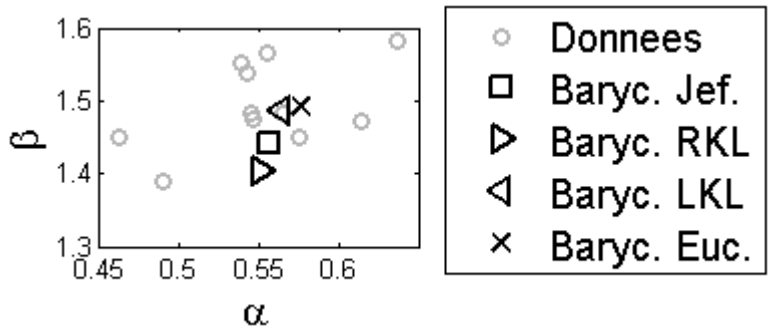


Figure V.6 – Représentation de vecteurs paramétriques dans le plan (α, β). La collection de vecteurs paramétriques $(\theta_k)_{k=1}^K$ représentés par des cercles gris. Quatre barycentres sont calculés. Le barycentre au sens de la distance euclidienne est représenté par une croix noire (respectivement le barycentre au sens de la divergence de Jeffrey est représenté par un carré noir, les barycentres orientés à gauche et à droite sont représentés par un triangle noir orienté à gauche et à droite). VisTex 6 sous-bande 2.

du maximum de vraisemblance par des cercles gris, dans le plan (α, β). Sur cette collection, quatre barycentres sont calculés. Le barycentre au sens de la distance euclidienne

$$\bar{\theta}_{\text{mo}} = \frac{1}{K} \sum_{k=1}^{K} \theta_k,$$

qui est la moyenne arithmétique, est représentée par une croix noire. $\bar{\theta}_{\text{mo}}$ est utilisée comme initialisation des algorithmes d'optimisation (cf. section 3.1) pour l'estimation des trois autres barycentres $\bar{\theta}^L, \bar{\theta}$ et $\bar{\theta}^R$. Le barycentre au sens de la divergence de Kullback-Leibler à gauche

$$\bar{\theta}^L = \underset{\theta}{arg\,min} \sum_{k=1}^{K} \text{KL}(\theta \parallel \theta_k)$$

est représenté par un triangle noir orienté vers la gauche. De même le barycentre au sens de la divergence de Kullback-Leibler à droite

$$\bar{\theta}^R = \underset{\theta}{arg\,min} \sum_{k=1}^{K} \text{KL}(\theta_k \parallel \theta)$$

est représenté par un triangle noir orienté vers la droite. Enfin le barycentre au sens de la divergence de Jeffrey

$$\bar{\theta} = \underset{\theta}{arg\,min} \sum_{k=1}^{K} J(\theta, \theta_k)$$

est représenté par un carré noir. Aucun des barycentres n'est confondu, montrant que les mesures de dissimilarité décrivent chacune une variété distincte. Les deux barycentres orientés à gauche et à droite sont donc clairement séparés, montrant l'importance de prendre en compte leur orientation. Le barycentre au sens de la divergence de Jeffrey $\bar{\theta}$ se situe entre les barycentres orienté à gauche $\bar{\theta}^L$ et à droite $\bar{\theta}^R$.

Ce qui coïncide avec le théorème de Nielsen et Nock [106] sur la position du barycentre symétrique à l'intersection de la géodésique reliant $\bar{\theta}^L$ à $\bar{\theta}^R$ et le bisecteur (ensemble de points aussi similaires à $\bar{\theta}^L$ et $\bar{\theta}^R$).

Cette représentation par projection est insuffisante pour apprécier les dissimilarités entre les images. Une image est représentée par un vecteur de descripteurs $\theta_{s,o}$ pour chaque sous-bande d'échelle s et d'orientation o. Dans l'exemple de la GGD, ce sont 6 vecteurs $\theta_{s,o}$ par image alors que cette représentation n'est qu'une projection sur une seule sous-bande. La dimension réelle du descripteur paramétrique est donc $6 \times 2 = 12$, ce qui est plus difficilement représentable. Dans la prochaine sous-section, les mesures de dissimilarité entre les images seront utilisées pour calculer une projection du nuage plus fidèle.

2.3.2 Représentation par la distance entre les images (PCA)

La représentation correcte des dissimilarités entre les images n'est possible qu'avec une réduction de dimension. En effet, la variété paramétrique Θ est immergée dans un espace de plus grande dimension. Une interprétation graphique des dissimilarités adaptatives est fournie par l'algorithme de cartographie des descripteurs isométriques (isomap) [146].

Algorithme V.1 : Pseudo-code de l'algorithme de cartographie des descripteurs isométriques

Données : Un ensemble de N_{Tr} images $(I_i)_{i=1}^{N_{\mathrm{Tr}}}$, un paramètre d'affichage $\sigma > 0$
Résultat : Les vecteurs propres V et les valeurs propres S

pour chaque (i, j) *une combinaison d'index d'images dans la base de données* **faire**
  $D_1(i,j) \leftarrow J(I_i, I_j)$;
fin
$D_2 \leftarrow \mathrm{Dijkstra}(D_1)$; `/* Réduction des dissimilarités par le choix du plus court chemin */`
$W \leftarrow \exp\left\{-\frac{1}{2}\sigma^{-2} D_2^2\right\}$; `/* Les dissimilarités sont utilisées comme poids */`
$\{F, S\} \leftarrow \mathrm{ACP}(W)$;

L'algorithme V.1 présente l'implémentation de l'isomap sur une variété paramétrique. La première étape de l'algorithme est l'évaluation des dissimilarités entre les images I_i. Mais la divergence de Jeffrey ne vérifie pas l'inégalité triangulaire, ce qui signifie que pour deux images I_i et I_j il peut exister une image I_k telle que

$$J(I_i, I_j) \geqslant J(I_i, I_k) + J(I_k, I_j)$$

La divergence de Jeffrey est une approximation de la distance riemannienne qui est la longueur de la géodésique reliant les deux points. La géodésique est la courbe la plus courte de Θ qui relie les deux vecteurs paramétriques θ et θ'. Un algorithme de plus court chemin permet de trouver les vecteurs

paramétriques intermédiaires entre θ et θ' de façon à ce que le parcours soit le plus court. L'algorithme Dijkstra [147] est un exemple d'algorithme de plus court chemin. De cette façon l'algorithme V.1 calcule une dissimilarité plus faible entre chaque vecteur et le stocke dans la matrice D_2.

L'algorithme V.1 utilise une analyse en composantes principales (ACP) afin de mettre en avant les plus grandes dissimilarités existantes dans les images I_i. Pour y parvenir, la distance minimale D_2 doit être transformée en énergie W. Cette énergie étant maximale lorsque le dissimilarités sont faibles, donc les images sont plus similaires. L'ACP est alors effectuée sur la matrice des poids W, elle retourne une collection F de vecteurs propres et les valeurs propres S. Le vecteur propre F_1 associé à la plus forte valeur propre contient un ensemble de coordonnées dans $\mathbb{R}$ pour chaque images I_i (de même pour F_2). Utiliser conjointement les vecteurs F_1 et F_2 permet d'obtenir une projection dans le plan à partir des dissimilarités mesurées entre les images. La réduction de dimension apportée par l'algorithme V.1 permet d'apprécier les dissimilarités réelles entre les images I_i.

Figure V.7 – Nuage de points projetés dans le sous-espace : (a) pour des images texturées présentant une faible diversité intra-classe et (b) pour des images texturées présentant plusieurs orientations.

La figure V.7 montre un nuage de points qui est le projeté en dimension 2 d'un ensemble d'images I_i. Le carré noir positionne le barycentre $\bar{\theta}$ estimé. F_1 et F_2 sont les deux vecteurs propres associés respectivement à la plus forte valeur propre et la seconde plus forte. Dans la figure V.7.(a), les images I_i ont une très faible diversité intra-classe. Les dissimilarités entre les images sont faibles et toutes les images peuvent être confondues. Dans la figure V.7.(b), les images I_i ont une forte diversité intra-classe introduite par des orientations différentes. Les dissimilarités entre deux images avec des orientations distinctes est plus élevée que la dissimilarité entre deux images de même orientation. Les images sont alors réparties en trois clusters séparés. Notez également que l'unique barycentre calculé est équidistant des trois clusters, ce qui peut pousser à utiliser plutôt 3 barycentres pour représenter cette collection d'images.

Deux procédures d'affichage ont été décrites dans cette section et doivent être perçues comme des

outils :

représentation d'une sous-bande : contrôler le déroulement des méthodes d'optimisation

plan des dissimilarités : visualiser la diversité intra-classe des images

Mais ce qui est plus utile que la visualisation des images pour la valorisation de l'algorithme de classification développé dans ce document, c'est l'évaluation des performances présenté dans la section suivante.

2.4 Évaluer les performances de classification

Cette section montre comment définir une évaluation automatique des performances de classification. L'état de l'art est utilisé ici comme illustration des concepts évoqués.

2.4.1 Matrice de confusion

Chellappa et Chatterjee [148] étudient la classification d'images texturées et souhaitent présenter les performances de l'algorithme qu'ils ont développé. Pour cela ils présentent des matrices de confusion [7, 13] C (aussi connues comme matrice d'erreurs). La base de données contient un nombre de classes N_c qui défini la dimension de la matrice C. Pour chaque classe, un nombre N_{Tr} d'images est utilisé pour faire la phase d'apprentissage de l'algorithme, tandis que les N_{Te} images restantes forment l'ensemble de test de l'algorithme. Pour une image test I, soit i la classe dont elle est extraite, aussi appelée texture *a priori*. L'algorithme de classification, dans sa phase de test, renvoie un entier j comme la texture estimée de l'image. La coordonnée verticale i de C correspond à la texture *a priori* de l'image et la coordonnée horizontale j correspond à la texture estimée de l'image. La valeur $C(i,j)$ est le nombre d'images de texture *a priori* i et de texture estimée j. Le cas optimal correspond donc à des $C(i,j)$ pour $i \neq j$ tous nuls et $C(i,i)$ égaux à N_{Te}.

Pour un algorithme de classification, cette matrice de confusion donne des informations sur les classes dont les images sont les plus confondues, indiquant par la même les problèmes de classification qui sont à résoudre pour obtenir une matrice de confusion C diagonale. Dans la section suivante, une valeur statistique est calculée sur la matrice de confusion pour présenter les performances de classification.

2.4.2 Précision globale moyenne

Gomez et Montero [149] définissent la précision globale moyenne

$$O^c = \frac{1}{N_c N_{\mathrm{Te}}} \sum_{i=1}^{N_c} C(i,i) \tag{V.1}$$

comme la fréquence moyenne des images correctement estimées. Pour l'algorithme de classification optimal, la matrice de confusion est diagonale, autrement dit $C(i,i) = N_{\text{Te}}$ ce qui conduit à avoir $O^c = 1$. Un algorithme de classification est donc optimal pour la base de données si la précision globale moyenne vaut 1. La valeur ainsi calculée est indépendante du nombre de classes. Cette valeur évalue globalement l'algorithme de classification là où la matrice de confusion est plus exhaustive.

2.4.3 Statistique Kappa

Cohen [150] introduit la statistique Kappa, c'est une évaluation des performances de classification qui enlève l'aléatoire du calcul. La statistique Kappa se calcule à partir de la précision globale moyenne :

$$K = \frac{O^c - p_e}{1 - p_e} \tag{V.2}$$

avec p_e le pourcentage d'éléments classifiés correctement par chance

$$p_e = \frac{1}{(N_c N_{\text{Te}})^2} \sum_{j=1}^{N_c} \left(\sum_{i=1}^{N_c} C(i,j) \right) \left(\sum_{i=1}^{N_c} C(j,i) \right).$$

Par définition, le nombre d'images appartenant à la classe j est connu avec

$$N_{\text{Te}} = \sum_{i=1}^{N_c} C(j,i)$$

ce qui nous permet de réécrire le pourcentage p_e comme

$$p_e = \frac{N_{\text{Te}}}{(N_c N_{\text{Te}})^2} \sum_{j=1}^{N_c} \sum_{i=1}^{N_c} C(i,j).$$

Ensuite une classe a été estimée pour chaque image de test, signifiant que toutes les images sont présentes :

$$N_c N_{\text{Te}} = \sum_{j=1}^{N_c} \sum_{i=1}^{N_c} C(i,j)$$

Ce qui permet de conclure que le pourcentage d'éléments classifiés correctement par chance est de

$$p_e = \frac{1}{N_c} \tag{V.3}$$

Considérons deux exemples de classifications que nous devons comparer. Le problème posé consiste à

classer $12+8+4+8 = 32$ images réparties dans $N_c = 4$ classes différentes. Autrement dit la seconde classe est composée de $N_{Te} = 12$ images, la troisième classe $N_{Te} = 4$ et $N_{Te} = 8$ pour les deux classes restantes. Les deux algorithmes de classification renvoient un estimé aléatoire : le premier n'étant pas uniforme (voir tableau V.1.(a)). La matrice de confusion asymptotique est remplie par la valeur $C_2(i,j) = N_{\mathrm{Te}}/N_c$ pour tout couple (i,j) (voir tableau V.1.(b)).

Les précisions globales des deux algorithmes sont égales : $O_1^c = O_2^c = 1/4$. Calculons le pourcentage $p_{e,1}$ d'éléments classifiés correctement pour les algorithmes :

$$p_{e,1} = \frac{9\times 8 + 8\times 12 + 6\times 4 + 9\times 8}{32\times 32} = \frac{1}{4} + \frac{1}{128}, \qquad p_{e,2} = \frac{8\times(8+12+4+8)}{32\times 32} = \frac{1}{4}$$

Nous montrons par ses exemples que le pourcentage d'éléments classifiés correctement par chance varie si le nombre d'images de test par classe n'est pas uniforme.

i/j	1	2	3	4
1	0	2	3	3
2	3	5	3	1
3	1	1	0	2
4	5	0	0	3

(a)

i/j	1	2	3	4
1	2	2	2	2
2	3	3	3	3
3	1	1	1	1
4	2	2	2	2

(b)

Tableau V.1 – Deux exemples de matrices de confusion avec $N_c = 4$ classes et des classes moins bien réparties. L'algorithme de classification renvoie : (a) plusieurs valeurs (b) un estimé uniformément aléatoire.

L'état de l'art de la classification d'images texturées est maintenant terminé, tous les outils ont été présentés dans les deux premiers chapitres et le reste de ce chapitre est consacré à la présentation des performances de classification de différents algorithmes. La première section porte sur une approche issue de la littérature qui a été étendue. La seconde section porte sur la méthode des « Sac de Mots Visuels » avec des descripteurs paramétriques.

3 Classification basée sur une distribution barycentrique (1-CB)

Choy et Tong [44] proposent une extension de la classification basée barycentre en utilisant une divergence de Kullback-Leibler à la place de la distance euclidienne usuelle. La classification basée distribution caractéristique, ainsi définie, est présentée dans la première sous-section. Les deux sous-sections suivantes présentent des résultats de classification.

3.1 Principe et extensions du 1-CB

La classification basée barycentre est définie dans un espace $\mathbb{R}^d$ de dimension d muni de la métrique euclidienne. Soient $p_{i,k}$ le point représentant la $k^{\text{ième}}$ image de la classe i. Les points $p_{i,n}$ de la classe i sont supposés normalement distribués dans l'espace $\mathbb{R}^d$. Par conséquent la classe est représentée par le barycentre

$$\bar{p}_i = \frac{1}{K}\sum_{k=1}^{K} p_{i,k}$$

et la variance σ_i autour du barycentre. Supposons alors la variance unitaire $\sigma_i^2 = 1$. La classification basée barycentre du point p_{Te} consiste à calculer la distance euclidienne entre p_{Te} et le barycentre $\bar{p}_i$ de chaque classe i. La classe estimée $\hat{i}$ correspond alors au barycentre le plus proche de p_{Te}

$$\hat{i} = \underset{i=1,\ldots,N_c}{arg\,\min} \|p_{\text{Te}} - \bar{p}_i\|^2.$$

Le principe général a plus de 50 ans (voir Julesz [9]) et il est adapté aux images texturées par Choy et Tong [44].

3.1.1 Définition du 1-CB

Dans ce document, une image est représentée par la distribution $p_{i,k}$ des coefficients de trames par analyse. Les trois hypothèses, évoquées au chapitre III, sont faites : indépendance inter-bandes, indépendance intra-bande et modélisation paramétrique. Par conséquent, une image est représentée par le vecteur paramétrique $\theta_{i,k}$ qui pilote la distribution $p_{i,k}$. La mesure de dissimilarité privilégiée est la divergence de Jeffrey (Choy et Tong présentent l'algorithme avec la divergence de Kullback-Leibler).

L'algorithme est constitué de deux phases : la phase d'apprentissage et la phase de test. La phase d'apprentissage consiste à estimer le barycentre $\bar{\theta}_i$ pour chaque classe de textures (nommé distribution caractéristique par Choy et Tong). Pour réaliser la phase de test, l'image test qui est représentée par θ_{Te} est comparée en similarité avec les barycentres $\bar{\theta}_i$ de chaque classe i. L'estimation de la classe $\hat{i}$ est le barycentre $\bar{\theta}_i$ le plus similaire à θ_{Te}

$$\hat{i} = \underset{i=1,\ldots,N_c}{arg\,\min} J(\theta_{\text{Te}}, \bar{\theta}_i).$$

La divergence de Jeffrey offre une mesure de dissimilarité équivalente dans la variété Θ et dans l'espace des distributions. L'algorithme complet est nommé classification basée sur 1 barycentre (1-CB).

1-CB est un algorithme qui ne prend pas en compte la dispersion réelle des vecteurs paramétriques

$\theta_{i,k}$ sur la variété Θ, l'extension que nous proposons est d'inclure la variance σ_i^2 dans le calcul. Ce qui permet de remonter jusqu'à la formulation de la distribution gaussienne concentrée sur une variété.

3.1.2 La classification basée barycentre et variance (1-CVB)

L'algorithme d'extension du 1-CB conserve les deux phases du 1-CB, la phase d'apprentissage et la phase de test. Une classe d'images texturées est une réalisation d'une distribution gaussienne concentrée sur la variété Θ [84]. Cette distribution gaussienne concentrée est paramétrée par le barycentre $\bar{\theta}_i$ et la variance σ_i^2 qui sont estimés pendant la phase d'apprentissage. La phase de test est modifiée dans le sens que la classe i estimée correspond au couple $(\bar{\theta}_i, \sigma_i^2)$ maximisant la probabilité d'appartenance de l'image test θ_{Te}

$$\hat{i} = \underset{i=1,\ldots,N_c}{arg\,max}\, p(\theta_{\mathrm{Te}} \mid \bar{\theta}_i, \sigma_i^2) \simeq \underset{i=1,\ldots,N_c}{arg\,max}\, \frac{1}{\sigma_i^d (2\pi)^{d/2}} \exp\left\{ -\frac{1}{2\sigma_i^2} J(\theta_{\mathrm{Te}}, \bar{\theta}_i) \right\}.$$

L'algorithme ainsi défini est nommé classification basée barycentre et variance (1-CVB). Cet algorithme apporte plus de souplesse au modèle 1-CB et permet de réduire les erreurs de classification. Néanmoins, lorsque la diversité intra-classe est présente dans certaines base de données les performances de classification de 1-CVB sont majorées. De ce fait, nous proposons une nouvelle extension du 1-CB dans la section suivante pour mieux gérer le problème de la diversité intra-classe.

3.1.3 La classification basée multi-barycentres (K-CB)

La diversité intra-classe impose une structure morcelée aux descripteurs paramétriques. Empiriquement, les descripteurs paramétriques d'une classe avec de la diversité intra-classe se présente sous la forme de clusters. Chaque cluster est supposé être un ensemble de réalisations d'une distribution gaussienne concentrée sur la variété paramétrique Θ. Alors, la classe d'image texturée est l'ensemble des réalisations d'un mélange de gaussiennes concentrées. L'algorithme proposé est encore constitué de deux phases : apprentissage et test.

Actuellement, le nombre de clusters K utilisés (et donc le nombre de distributions gaussiennes concentrées dans la loi mélange pour une classe) est fixé *a priori* [65, 67, 75]. La phase d'apprentissage consiste à estimer les hyper paramètres de K distributions gaussiennes concentrées (poids $w_{i,k}$, barycentre $\bar{\theta}_{i,k}$ et variance $\sigma_{i,k}^2$). La phase de test consiste à trouver le mélange de gaussiennes concentrées qui maximise la vraisemblance pour le descripteur paramétrique θ_{Te}

$$\hat{i} \simeq \underset{i=1,\ldots,N_c}{arg\,max} \sum_{k=1}^{K} w_{i,k} \frac{1}{\sigma_{i,k}^d (2\pi)^{d/2}} \exp\left\{ -\frac{1}{2\sigma_{i,k}^2} J(\theta_{\mathrm{Te}}, \bar{\theta}_{i,k}) \right\}.$$

Comme simplification, considérons le cadre optimal ou les distributions gaussiennes sont à support disjoint. Pour un descripteur paramétrique θ_{Te} de test, l'aspect supervisé de notre approche permet de connaître le cluster k_{Te} et la classe i_{Te} d'appartenance de ce descripteur. Par conséquent, le cadre optimal indique qu'il existe un réel ϵ strictement positif tel que :

$$\begin{cases} \frac{1}{\sigma_{i,k}^d (2\pi)^{d/2}} \exp\left\{-\frac{1}{2\sigma_{i,k}^2} J(\theta_{\mathrm{Te}}, \bar{\theta}_{i,k})\right\} \leqslant \epsilon & \text{si } (i,k) \neq (i_{\mathrm{Te}}, k_{\mathrm{Te}}) \\ \frac{1}{\sigma_{i_{\mathrm{Te}},k_{\mathrm{Te}}}^d (2\pi)^{d/2}} \exp\left\{-\frac{1}{2\sigma_{i_{\mathrm{Te}},k_{\mathrm{Te}}}^2} J(\theta_{\mathrm{Te}}, \bar{\theta}_{i_{\mathrm{Te}},k_{\mathrm{Te}}})\right\} > \epsilon & \text{sinon} \end{cases}$$

L'inégalité obtenue de la définition permet de majorer la probabilité calculée par le mélange. Premier cas avec $i \neq i_{\mathrm{Te}}$:

$$\sum_{k=1}^{K} w_{i,k} \frac{1}{\sigma_{i,k}^d (2\pi)^{d/2}} \exp\left\{-\frac{1}{2\sigma_{i,k}^2} J(\theta_{\mathrm{Te}}, \bar{\theta}_{i,k})\right\} \leqslant \sum_{k=1}^{K} w_{i,k}\epsilon = \epsilon$$

Second cas avec $i = i_{\mathrm{Te}}$

$$\begin{aligned} &\sum_{k=1}^{K} w_{i_{\mathrm{Te}},k} \frac{1}{\sigma_{i_{\mathrm{Te}},k}^d (2\pi)^{d/2}} \exp\left\{-\frac{1}{2\sigma_{i_{\mathrm{Te}},k}^2} J(\theta_{\mathrm{Te}}, \bar{\theta}_{i_{\mathrm{Te}},k})\right\} \\ &\qquad \leqslant \frac{w_{i_{\mathrm{Te}},k_{\mathrm{Te}}}}{\sigma_{i_{\mathrm{Te}},k_{\mathrm{Te}}}^d (2\pi)^{d/2}} \exp\left\{-\frac{1}{2\sigma_{i_{\mathrm{Te}},k_{\mathrm{Te}}}^2} J(\theta_{\mathrm{Te}}, \bar{\theta}_{i_{\mathrm{Te}},k_{\mathrm{Te}}})\right\} + \sum_{k=1, k\neq k_{\mathrm{Te}}}^{K} w_{i_{\mathrm{Te}},k}\epsilon \\ &\qquad < \epsilon + \frac{1}{\sigma_{i_{\mathrm{Te}},k_{\mathrm{Te}}}^d (2\pi)^{d/2}} \exp\left\{-\frac{1}{2\sigma_{i_{\mathrm{Te}},k_{\mathrm{Te}}}^2} J(\theta_{\mathrm{Te}}, \bar{\theta}_{i_{\mathrm{Te}},k_{\mathrm{Te}}})\right\} \end{aligned}$$

Autrement dit, dans le cadre optimal et en sachant que ϵ est indépendant de i et de k, que la règle de décision revienne à écrire

$$\begin{aligned} \hat{i} &\simeq \underset{i=1,\ldots,N_c}{arg\max} \left[\sum_{k=1}^{K} w_{i,k} \frac{1}{\sigma_{i,k}^d (2\pi)^{d/2}} \exp\left\{-\frac{1}{2\sigma_{i,k}^2} J(\theta_{\mathrm{Te}}, \bar{\theta}_{i,k})\right\}\right] \\ &\simeq \underset{i=1,\ldots,N_c}{arg\max} \left[\max_{k=1,\ldots,K} \frac{1}{\sigma_{i,k}^d (2\pi)^{d/2}} \exp\left\{-\frac{1}{2\sigma_{i,k}^2} J(\theta_{\mathrm{Te}}, \bar{\theta}_{i,k})\right\}\right] \end{aligned}$$

Ensuite, par stricte positivité du logarithme népérien, nous obtenons :

$$\hat{i} \simeq \underset{i=1,\ldots,N_c}{arg\min} \min_{k=1,\ldots,K} \left[\log\{\sigma_{i,k}\} + \frac{1}{2\sigma_{i,k}^2} J(\theta_{\mathrm{Te}}, \bar{\theta}_{i,k})\right] \tag{V.4}$$

L'algorithme de classification basé sur K barycentres (K-CB) correspond à l'algorithme précédent avec l'hypothèse d'homoscédasticité ($\sigma_{i,k}^2 = 1$, pour tout couple (i,k)). Dans la suite de cette section, les résultats de différents tests de performances sont affichés et comparés.

3.2 Résultats et discussion

3.2.1 Apport de la loi *a priori* intrinsèque

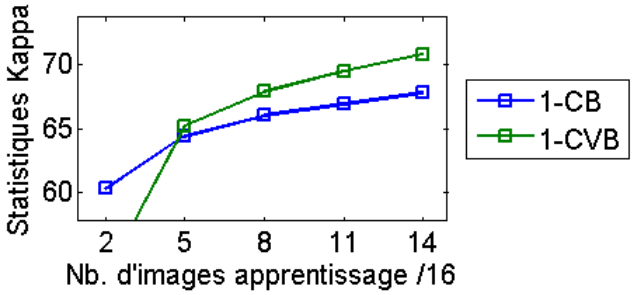


Figure V.8 – Performance de l'algorithme de classification 1-CB et 1-CVB. Modèle paramétrique univarié GΓD et distance euclidienne. Base de données VisTex

La figure V.8 montre l'évolution de la statistique Kappa en fonction du nombre d'images utilisées pour la phase d'apprentissage. Comme les images d'apprentissage ont leur importance pour avoir une bonne représentativité du barycentre, les images sont sélectionnées aléatoirement. Tous les résultats présentés sont des valeurs moyennes sur 100 lancés Monté-Carlo. La base de données utilisée ici est la base de données VisTex et le modèle paramétrique, avec supposition d'indépendance inter-bandes et intra-bande, est la GΓD. Dans un premier temps, la distance euclidienne est utilisée comme mesure de dissimilarité. Cette mesure est considérée extrinsèque à la variété Θ mais le barycentre calculé coïncide avec la moyenne arithmétique, de complexité calculatoire plus faible. La courbe bleue présente les performances de classification d'un algorithme 1-CB (respectivement la courbe verte présente les performances de classification d'un algorithme 1-CVB).

Avec seulement 2 images d'apprentissage, l'estimateur de la variance est biaisé, ce qui a pour conséquence de faire baisser les statistiques Kappa d'une dizaine de points entre les algorithmes 1-CB et 1-CVB. Néanmoins, pour plus de 5 images d'apprentissage, la variance σ_i^2 dans les performances de classification apporte un gain de 3 points.

La figure V.9 montre l'évolution de la statistique Kappa en fonction du nombre d'images utilisées pour la phase d'apprentissage. Les conditions de l'expérience sont similaires à l'expérience conduisant à la figure V.8 mis à part que la divergence de Jeffrey est utilisée comme mesure de dissimilarité intrinsèque à la variété Θ. Le gain en performances de classification entre les algorithmes 1-CVB et 1-CB n'est visible que pour un minimum de 8 images d'apprentissage. Ce qu'il faut noter entre les figures V.9 et V.8 c'est le gain en statistiques Kappa de 20 points par l'utilisation d'une mesure de dissimilarité intrinsèque à la variété. Ce gain en performances justifie une complexité calculatoire plus élevée de l'algorithme.

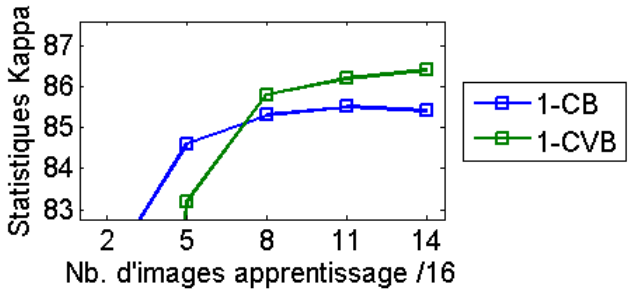


Figure V.9 – Performance de l'algorithme de classification 1-CB et 1-CVB. Modèle paramétrique univarié GΓD et divergence de Jeffrey. Base de données VisTex

3.2.2 Classification 1-CVB contre 1-CB

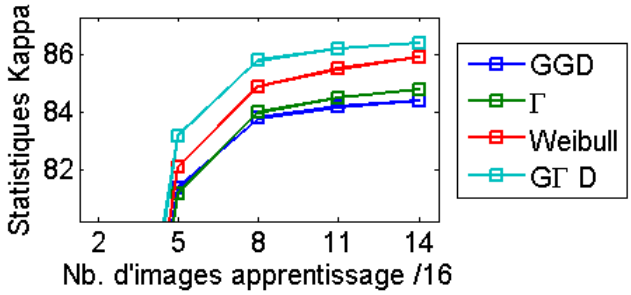


Figure V.10 – Performance de l'algorithme de classification 1-CVB avec la divergence de Jeffrey. 4 modèles paramétriques univariés la GGD, la distribution Gamma, la distribution Weibull et la GΓD. Base de données VisTex

La figure V.10 montre l'évolution de la statistique Kappa en fonction du nombre d'images d'apprentissage d'un algorithme 1-CVB. Comme montré dans la section précédente, la mesure de dissimilarité intrinsèque est adaptée à la géométrie pour l'implémentation de l'algorithme 1-CVB. Pour améliorer les performances de classification, il est possible de modifier le modèle paramétrique utilisé. Dans la figure V.10, une courbe bleue présente un algorithme utilisant une GGD (respectivement une courbe verte pour une distribution Gamma, une courbe rouge pour une distribution Weibull et une courbe cyan pour une GΓD). Les 4 modèles comparés dans cette figure sont univariés - distributions GGD, GΓD, Weibull, Gamma - et correspondent aux modèles comparés pour le test Kolmogorov-Smirnov de la figure V.4 (voir page 134). Ici, utiliser la distribution GΓD comme modèle stochastique montre des statistiques Kappa supérieures quel que soit le nombre d'images d'apprentissage. Sans perte de généralité, la GΓD est le modèle le plus fidèle aux distributions empiriques et la GΓD est le modèle univarié permettant d'obtenir l'algorithme le plus performant.

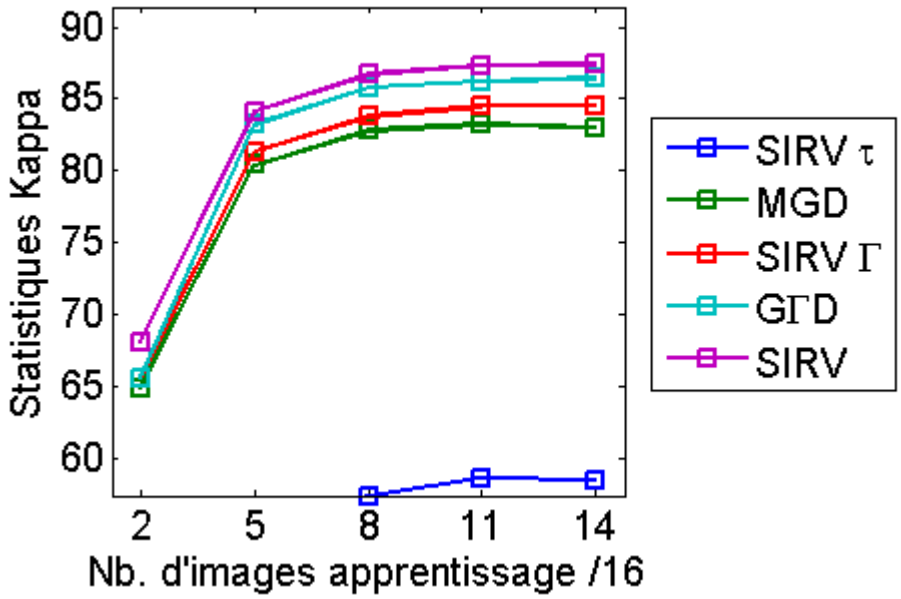


Figure V.11 – Performance de l'algorithme de classification 1-CVB avec la divergence de Jeffrey. Comparaison du modèle GΓD avec des modèles multivariés comme la distribution gaussienne multivariée et le modèle SIRV avec distribution Weibull pour le multiplieur. Pour un modèle SIRV, deux expériences supplémentaires sont réalisées : seule la matrice de covariance M est utilisée ou seule le multiplieur τ est utilisé. Base de données VisTex

3.2.3 Classification multivariée contre univariée

La figure V.11 montre l'évolution de la statistique Kappa en fonction du nombre d'images d'apprentissage d'un algorithme 1-CVB. La GΓD est le modèle paramétrique univarié le plus efficace sur cette base de données, ce modèle est alors comparé à différents modèles multivariés (la courbe de couleur cyan correspond à l'algorithme utilisant la GΓD). Soit l'existence d'une dépendance intra-bande dans un voisinage 3×3 autour de chaque coefficient de trame par analyse. Une approche simple est d'utiliser une distribution gaussienne multivariée dans le 1-CVB, la courbe de couleur verte correspond aux performances obtenues. Le 1-CVB avec la distribution gaussienne multivariée donne des statistiques Kappa inférieure de 4 points par rapport au 1-CVB univarié, donc ce modèle est insuffisant.

Le modèle SIRV est le modèle multivarié présenté dans ce document pour réaliser de la classification d'images texturées. Ce modèle est paramétré par une matrice de covariance M et un multiplieur τ. La mesure de dissimilarité entre deux modèles SIRV à multiplieur Weibull est approchée par la mesure de dissimilarité entre les distributions jointes $\vec{Y} = (\vec{g}, \tau)$. Pour démontrer que la distribution jointe est nécessaire, trois implémentations distinctes de 1-CVB sont proposées. La courbe bleue correspond à un 1-CVB où seul τ est utilisée, les statistiques Kappa de cette implémentation sont inférieures à 60%. La courbe rouge correspond à une implémentation de 1-CVB avec seule la matrice de covariance M. Cette

implémentation est similaire à une distribution gaussienne multivariée car elle n'est paramétrée qu'avec une matrice de covariance M. Mais l'estimation de la matrice M du modèle SIRV est plus efficace de 2 points qu'une simple matrice de covariance. De plus, l'algorithme 1-CVB univarié est plus efficace que les implémentations 1-CVB basées uniquement sur τ ou M.

Enfin, la courbe magenta montre les statistiques Kappa d'un 1-CVB avec la loi jointe $\vec{Y}$ approchant le modèle SIRV. L'utilisation de la distribution jointe est plus efficace que l'utilisation des marginales et présente un gain de 2 points par rapport à un 1-CVB univarié. Dans cette section, nous montrons que chercher à modéliser les dépendances intra-bandes est réalisable. De plus les performances de classification sont supérieures à celles d'une classification univariée. Afin de poursuivre sur la modélisation de la diversité intra-classe, la sous-section suivante présente les performances de classification du K-CB.

3.3 Classification à l'honneur : K-CB avec modèle multivarié

3.3.1 Résultats comparatifs

Nous présentons ici des résultats de classification avec 3 clusters par classe. La figure V.12 présente les performances de classification en statistique Kappa en fonction du nombre d'images d'apprentissage.

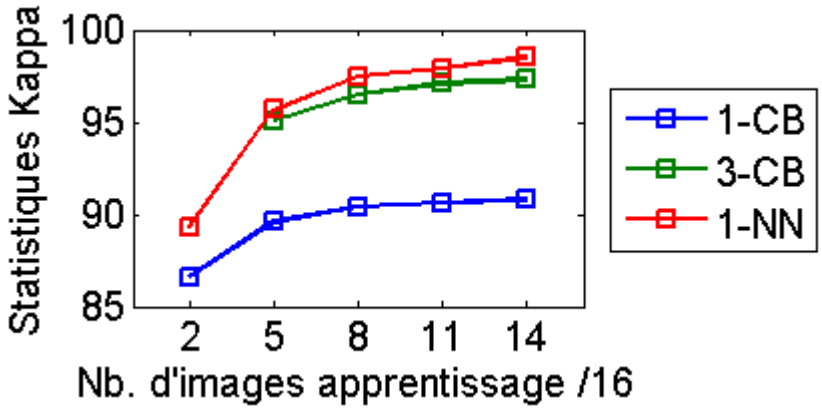


Figure V.12 – Performance de l'algorithme de classification 1-CB, 3-CB et 1-NN. Modèle paramétrique multivarié SIRV et divergence de Jeffrey. Base de données VisTex

Le modèle SIRV est utilisé pour modéliser les dépendances spatiales dans les sous-bandes de décomposition. La figure V.11 montre qu'utiliser un modèle multivarié est préférable à l'utilisation d'un modèle univarié pour la classification. La courbe bleue de la figure V.12 correspond aux performances obtenues par un algorithme de classification basée sur un barycentre. Sans perte de généralité, nous considérerons ici que les paramètres de variances sont égaux.

La diversité intra-classe a pour effet de créer des clusters au sein des descripteurs paramétriques d'une

même classe. Nous proposons d'utiliser l'algorithme de clustering « modèle de mélange de gaussiennes concentrées » pour l'estimation des $N_{cl} = 3$ clusters présents dans les classes de la base de données VisTex. Le fait de fixer *a priori* le nombre de clusters augmente la probabilité d'apparition de sur-segmentation des classes, ce qui à une conséquence pour la complexité calculatoire de l'estimation des clusters. Sans perte de généralité, le nombre de clusters $N_{cl} = 3$ permet d'observer un gain en performances pour l'algorithme de classification.

La courbe verte de la figure V.12 montre les performances sous forme de statistiques Kappa de l'algorithme 3-CB par rapport au nombre d'images d'apprentissage. Remarquons qu'avec seulement 2 images d'apprentissage, aucune performances du 3-CB n'est enregistrée. La question de l'estimation de 3 barycentres à partir d'un ensemble d'au maximum 2 descripteurs paramétriques n'est pas abordé dans ce mémoire.

L'algorithme des K plus proches voisins (K-NN) est un algorithme de classification sans phase d'apprentissage. Tous les descripteurs paramétriques des images d'apprentissage sont conservés pour la phase de test. Lors de la phase de test, le descripteur paramétrique est comparé, au moyen d'une mesure de dissimilarité, à chacun des descripteurs paramétrique d'apprentissage. Puis les K descripteurs paramétriques présentant la plus forte similarité avec le descripteur de test sont considérés. La classe la plus représentée parmi les K plus proches voisins représente la classe estimée de l'image test. L'algorithme de classification 1-CB et 3-CB ne considèrent que le barycentre le plus proche. L'algorithme 1-NN ainsi défini est comparable aux algorithmes 1-CB et 3-CB du point de vue de l'implémentation. L'algorithme 1-NN présente des statistiques Kappa supérieures aux implémentations 1-CB et 3-CB.

Phase	Construction dictionnaire	Phase Test
1-CB	$6[2(81K+1)+N_1+N_2N_3]$ opérations	$6N_cN_5$ opérations
3-CB	$3N_4(6[2(81K+1)+N_1+N_2N_3]+KN_5)$ opérations	$3(6N_cN_5)$ opérations
1-NN	Gratuit : assignation des K images d'apprentissage	$6KN_cN_5$ opérations

Tableau V.2 – Complexité calculatoire de trois algorithmes 1-CB, 3-CB et 1-NN

Une simple méthode des plus proches voisins donne des performances statistiques Kappa maximales. Nous défendons notre 3-CB par une discussion sur le complexité calculatoire.

La table V.2 présente les complexité calculatoire des implémentations 1-CB, 3-CB et 1-NN. La première colonne correspond à la complexité calculatoire lors de l'apprentissage du dictionnaire. Lors de l'extraction des descripteurs paramétriques, les images sont décomposées en 6 sous-bandes ($N_s = 2$ échelles de décomposition et $N_o = 3$ orientations de décomposition). La distribution des coefficients d'on-

delette au sein d'une sous-bande est modélisée avec un modèle SIRV. Pour un modèle SIRV, le barycentre est constitué d'une matrice M et d'un réel a strictement positif. Supposons tout d'abord le calcul d'un unique barycentre pour la classe (implémentation du 1-CB). Dans ce cas, le barycentre est composé d'une matrice de covariance $\bar{M}$ et d'un réel $\bar{a}$. La matrice $\bar{M}$ est le résultat d'une dichotomie entre le barycentre suivant la divergence de Kullback-Leibler à gauche $\bar{M}^L$ et le barycentre suivant la divergence de Kullback-Leibler à droite $\bar{M}^R$. Nous supposerons que le nombre de boucles dans cette dichotomie est égale à N_1. Ensuite, le barycentre $\bar{M}^L$ est obtenu suite à la somme des K matrices M_k estimées sur les images d'apprentissage. Les matrices M sont de dimension 9×9 ce qui fait 81 réels par matrice et donc la moyenne $\bar{M}^L$ c'est plus de $81K$ additions suivi d'une division par K. Le barycentre $\bar{M}^L$ nécessite $81K+1$ opérations pour être calculé, tout comme le barycentre $\bar{M}^R$. En remontant au calcul du barycentre $\bar{M}$, ce sont $2(81K + 1) + N_1$ opérations qui sont nécessaires.

Pour le calcul du réel $\bar{a}$, une descente de gradient est choisie. Cette descente de gradient est réalisée en N_2 boucles. Cette descente de gradient minimise une fonction de coût une somme de K formules, chacune contenant 10 calculs de fonction Gamma, 1 calcul de fonction Digamma, 17 divisions, 8 multiplications, 2 puissances non entières, 2 logarithmes et 13 additions. Disons que cette fonction de coût nécessite N_3 opérations. Cela signifie que dans l'implémentation du 1-CB, le calcul d'un texton requiert $6[2(81K + 1) + N_1 + N_2N_3]$ opérations.

Pour une implémentation du 3-CB, un algorithme 3-MLE est utilisé. Cet algorithme de clustering nécessite N_4 boucles pour converger et, à chaque boucle, calcule 3 barycentres. Ensuite l'étape d'association nécessite de calculer $3K$ divergences de Jeffrey (qui nécessite N_5 opérations). Le nombre d'opérations nécessaires pour le calcul des 3 textons pour la classe est égal à :

$$3N_4(6[2(81K + 1) + N_1 + N_2N_3] + KN_5)$$

Pour une implémentation du 1-NN, toutes les images d'apprentissage sont utilisées comme autant de représentants de la classe. Le coût est celui de créer de la place dans la mémoire pour ranger les K descripteurs paramétriques.

Une fois le dictionnaire appris, il est possible de s'intéresser à la complexité calculatoire de la phase de test, le nombre d'opérations requises pour estimer la classe d'une image de test. Dans les trois algorithmes implémentés 1-CB, 3-CB et 1-NN, la phase de test est une opération de mesure de dissimilarité minimale. La mesure de dissimilarité choisie est une divergence de Jeffrey entre les distributions paramétriques de chaque sous-bande. La divergence de Jeffrey entre deux modèles SIRV est en fait une somme de

divergences de Jeffrey pour chaque distribution marginale. Par exemple, la divergence de Jeffrey entre deux distributions gaussiennes multivariées n'est autre que la somme de la trace d'une matrice produit avec la soustraction de 2 logarithmes de déterminant de matrices. Pour résumé, une divergence de Jeffrey entre 2 modèles SIRV va utiliser N_5 opérations.

Pour l'implémentation du 1-CB, l'image test est comparée avec l'unique texton de chaque classe. Si N_c est le nombre de classes, alors le nombre d'opérations pour la phase de test est de $6N_cN_5$. Pour une implémentation du 3-CB, il y a 3 textons par classe ce qui signifie que le nombre d'opérations a triplé. Enfin, pour l'implémentation du 1-NN, la phase de test requiert $6KN_cN_5$ opérations puisque les K descripteurs paramétriques extraits des images d'apprentissage de la classe sont utilisés comme représentants de cette classe.

La courbe rouge de la figure V.12 montre les performances sous forme de statistiques Kappa de l'algorithme 1-NN par rapport au nombre d'images d'apprentissage. La complexité calculatoire de la phase de test est plus élevée pour le 1-NN que pour le 3-CB, en effet avec 8 images d'apprentissage par classe le 1-NN calcule $8 * N_c$ dissimilarités là où le 3-CB calcule $3 * N_c$ dissimilarités.

3.3.2 Discussion

La figure V.12 montre que plus le nombre de représentants par clusters est élevé, meilleures sont les performances. En effet, le 1-CB utilise un barycentre par classe, le 3-CB utilise trois barycentres par classe et le 1-NN utilise le nombre d'images d'apprentissage N_{tr} de représentants par classe. L'algorithme 3-CB présente un bon compromis entre un faible nombre de représentants par classe et une performance de classification inférieure de moins d'un point des performances du 1-NN.

Les textures de la base de données VisTex sont plutôt homogènes en contenu, ce qui conduit l'algorithme 3-CB à sur-segmenter les classes. Sur-segmenter n'influence pas les performances de classification et augmente la complexité calculatoire. Mais certaines classes présentent des variations en luminosité au sein de la même classe. Sur ces quelques classes, l'algorithme de clustering permet de faire croître les performances de classification par rapport à un algorithme 1-CB.

L'expérience montre également qu'une recherche exhaustive permet d'obtenir des performances supérieures. Cet expérience est réalisée sur une base de donnée avec seulement 16 descripteurs paramétriques par classe, l'application finale travaillera avec un nombre de descripteurs paramétriques suffisamment élevé pour réduire la probabilité de mauvaise classification. Mais la complexité calculatoire de la phase de test en sera décuplée. La recherche exhaustive n'est pas une solution pratique.

Le travail sur les algorithmes de K-CB devrait alors se poursuivre avec le développement d'une estimation automatisée du nombre K de clusters par classe. Adapter le nombre K de clusters par classe permet de réduire de nouveau la complexité calculatoire de la phase de test. Par exemple, les critères d'information d'Akaike [76, 77] ou bayésiens [78] pourraient être envisagés dans l'optique d'établir automatiquement le nombre de clusters par classe.

Cette section du mémoire a présenté un ensemble d'expériences simples pour valider les choix fait précédemment : utiliser une mesure de dissimilarité, un choix du modèle stochastique et l'utilisation de barycentres et de variances. La section suivante utilise conjointement les choix pour réaliser la classification SMV avec descripteurs paramétriques.

4 Classification par sac de mots visuels

Les trois premiers chapitres de ce mémoire présentent la classification SMV basée sur des descripteurs paramétriques. Au moyen d'un exemple simple qu'est la classification basée barycentre, la section précédente illustre les choix effectués pour la manipulation des descripteurs paramétriques : le modèle stochastique, la divergence de Jeffrey, ou encore l'algorithme d'optimisation. Mais pour valider le modèle par rapport aux implémentations de la « littérature », il est nécessaire d'implémenter spécifiquement un algorithme SMV basé sur des descripteurs paramétriques.

Nous présentons dans cette dernière section les détails de l'implémentation de cette classification. À des fins de comparaisons, un descripteur patch proposé par Varma et Zisserman est également implémenté [10]. Plutôt que d'appliquer des filtres sur les patchs, Varma et Zisserman proposent de travailler directement les valeurs prises par les niveaux de gris des patchs. Nous comparerons les deux algorithmes par rapport au dimensions de leurs espace respectifs et au nombre de patchs utilisés.

La base de donnée CUReT est utilisée dans cette partie du document. Cette base de données présente plus de variabilité intra-classe que la base de donnée VisTex, ce qui constitue un problème de classification plus ardu qu'avec la base de donnée VisTex. Varma et Zisserman présentent des performances de 95,33 %, 95,62 % et 96,19% pour des descripteurs patchs de dimensions 3×3 pixels, 5×5 pixels et 7×7 pixels respectivement [10]. Les éléments fournis par Varma et Zisserman sont insuffisants pour la reproduction des performances de classification qu'il présentent.

Cette section de présentation de résultats est divisé en trois sous-sections. Premièrement, nous présenterons les détails de l'implémentation de l'algorithme de classification. Deuxièmement les performances

des implémentations seront présentées. Enfin, une troisième sous-section proposera une discussion sur les résultats présentés.

4.1 Détails de l'implémentation

4.1.1 Définition du patch

Le descripteur global qui est utilisé ici est plus complet que le descripteur obtenu après la phase d'apprentissage du SMV. Soient H, L et d_p trois entiers strictement positifs. De l'image $H \times L$ pixels sont extraites des sous-images de taille réduite $d_p \times d_p$, nommée patchs.

Soit v_p un entier strictement positif appelé disparité. La disparité entre deux patchs consécutifs représente la distance en pixels entre le coin supérieur gauche de chaque patch. Par exemple, si $v_p = d_p$ alors la disparité est suffisamment grande pour que les patchs ne se recouvrent pas. De plus, les patchs extraits de l'image ne peuvent pas être défini à l'extérieur de celle ci. Ce qui prévient tout problème d'effets de bords. Le nombre maximum de patchs positionnés horizontalement sur l'image est donc de

$$N_{Lp} = \frac{L - (d_p - 1)}{v_p} \tag{V.5}$$

Le patch ne peut pas être défini à l'extérieur de l'image, donc le coin supérieur gauche d'un patch ne peut pas être positionné au delà de $L - (d_p - 1)$. Ensuite la disparité v_p est appliquée, il s'agit de l'écart minimum entre deux coins. De même, le nombre maximum de patchs positionnés verticalement dans l'image est de

$$N_{Hp} = \frac{H - (d_p - 1)}{v_p} \tag{V.6}$$

Par conséquent, le nombre de patchs N_p extraits d'une image est égal à $N_{Hp}N_{Lp}$.

Le mélange de gaussiennes concentrées est calculé sur l'ensemble des N_p patchs d'une image plus les N_p patchs des autres images de la classe. Cette approche des descripteurs locaux conduit à une plus grande population de descripteurs pour estimer le barycentre et la variance. Ainsi la probabilité que l'estimateur se trompe est plus faible.

Il est important de noter que pour une disparité vérifiant $v_p > d_p$ alors certaines zones de l'image ne seront pas couverte. Une valeur trop élevée de la disparité conduit à une perte d'information.

4.1.2 La classification SMV par descripteurs patchs

Une approche de la littérature est implémentée et comparée avec notre proposition, la classification SMV par descripteurs patchs introduite par Varma et Zisserman [10]. Plutôt que d'optimiser le filtre appliqué au patchs, les auteurs proposent d'utiliser directement les niveaux de gris des patchs (après normalisation). Ce qui leur permet de proposer un algorithme SMV avec des performances supérieures à l'utilisation des filtres MR8 qu'ils proposaient en 2005 [11]. Varma et Zisserman proposent dans [10] un algorithme SMV basé sur une distribution jointe qui est plus efficace que l'algorithme SMV par descripteurs patchs. Sans perte de généralités, l'algorithme de classification par patchs est utilisé comme référence pour la paramétrisation de notre algorithme SMV basé sur les descripteurs paramétriques.

Voici quelque notations pour la suite de ce paragraphe, y_n est le descripteur patch et $(V_n)_{n=1}^{HL}$ représente les niveaux de gris de l'image. Trois étapes sont nécessaires pour passer des niveaux de gris $(V_n)_{n=1}^{HL}$ au descripteur y_n : une normalisation précédent l'extraction permet d'obtenir les valeurs normalisées $(t_n)_{n=1}^{HL}$ des niveaux de gris ; l'extraction convertie les valeurs t_n en valeurs $(s_n)_{n=1}^{d_p^2}$; une normalisation post extraction peut également être menée afin d'obtenir le descripteur y_n.

Une normalisation de l'image est effectuée afin de réduire l'impact que peut avoir une diversité intra-classe basée sur un éclairage plus puissant sur certaines images. Cette normalisation précède l'extraction des patchs. Soit $(V_n)_{n=1}^{HL}$ le niveau de gris des patchs vu comme une variable aléatoire réelle. Soient $\hat{\mu}$ et $\hat{\sigma^2}$ les estimateurs empiriques de la moyenne et de la variance de V_n respectivement. La normalisation d'une image est le résultat de l'opération

$$t_n = \frac{V_n - \hat{\mu}}{\hat{\sigma}} \tag{V.7}$$

L'extraction du patch consiste simplement à récupérer les valeurs normalisées pour le patch courant. Soit $(t_m)_{m=1}^{d_p^2}$ une sous-suite de $(t_n)_{n=1}^{HL}$ correspondant au patch i_p actuel. Alors le descripteur patch extrait s'écrit

$$s_m = t_m, \qquad \forall m = 1, \ldots, d_p^2 \tag{V.8}$$

Une normalisation des descripteurs locaux est réalisée après leur normalisation et extraction. Cette nouvelle normalisation a pour but de réduire les différences de contraste entre les descripteurs. Soit $(s_n)_{n=1}^{d_p^2}$ le descripteur patch extrait, vue comme une variable aléatoire. Malik et al. [151, 152] introduisent la normalisation

$$y_n = \frac{s_n}{||s_n||_2} \log\left\{1 + \frac{||s_n||_2}{0.03}\right\} \tag{V.9}$$

La mesure de dissimilarité utilisée pour le descripteur patch est une simple distance euclidienne. Cette géométrie est invariante aux normalisations effectuées, ce qui justifie son utilisation. Les choix de paramétrisation possible pour l'algorithme SMV par descripteurs patchs sont les entiers d_p, v_p. Dans [10], Varma et Zisserman indiquent que l'entier d_p prend comme valeurs 3, 5, 7, 9, ... Néanmoins la valeur de disparité v_p n'est pas évoquée. Elle peut donc valoir 1 ou d_p pour des patchs qui ne se recouvrent pas.

4.1.3 La classification SMV par descripteurs paramétrique

L'implémentation de la classification SMV par descripteurs paramétriques suivra l'implémentation du descripteur SMV par descripteurs patchs. Néanmoins, de puissants effets de bords peuvent apparaître lors de la décomposition en trames par analyse des patchs. Nous proposons alors, d'extraire les patchs après la décomposition en trames par analyse. Sans perte de généralité, une décomposition en ondelettes non décimée est choisie pour l'ensemble de ses tests. Les sous-bandes de décomposition sont de même dimension que l'image d'origine et l'extraction des patchs se fait de manière similaire.

Comme pour l'extraction de descripteurs patch, trois étapes sont réalisées consécutivement : normalisation pré-extraction, extraction et normalisation post-extraction. La première étape consiste à normaliser notre image au moyen de l'équation précédente (V.7).

Soit $(t_n)_{n=1}^{HL}$ les valeurs normalisées de l'image vues comme une variable aléatoire. Il est important de vérifier que l'image est de taille dyadique, autrement dit que H et L soient deux entiers égaux à une puissance de 2. La décomposition en ondelettes stationnaires conduit à l'obtention des coefficients de sous-bande $x_{n,s,o}$ pour l'échelle $s = 1, \ldots, N_s$ et l'orientation $o = 1, \ldots, N_o$. Soit $(x_{m,s,o})_{m=1}^{d_p^2}$ une sous-suite de $(x_{n,s,o})_{n=1}^{HL}$ correspondant au patch i_p actuel. Soit θ l'estimateur au sens du maximum de vraisemblance pour le modèle paramétrique choisi. Donc l'extraction des descripteurs se poursuit par l'estimation au sens du maximum de vraisemblance :

$$\hat{\theta}_{s,o} = \theta((x_{m,s,o})_{m=1}^{d_p^2}) \tag{V.10}$$

Ensuite, le descripteur paramétrique est exprimé comme la collection $\hat{\theta} = (\hat{\theta}_{s,o}, s =, 1, \ldots, N_s, o = 1, \ldots, N_o)$.

L'espace des descripteurs paramétriques lui peut ne pas être invariant à la normalisation des descripteurs. Sans perte de généralité, les descripteurs paramétriques ne sont pas normalisés dans l'algorithme SMV. Le descripteur paramétrique y s'écrit

$$y = \hat{\theta}$$

d_p	dimension descripteur
17	289
9	81

Tableau V.3 – Exemples de dimensions du descripteur patch en fonction de l'entier d_p

Ensuite la divergence de Jeffrey est la mesure de dissimilarité choisie pour le test.

La classification SMV par descripteurs paramétriques repose sur le choix du modèle paramétrique. En suivant les résultats obtenus par la classification basée barycentre (voir la section 3.2.3), le modèle SIRV est privilégié pour ce test. Dès lors, le nombre de paramètres pour cette classification est 2, les entiers d_p et v_p. Ensuite, la valeur de l'entier d_p doit être suffisamment élevée afin de permettre une estimation au sens du maximum de vraisemblance de $\hat{\theta}$. d_p vaudra alors 16, 32 ou encore 64 [4, 153]

4.1.4 Discussion autour de la dimension et du nombre

La comparaison des deux algorithmes se fera sur une même base de données et dans des conditions similaires. Cela signifie que les entiers d_p et v_p seront modifiés afin que le nombre de descripteurs locaux et la dimension de ses descripteurs locaux soient sensiblement équivalent. Les calculs suivants montrent qu'utiliser des valeurs entières pour d_p et v_p complique le calcul.

Premièrement considérons la dimension de l'espace des descripteurs. Le descripteur par patch contient l'ensemble des niveaux de gris d'un patch $d_p \times d_p$. Par conséquent, la dimension de ce descripteur vaut d_p^2. Par exemple, pour $d_p = 17$ nous avons une dimension de descripteur égal à 289 (voir table V.3).

Le descripteur paramétrique est lié au modèle SIRV sur un voisinage 3×3. Ce modèle paramétrique est constitué d'une matrice de covariance 9×9 et d'un paramètre de forme a. La dimension d'un estimé au sens du maximum de vraisemblance est la donnée d'une matrice de covariance qui est symétrique donc de dimension est de $9 * 10/2 = 45$ et la donnée d'un réel a de dimension 1. Un estimé $\hat{\theta}_{s,o}$ est considéré de dimension 46. Maintenant, la décomposition en ondelettes choisie réalise $N_o = 3$ sous-bandes par orientation pour une échelle, et $N_s = 2$ échelles de décomposition. Alors le descripteur paramétrique contient $3*2 = 6$ estimés $\hat{\theta}_{s,o}$. Par conséquent la dimension du descripteur paramétrique vaut $6*46 = 276$. Remarquez que la dimension est ici indépendante des entiers d_p et v_p.

Deuxièmement, le nombre de descripteurs extrait d'une image dépend de la dimension $H \times L$ de l'image puis des deux entiers d_p et v_p par le calcul

$$N_p = \frac{L - (d_p - 1)}{v_p} \times \frac{H - (d_p - 1)}{v_p} = \frac{d_p^2 - (2 + L + H)d_p + 1 + LH}{v_p^2} \tag{V.11}$$

d_p/v_p	17	9	4	2	1
17	49	169	841	3249	12544
9	64	196	961	3721	14400

Tableau V.4 – Nombre de descripteurs en fonction des entiers d_p et v_p. L'entier d_p représente la dimension des patchs en pixels par ligne (sachant que ces patchs sont carrés) tandis que l'entier v_p représente la distance en pixel minimale entre le centre de deux patchs différents. L'image de départ est de taille 128×128

Le tableau V.4 montre le nombre de descripteurs N_p qu'il est possible d'obtenir en fonction de deux entiers que sont la dimension d_p et la disparité v_p. Pour cette simulation, l'image est de dimensions 128×128 pixels. Notons que pour $d_p = 9$ et $v_p = 17$ toute l'image n'est pas prise en compte dans l'extraction des patchs, synonyme d'une perte d'informations lors de l'apprentissage.

Le tableau V.4 montre également que le nombre maximum de patchs dans une image est supérieur pour un patch de taille 9×9 pixels que pour un patch de dimensions 17×17. Le nombre de patchs doit alors être fixé à partir du patch de plus grande dimension. Pour la suite de cet ensemble de tests nous respecterons

— une classification SMV par descripteurs patchs avec $d_p = 9$ et $v_p = 9$

— une classification SMV par descripteurs paramétriques avec $d_p = 64$ et $v_p = 5$

Par conséquent, une image est représentée par $N_p = 196$ patchs. Le descripteur global étant la fréquence d'apparition des textons les plus similaires aux 196 patchs de l'image.

Notons que dans l'implémentation de Varma et Zisserman [10], les images complètes de dimension 200×200 pixels sont utilisées à la place de nos sous-images de dimension 128×128 pixels. Considérons un descripteur patch avec $d_p = 9$ et $v_p = 9$, 196 patchs sont extraits d'une image 128×128 pixels alors que 484 patchs sont extraits d'une image 200×200 pixels. Néanmoins, une telle population ne peut être atteinte avec les descripteurs paramétriques qui nécessitent des patchs de grande dimensions et une image de dimensions réduites à 128×128 pixels.

4.1.5 Nombre d'images pour l'apprentissage

La base de donnée CUReT est composé de 92 images par classe. Pour des soucis d'évaluation de l'algorithme de classification, cette base de donnée est découpée en 2. 46 images sont utilisées pour la phase d'apprentissage (et l'estimation des textons). Les 46 images restantes sont utilisées pour l'évaluation lors de la phase de test.

4.1.6 Nombre de textons par classe

Le nombre de textons par classe influe sur la complexité calculatoire et la performance de classification. La diversité intra-classe impose l'utilisation de N_{cl} clusters par classe avec $N_{cl} > 1$. De l'autre côté, utiliser $N_{cl} = K$ clusters, avec K le nombre d'images d'apprentissage, impose une forte charge lors de la phase de test. Sans perte de généralité, le nombre N_{cl} de cluster n'est pas estimé automatiquement mais fixé *a priori*.

Dans leurs résultats, Varma et Zisserman sont partis avec $N_{cl} = 10$ textons pour travailler sur la base de donnée CUReT. Nous allons donc travailler avec ce même nombre de textons par classe. Le descripteur global d'une image de CUReT avec 92 classes est de dimension $92 \times 10 = 920$. Le descripteur global étant le nom donné à l'histogramme obtenu en sortie de la phase d'apprentissage du SMV.

4.1.7 Représentation du cluster

Deux approches sont employées dans les résultats suivants. Dans la méthode de Varma et Zisserman les clusters ne sont représentés que par un texton. Cette approche coïncide avec un mélange de gaussiennes concentrées pour lesquelles la variance est supposée égale pour chaque distribution gaussienne. Cela conduit à la seconde approche qui estime sur chaque cluster le barycentre, la variance et le poids (proportion d'échantillons d'apprentissage dans ce cluster). Rappelons qu'estimer conjointement le barycentre et la variance puis les utiliser dans la partie test c'est révélé primordial en termes de performances de classification basée barycentre par rapport à n'utiliser qu'un barycentre.

4.2 Résultats de classification

4.2.1 Performances de classification

Le tableau V.5 présente les performances obtenues par des implémentations de classification SMV. Trois descripteurs différents sont utilisés ici : les descripteurs patchs extraits d'une sous-image de dimension 128×128 pixels, les descripteurs patchs extraits d'une sous-image de dimension 200×200 pixels et les descripteurs paramétriques extraits de patchs de dimension 64×64 pixels avec au moins 5 pixels entre chaque centre de patchs. Les implémentations sont au nombre de trois : une classification avec 2 clusters par classe et des clusters représentés par un barycentre, une variance et un poids, une classification avec 10 clusters par classe et des clusters représentés par un barycentre, une variance et un poids, une classification avec 10 clusters par classe et des clusters représentés uniquement par un barycentre.

Descripteurs	2 clusters/classe barycentre + variance	10 clusters/classe barycentre + variance	10 clusters/classe barycentre uniquement
Patchs	46,27%	52,02%	53,04%
Patchs	63,08%	71,59%	72,56%
SIRV Weibull 3x3	81,09%	90%	88,66%
Descripteurs	dimen. descr.	nb. descr.	
Patchs	81	9016	
Patchs	81	22264	
SIRV Weibull 3x3	276	9016	

Tableau V.5 – Performances données en statistiques Kappa pour une classification SMV sur la base de données CUReT. La troisième ligne présente des descripteurs patchs de dimension 9×9 pixels avec un minimum de 9 pixels entre le centre de deux patchs, l'image est avant réduite à une dimension de 128×128 pixels. La quatrième ligne présente des descripteurs similaires à la troisième ligne sauf que l'image est aux dimensions d'origine 200×200. Enfin la cinquième ligne présente les descripteurs paramétriques estimés sur des patchs de dimension 64×64 pixels avec au minimum 5 pixels entre les centre des patchs, l'image est avant réduite à une dimension de 128×128 pixels.

La ligne 2 (respectivement ligne 3 et ligne 4) du tableau V.5 correspond à un descripteur patch extrait d'une sous-image de dimension 128×128 pixels (resp. descripteur patch extrait d'une sous-image de dimension 200×200 pixels et descripteur paramétrique extrait d'une sous-image de dimensions 128×128 pixels).

La colonne 2 (respectivement colonne 3 et colonne 4) du tableau V.5 correspond à une implémentation SMV avec 2 clusters par classe plus variance (resp. 10 clusters par classe plus variance et 10 clusters par classe sans variance). La colonne 5 du tableau V.5 présente la dimension des descripteurs et la colonne 6 du tableau V.5 représente le nombre de descripteurs d'apprentissage pour cette implémentation.

4.3 Discussion autour des résultats

Peu de résultats sont présentés dans cette section pour valider notre proposition. Nous discuterons séquentiellement sur le nombre de descripteurs disponibles pour représenter une image puis la comparaison de notre algorithme de classification SMV avec descripteurs paramétriques avec un algorithme proposé par Varma et Zisserman.

La première discussion porte sur le nombre de descripteurs. Un descripteur patch peut être extrait d'une sous-image de dimension 128×128 pixels, comme observé dans les tableaux V.5, ou d'une sous-image de dimension 200×200 pixels, comme observé dans les tableaux V.5. La différence entre les deux types d'extraction étant de $484 - 196 = 288$ descripteurs par image d'apprentissage. Un gain de 12 à 20 points en statistique Kappa est observable. La bonne approche serait d'augmenter encore le nombre de descripteurs pour atteindre les performances de plus de 96 % proposées par la littérature [10].

Pour les descripteurs paramétriques, les patchs extraits sont de plus grande dimension et leur nombre ne peut pas atteindre le nombre de descripteurs patchs observés dans le paragraphe précédent. Il convient donc de se comparer à population de descripteurs équivalent. Ce qui permet de montrer des performances équivalentes entre notre implémentation de classification SMV avec descripteurs paramétriques et l'implémentation de classification SMV avec descripteurs patchs.

La seconde discussion porte sur le comportement des algorithmes avec une base de données complète (voir tableau V.5). Malheureusement, différents facteurs ne nous permettent pas de réaliser une centaine de lancés Monté-Carlo pour ce test avec la base de données complète (complexité calculatoire, nombre de boucles dans les algorithmes de clustering, qualité de l'implémentation, utilisation d'un langage haut-niveau). Nous présentons alors un lancé Monté-Carlo pour la base de donnée complète. Pour l'ensemble des tests, les images d'apprentissage sont les mêmes avant extraction des descripteurs.

Dans ces même conditions, l'implémentation de classification SMV avec descripteurs paramétriques montre un gain en statistiques Kappa de l'ordre de 20 points par rapport à l'implémentation de la classification SMV par descripteurs patchs.

Une implémentation de classification SMV avec 10 clusters par classe fait mieux qu'une implémentation de classification SMV avec 2 clusters par classe puisque la diversité intra-classe est mieux modélisée. C'est pour cette raison que les statistiques Kappa montrent un gain de 8 points pour l'implémentation de la classification SMV avec 10 clusters par classe.

Un autre test effectué est une implémentation de classification SMV avec des clusters représentés uniquement par le barycentre. Utiliser la variance et le poids en plus du barycentre permet d'améliorer les statistiques Kappa de l'implémentation de classification SMV de 1 point par rapport au fait de n'utiliser que le barycentre. Malheureusement, l'implémentation de classification SMV par descripteur patch ne présente pas les performances présentées par Varma et Zisserman [10] et l'implémentation de classification SMV par descripteur paramétrique ne propose pas de performances équivalentes.

Nous avons implémenté un algorithme de classification SMV basé sur des descripteurs paramétriques et montré sont intérêt en le comparant en terme de performances à des implémentations de classification SMV basé sur des descripteurs patchs. Avec un nombre de descripteurs équivalent et un nombre de clusters par classe suffisamment élevé (supérieur à 2) alors les deux implémentations sont équivalentes.

Pour comparer le nombre de descripteurs, nous avons défini une description du découpage d'une image au moyen de deux valeurs que sont la largeur des patchs carrés et la distance minimum entre les centre

de deux patchs avec le pixel comme unité. Nous présentons alors les calculs explicites pour estimer le nombre de descripteurs. La classification SMV avec des descripteurs paramétriques est donc une option valable pour la classification d'images texturées. Malheureusement, la complexité calculatoire du calcul du barycentre au sens de la divergence de Jeffrey est un dernier frein pour envisager une implémentation temps réel.

5 Conclusion

Ce chapitre présente le contexte applicatif du travail de recherche. Il existe actuellement plus d'une dizaine de bases de données d'images texturées représentant autant d'applications possibles. Nous proposons de résoudre ce problème de classification avec des descripteurs paramétriques. Nous avons validé les modèles proposés en les comparant aux distributions empiriques, en présentant des résultats de classification basée barycentre et en présentant des descripteurs globaux estimés sur des images. Les résultats montrent le gain progressif pour la modélisation choisie de la diversité intra-classe dans la variété riemannienne des descripteurs paramétriques. Le recours aux techniques de clustering est justifié par les gains en performance observée. L'utilisation d'un modèle pour la dépendance spatiale est justifiée par un autre gain en performance.

Si nous revenons à la classification SMV, nous avons observé que pour notre implémentation la classification SMV basée sur des descripteurs paramétriques présente de meilleures performances de classification qu'un descripteur patch. Il reste encore du travail avant d'atteindre les performances affichées dans la littérature mais tout converge dans le sens de notre proposition. La modélisation des classes d'images texturées au moyen d'un mélange de gaussiennes concentrées est un apport net à la solution de classification. De son côté, le descripteur paramétrique démontre une grande fidélité au contenu textural par rapport aux descripteurs de la littérature.

La prochaine étape pour l'implémentation d'un algorithme de classification basé sur des descripteurs paramétriques est la validation sur un plus grand nombre de bases de données. Cet objectif, quoi que simple, se heurte au problème de la dimension des bases de données des images texturées. Plus de classes et plus de diversité intra-classe augmente la complexité calculatoire de la phase d'apprentissage de l'algorithme de classification. L'impact sur la phase de test de la dimension des bases de données est plutôt linéaire en comparaison. Donc la complexité calculatoire ne permet pas de lancer suffisamment de test pour optimiser les paramètres de l'algorithme qui ont été présentés dans les trois premiers chapitres de ce mémoire.

Chapitre VI

Conclusion générale et perspectives

1 Conclusion du travail effectué

Dans ce mémoire, nous nous sommes attachés à explorer une méthode supervisée de classification d'images texturées exploitant la modélisation probabiliste paramétrique. Nous nous sommes notamment focalisés sur l'approche basée « dictionnaire » car, d'une part, l'approche à dictionnaire est un bon moyen de tenir compte de la diversité intra-classe qui peut être très marquée dans le cas des images texturées et d'autre part elle n'a pas à notre connaissance été développée dans le cadre paramétrique. En effet, concernant les approches SMV ou à dictionnaire, l'état de l'art fait ressortir, comme méthode de référence, le travail proposé par Varma et al. qui proposent une approche non-paramétrique par « patches » [10]. Nous avons donc décidé d'explorer le formalisme paramétrique car de nombreux articles récents ont montré la pertinence de ce type d'approches en indexation ou en classification d'images texturées. Une des limitations de ces travaux récents est notamment qu'ils ne prennent pas en compte la diversité intra-classe ce qui limite fortement leurs performances. Il était donc d'intérêt d'étudier la possibilité d'intégrer dans l'approche paramétrique la notion de diversité via la construction d'un dictionnaire adapté.

Sur un plan théorique, notre motivation concernant l'exploration de la voie paramétrique s'est aussi fortement appuyée par le fait que la recherche d'un dictionnaire dans l'espace paramétrique nécessite la définition d'une nouvelles forme de lois *a priori* : la classe des lois intrinsèques. Il s'avère en effet que cette classe de lois *a priori* est encore très peu explorée par la communauté ce qui rend donc son étude intéressante et en outre elle s'inscrit dans le cadre élégant du modèle hiérarchique bayésien à deux niveaux. Le caractère intrinsèque de ces lois *a priori* est à opposer aux lois usuelles utilisées en bayésien qui sont extrinsèques aux variétés paramétriques que nous ciblons, notamment les lois conjuguées. Dans ce contexte, nous avons proposé des lois inspirées de la littérature pour étendre notamment la distribution

gaussienne sur la variété riemannienne appelée aussi gaussienne concentrée. L'originalité est notamment de prendre en compte un modèle utilisant la divergence symétrique de Jeffrey, la distance riemannienne n'admettant pas de forme explicite. La divergence de Jeffrey symétrique présente notamment l'avantage de posséder une forme explicite en termes de paramètres pour les modèles de vraisemblance que nous utilisons. Les modèles de vraisemblance ou d'attache aux données que nous avons considérés sont d'une part les distributions univariées de types gaussienne généralisée (GGD) ou Gamma généralisée (GΓD) et d'autre part pour les lois multivariées, dans le cas de la dépendance spatiale intra-bande, les modèles SIRV avec multiplieur de type Weibull. Pour ces différentes lois de vraisemblance, nous avons défini des lois *a priori* exploitant une seule distribution gaussienne concentrée ou un mélange de gaussiennes concentrées. La loi mélange nous permet notamment de caractériser chaque classe par une collection de composantes ou clusters représentés par une position $\bar{\theta}$, d'une variance σ^2 et d'un poids w.

Les aspects modélisations étant fixés, nous nous sommes attachés à définir des approches algorithmiques génériques afin d'estimer les différents paramètres de chaque classe en supposant le nombre de composantes du mélange fixé et commun à toutes les classes. Les algorithmes ainsi définis concernent notamment la phase d'apprentissage de la méthode de classification supervisée. Comme nous l'avons montré, les algorithmes proposés sont issus de l'hypothèse d'un modèle hiérarchique bayésien sous-jacent. Nous avons considéré pour cela une vraisemblance marginalisée et une recherche de solution issue d'une approximation de type Laplace qui est une approximation en $\mathcal{O}(1/N)$ où N désigne le nombre d'observations, ici proportionnel au nombre de patches et au nombre de pixels par patch.

Dans un premier temps, nous avons proposé des algorithmes visant l'estimation du barycentre et de la variance dans le cas d'une seule composante et pour les cas des GGD et GΓD. Pour cela, nous étendons au cas des variétés l'algorithme d'optimisation proposé par Amari et Douglas [109] nommé d'adaptation du gradient naturel. Nous montrons également que cette approche converge plus rapidement qu'une méthode de Newton-Raphson ou une descente de gradient usuelle. L'algorithme est maitrisé dans son évolution afin de le contraindre à rester sur la variété riemannienne ce qui lui permet de converger. Dans le cas du modèle multivarié SIRV, des concessions ont été faites afin de pouvoir proposer une approche opérationnelle car il n'existe pas de forme analytique d'une divergence pour ce modèle. Nous avons considéré un schéma numérique séparé entre l'estimation du barycentre pour le multiplieur Weibull et celui pour la matrice de covariance $\bar{M}$. Cela revient à considérer la loi jointe entre une gaussienne multivariée et une distribution Weibull pour le multiplieur.

Dans un second temps, nous nous sommes intéressés à la loi mélange. L'estimation des paramètres du

mélange de gaussiennes concentrées n'utilise pas d'algorithme Espérance-Maximisation qui est classique pour résoudre ce problème mais un algorithme nommé k-MLE inspiré des travaux de [95] avec qui nous avons collaboré. Sa construction est similaire à celle d'un algorithme de k-moyennes et présente des garanties de convergence plus rapide qu'un algorithme Espérance-Maximisation.

Pour la phase de décision et pour aboutir à un schéma complet de classification, nous avons évidemment pris en compte le modèle hiérarchique ce qui nous a permis de proposer une étape de décision tenant compte de l'*a priori*. Un point intéressant dans ce schéma est que la décision, intégrant une loi *a priori* intrinsèque, se ramène à une décision directement sur la variété. Contrairement au cas usuel du test de vraisemblance, sans *a priori* et nécessitant d'estimer une vraisemblance dans l'espace des données ce qui est donc très coûteux en calcul. Cette propriété a l'avantage de ne plus faire intervenir les données dans la décision. Mais revient à ne considérer que les paramètres estimés de l'attache aux données. Cette propriété se traduit par une distance intrinsèque peu couteuse en calcul pour construire l'histogramme empirique pour le codage par dictionnaire ou pour estimer la loi *a posteriori* liée à la loi mélange.

Finalement, considérant les phases d'apprentissage et de décision, nous avons une chaîne complète de classification par apprentissage d'un dictionnaire dans le cas où le descripteur est une collection de valeurs de paramètres pour des modèles probabilistes paramétriques. Pour notre application en texture, nous avons considéré les modèles de type GGD, GΓD ou SIRV sur les coefficients de sous-bandes car ils sont les principaux modèles proposés par la communauté. Nous avons développé les algorithmes nécessaires à l'estimation des différents paramètres caractérisant les classes et les échantillons de test. Nous avons étendu la classification basée barycentre, comme Varma et al. le proposent, au cas de la classification basé barycentre et variance [10]. Pour cela, les classes sont représentées par une variance en plus du simple barycentre. Nous montrons que la classification basée barycentre et variance propose de meilleures performances que la classification basée barycentre uniquement notamment dans le cas paramétrique. De même, une proposition a été faite sur l'implémentation d'un schéma de décision sur la base d'une distance entre histogrammes associé au dictionnaire. Nous avons pu observer que la phase d'apprentissage est l'étape la plus coûteuse et peut être réalisée hors ligne. Les différents développements d'algorithmes ont nécessité la prise en main d'outils et de concepts théoriques à l'intersection de nombreux domaines scientifiques en géométrie différentielle, en géométrie de l'information, en théories de la décision et du signal/image. L'ensemble des travaux de ce mémoire ont été valorisés par la rédaction d'articles de conférences internationales reconnues [62–67, 75, 84].

Sur le plan des résultats expérimentaux, même si il reste encore du travail, nous avons réalisés plusieurs

tests qui démontrent certaines tendances intéressantes permettant de mieux comprendre l'apport de l'approche paramétrique. Pour une base difficile, nous avons montré que l'approche paramétrique semble plus robuste à un nombre limité de données pour l'apprentissage ainsi qu'au nombre de composantes du mélange ou du dictionnaire que le cas non-paramétriques. Dans tous les cas, si nous disposons d'un dictionnaire large ainsi que d'une base de données d'apprentissage très fournie alors les méthodes semblent présenter les mêmes tendances en termes de performances.

2 Perspectives

Sur le plan des perspectives, elles sont évidemment nombreuses. Sans parler du problème du choix de la décomposition dans l'espace échelle/direction, nous avons l'opportunité notamment de développer les points suivants :

1. Nous nous sommes limités aux approches linéaires pour l'estimation des barycentres, à savoir les approches « gradient naturel » additives. Il nous reste à considérer des approches permettant de respecter de manière plus complète le caractère intrinsèque de l'espace dans lequel la descente est réalisée. Il s'agit d'étendre le cas du gradient euclidien à la notion de rétraction qui est un projeté sur la variété.
2. Dans notre approche nous avons, pour l'instant, considéré un nombre fixé de composantes du mélange. Évidemment, il serait intéressant de chercher à estimer ce paramètre afin de mieux adapter le modèle au contenu.
3. Nous avons considéré le cas d'estimation des textons à partir des seuls descripteurs locaux d'une classe. Il n'est pas certain que l'alternative consistant à estimer un dictionnaire sur tous les descripteurs locaux ne permette par de mieux traduire la diversité inter-classe.
4. Le dictionnaire dans le cas paramétrique peut devenir très creux. Que dire de la pertinence de prendre en compte cette caractéristique afin d'améliorer la performance de classification (distance adaptée pour la décision, adaptation des méthodes d'estimation etc.)
5. Pour la question de la convergence et de l'unicité du barycentre dans le cas de la divergence de Jeffrey, une piste de démonstration consisterait à considérer non pas la convexité classique mais la convexité géodésique
6. Concernant le modèle hiérarchique, nous avons proposé de résoudre l'intégrale de la vraisemblance marginalisée par l'approximation de Laplace. Évidemment, il est possible de proposer des alterna-

tives fondées sur les méthodes d'échantillonnage de type champs de Markov Monté-Carlo.

7. Une étude doit aussi être menée concernant la robustesse de l'approche paramétrique dans le cas où nous considérons que certains échantillons soient bruités : soit les échantillons de la base, soit les échantillons de tests ou les deux. L'approche paramétrique permettrait, par rapport à la méthode proposée par Varma, de tenir compte, dans le modèle général ou dans la construction des estimateurs, de la présence de bruit.

Annexe A

Existence et unicité du barycentre

1 Introduction

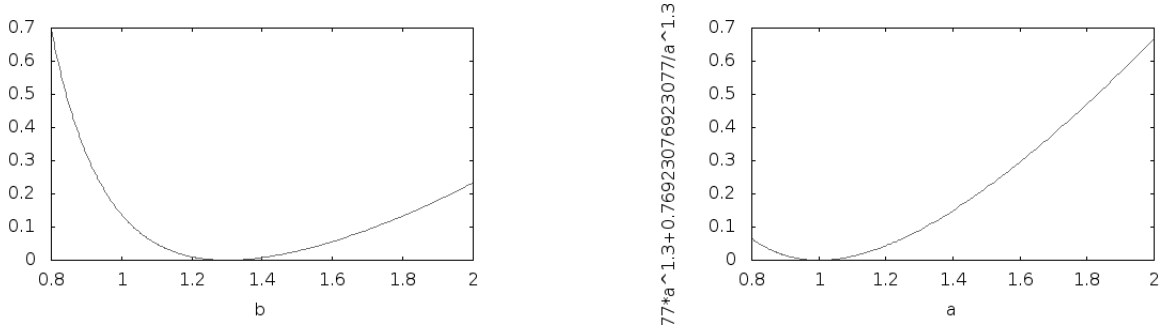
La distribution barycentrique est utilisée dans notre modèle de classification d'images texturées. Lors du chapitre 3 qui présente le barycentre, l'unicité du barycentre est montrée de façon expérimentale mais une approche théorique est proposée dans cette annexe.

2 Perte de convexité de la fonction coût

La stricte convexité d'une fonction nous assure de l'existence et de l'unicité du minimum de la fonction. De ce fait, la stricte convexité de la fonction coût nous donnerait l'existence et unicité du barycentre. Pour la vraisemblance GGD noté $p(x \mid \theta)$, la fonction de coût l est deux fois dérivable. Dans ce cas, l'étude de la matrice hessienne $\mathcal{H}l$ nous donnera la stricte convexité de la fonction de coût l. La méthode est simple, il suffit de montrer que les valeurs propres de la matrice hessienne $\mathcal{H}l$ sont strictement positives.

Pour cet exemple, l'espace Θ des paramètres est de dimension 2. Cela signifie que la matrice hessienne $\mathcal{H}l$ est une matrice symétrique définie positive 2×2. La table IV.2 nous explicite justement ses composants, néanmoins l'expression des valeurs propres est loin d'être explicite et manipulable. Nous nous tournons alors vers le déterminant et la trace de la matrice hessienne $\mathcal{H}l$. En effet, le produit des valeurs propres correspond au déterminant et la somme des valeurs propres correspond à la trace de la matrice hessienne $\mathcal{H}l$. Pour avoir la stricte positivité des valeurs propres il suffit d'avoir la stricte positivité du déterminant et de la trace de la matrice hessienne $\mathcal{H}l$.

Soit une variété constituée de points $(x, y, \|(x, y)\|)$, utilisant la distance à l'origine du point de coordonnées (x, y) comme valeur pour la troisième dimension. La surface ainsi crée est une fonction d'ordre 2



(a) (b)

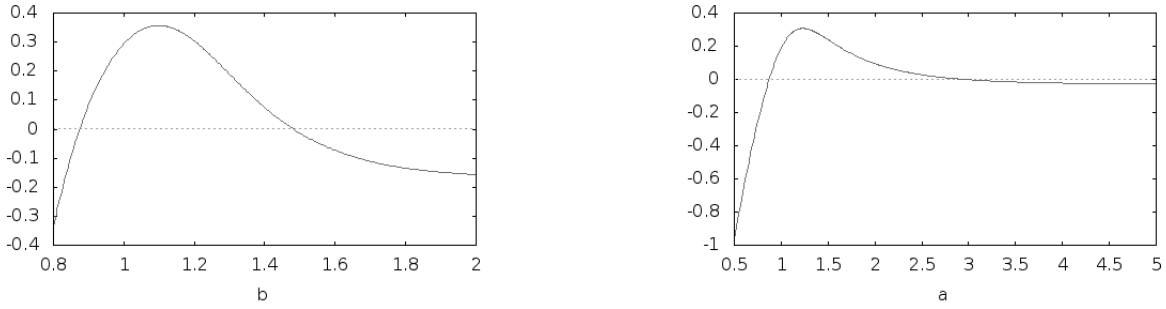
Figure A.1 – Soit $\theta_1 = (1; 1, 3)$ et nous regardons la divergence de Jeffrey entre θ_1 et $\bar{\theta} = (\bar{\alpha}; \bar{\beta})$. D'une part pour $\bar{\alpha} = 1$ (a) et d'autre part pour $\bar{\beta} = 1.3$ (b)

en les variables x et y. Cette surface est une cloche et elle est convexe. Appliquer ce même raisonnement à la divergence de Jeffrey, permet de montrer que un cadre simplifié ne suffit pas à avoir la convexité recherchée.

Nous allons regarder un cas simplifié de la fonction de coût. Nous supposons ici que le jeu de paramètres revient au seul vecteur $\theta_1 = (1; 1, 3)$. Nous regardons dans la figure A.1.(a) la valeur de la fonction de coût $l(\bar{\theta})$ dans le cas où le paramètre d'échelle est fixé $\bar{\alpha} = 1$. Nous constatons donc que pour $\bar{\beta} = 1, 3$ la fonction de coût s'annule. En effet, la divergence de Jeffrey s'annule si et seulement si les deux entrées sont égales, ce qui est bien le cas ici. Nous pouvons également regarder le comportement de la fonction de coût $l(\bar{\theta})$ dans le cas où le paramètre de forme est fixé $\bar{\beta} = 1, 3$, et nous constatons que la fonction de coût l s'annule toujours en $\bar{\theta} = \theta_1$.

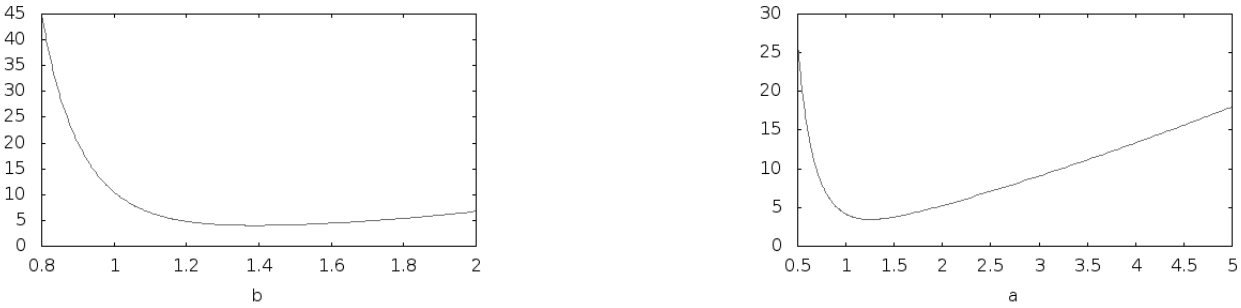
La variété complète, dont les figures A.1.(a) et (b) sont de simple coupes, possède une propriété de courbure. Sans rentrer dans les détails, c'est par l'étude des valeurs propres de la matrice hessienne que la convexité sera démontrée.

Pour ce cadre simplifié, nous pouvons évaluer numériquement les deux valeurs propres de la matrice hessienne $\mathcal{H}l(\bar{\theta})$ sans présenter d'expressions explicites. Une fois la valeur numérique à disposition, nous pouvons afficher les valeurs que peuvent prendre les valeurs propres, cette valeur dépend de la position de $\bar{\theta} = (\bar{\alpha}; \bar{\beta})$. Pour simplifier l'affichage, et avec la continuité des valeurs propres sur le domaine de définition, la surface ne sera représentée que par des coupes transversales. Premièrement, nous fixons le paramètre d'échelle $\bar{\alpha} = 1$ et faisons varier seulement le paramètre de forme $\hat{\beta}$. La continuité de la valeur propre garantie la pertinence des résultats pour un petit voisinage de $\bar{\alpha} = 1$. Dans un deuxième temps, c'est le paramètre de forme $\bar{\beta} = 1.3$ qui est fixé. Nous cherchons à montrer que la stricte convexité de la divergence de Jeffrey n'est garantie que localement, et les figures suivantes sont suffisantes pour le



(a) (b)

Figure A.2 – Soit $\theta_1 = (1; 1, 3)$ et nous regardons la première valeur propre de la hessienne de la divergence de Jeffrey entre θ_1 et $\bar{\theta} = (\bar{\alpha}; \bar{\beta})$. D'une part pour $\bar{\alpha} = 1$ (a) et d'autre part pour $\bar{\beta} = 1.3$ (b)



(a) (b)

Figure A.3 – Soit $\theta_1 = (1; 1, 3)$ et nous regardons la deuxième valeur propre de la hessienne de la divergence de Jeffrey entre θ_1 et $\bar{\theta} = (\bar{\alpha}; \bar{\beta})$. D'une part pour $\bar{\alpha} = 1$ (a) et d'autre part pour $\bar{\beta} = 1.3$ (b)

montrer.

Si nous regardons la première valeur propre de la matrice hessienne pour $\bar{\alpha} = 1$ avec la figure A.2.(a) ou pour $\bar{\beta} = 1, 3$ avec la figure A.2.(b), nous constatons que cette première valeur propre est strictement positive que dans un voisinage du minimum $\bar{\theta} = \theta_1$. La stricte positivité nous vient de l'équivalence locale de la divergence de Jeffrey avec la distance de Rao (III.25), qui est convexe. Le changement de signe indique que la fonction de coût l perd sa convexité lorsque l'on s'éloigne du barycentre $\bar{\theta} = \theta_1$.

Si nous regardons la seconde valeur propre de la matrice hessienne pour $\bar{\alpha} = 1$ avec la figure A.3.(a) ou pour $\bar{\beta} = 1.3$ avec la figure A.3.(b), nous constatons que cette deuxième valeur propre est strictement positive et elle pourrait n'être que strictement croissante en dehors. Entre les deux paramètres de la GGD, seul un paramètre nous apporte une instabilité suffisante pour perdre le caractère convexe de la fonction de coût l.

Le premier constat qui est fait reste la différence de comportement entre une divergence de Jeffrey et

ce qu'il est possible d'obtenir avec la distance euclidienne. La distance euclidienne est strictement convexe dans le cadre simple de la distance au barycentre θ_1 là ou la divergence pers la stricte convexité avec la distance au barycentre θ_1. Un autre test est mené avec un nombre d'échantillons plus grand, l'objectif est encore de surveiller la forme que prend la surface correspondant à la fonction de coût l.

La figure A.4 montre une généralisation avec un plus grand jeu de vecteurs de paramètres $(\theta_n)_{n=1}^N$. Sur la figure A.4.(a) (respectivement A.4.(b)) nous représentons les vecteurs $(\theta_n)_{n=1}^N$ par des cercles gris, ils sont répartis suivant une distribution gaussienne de moyenne $(1,5; 1,5)$ et de variance 0,5 (respectivement 0,1). Le barycentre calculé est représenté par la croix bleue. Maintenant nous fixons $\bar{\alpha} = 1,5$ et nous étudions l'évolution de la fonction de coût l dans la figure A.4.(c) (respectivement A.4.(d)), fonction de coût admettant alors un minimum en $\bar{\beta} = 1$ (respectivement $\bar{\beta} = 1,5$). Ce qui correspond à la valeur du barycentre $\bar{\beta}$, et montre que le paramètre d'échelle n'a que peu d'impact car seulement un rapport de paramètre d'échelle est en jeu ici.

Le changement de convexité est plus explicite dans la figure A.4.(d) que dans la figure A.4.(c) sans en être absent. Cela signifie que nous conservons la convexité que très localement autour du minimum de la fonction de coût l. Mais nous constatons cette forme de cuvette lisse indique que le barycentre devrait être unique (et donc exister), mais l'hypothèse de stricte convexité n'est pas utilisable pour le démontrer théoriquement. Pour la section suivante, le problème est vu différemment : il s'agit de trouver tous les minima et de les dénombrer.

3 Recherche gloutonne des minimums

Le minimum de la fonction de coût l correspond à l'estimateur du maximum de vraisemblance du barycentre $\bar{\theta}$. Nous avons numériquement que ce barycentre existe et que la configuration de la variété Θ se prêterai bien à l'unicité du minimum de la fonction de coût l. Nous ne pouvons pas conclure à la convexité de la fonction de coût (pour ce qui est de essayer un changement de variables, nous ignorons lequel choisir) et nous allons chercher à démontrer la proposition 6. Il s'agit de dénombrer tous les extrema et de montrer que ce sont des minima, dans ce cas la proposition nous assure l'unicité du barycentre.

Proposition 6. *Soit V un ouvert de Θ. Si $\forall \theta \in V$ tel que $\nabla l(\theta) = 0$ vérifie $\mathcal{H}l(\theta) \geqslant 0$, alors le minimum est unique.*

Démonstration. Raisonnons par contradiction, si nous disposons de deux minima disjoints, alors nous pouvons trouver un point col entre les cuvettes associées à chaque minimum. Le point col est un extremum

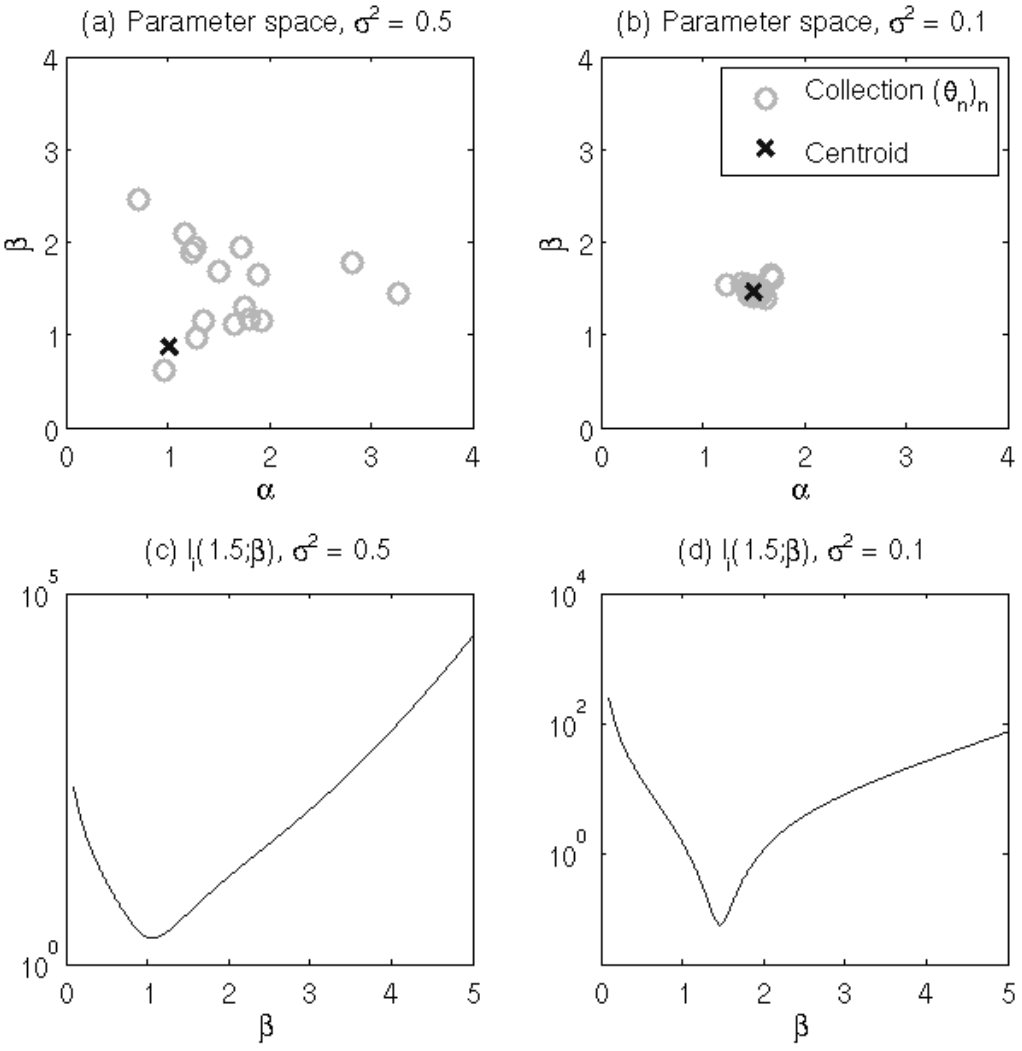


Figure A.4 – Soit un jeu de vecteurs de paramètres $(\theta_n)_{n=1}^N$ positionné par des cercles gris. Le barycentre est représenté par une croix noire. La variance σ^2 est cinq fois plus forte pour (a) et (c) que pour (b) et (d). Les graphiques (c) et (d) présentent l'évolution de la fonction coût $l(1.5;\hat{\beta})$.

$\nabla l(\theta) = 0$ mais il ne vérifiera pas $\mathcal{H}l(\theta) \geqslant 0$ sinon ce serai un minimum et non un point col. Nous avons donc montrer que si il y à deux minima, alors nous avons la présence d'autres extrema ne vérifiant pas les conditions. □

Notre objectif est donc de localiser tous les extrema de la fonction de coût l. Nous allons donc annuler chacune des dérivées partielles de l. Pour une vraisemblance $p(x \mid \theta)$ GGD, les dérivées partielles de l sont données par une combinaison linéaire des dérivées partielles de la divergence de Jeffrey données dans la table IV.1 page 112.

Nous allons travailler séparément chaque dérivée partielle et trouver les racines de chaque dérivée partielle. Les dérivées partielles sont des fonctions bivariées continues, cela signifie que s'il existe un vecteur θ_o qui annule une des dérivées partielles, alors dans un voisinage $V(\theta_o)$ la valeur de cette dérivée partielle est faible. Prenons $\theta_+ \in V(\theta_o)$ tel que $\partial l(\theta_+) > 0$, alors il existe un voisinage ouvert $V(\theta_+)$ de valeurs positives pour la dérivée partielle. De même pour $\theta_- \in V(\theta_o)$ tel que $\partial l(\theta_-) < 0$, alors il existe un voisinage ouvert $V(\theta_-)$ de valeurs négatives pour la dérivée partielle.

Maintenant que nous avons défini trois voisinages, par intersection il est possible de construire un nouvel ensemble dans lequel la dérivée partielle est faible sans être ni positive ni négative, autrement dit un ensemble de points annulant la dérivée partielle. L'ensemble $V_1(\theta_o) = V(\theta_o) \bigcap (\Theta \backslash V(\theta_+)) \bigcap (\Theta \backslash V(\theta_-))$ montre par récurrence que nous pouvons contraindre le voisinage de θ_o jusqu'à obtenir une courbe de Θ sur laquelle s'annule la dérivée partielle. Donc si une dérivée partielle s'annule, il existe une hyper-courbe de vecteurs de paramètres annulant cette dérivée partielle.

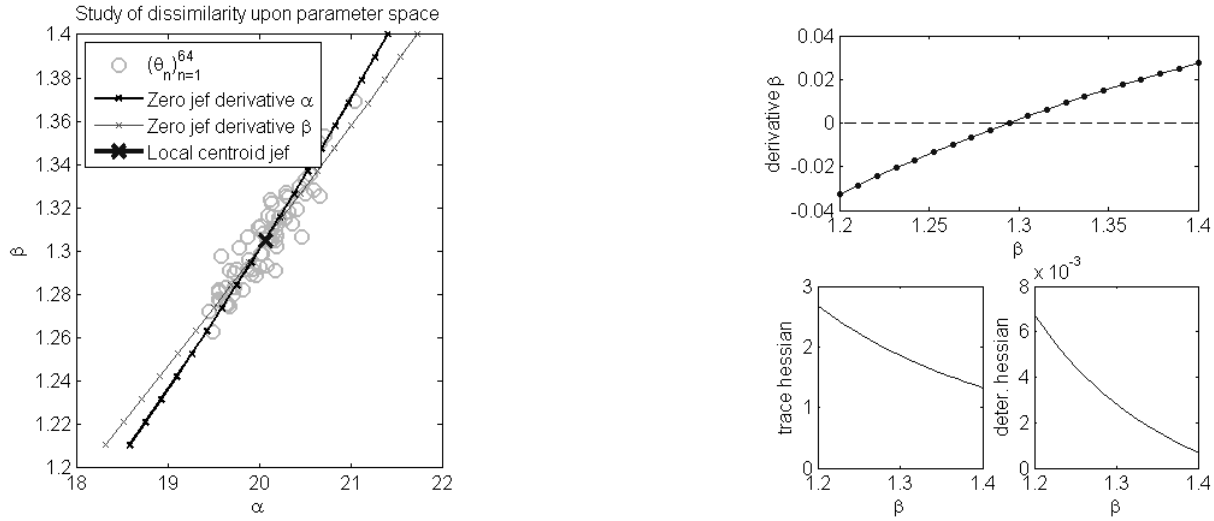
Considérons la droite de Θ tel que le paramètre de forme équivaut à $\beta = \beta_0$. Pour chaque vecteur de paramètres de la droite sur laquelle la dérivée partielle (en α) s'annule, il existe une courbe de Θ qui annule la dérivée partielle. Explicitons la dérivée partielle en α de la fonction coût :

$$\frac{\partial l(\alpha;\beta_0)}{\partial \alpha} = \sum_{n=1}^{N} w_n \frac{1}{\alpha} \left(\beta_n A_L + \beta_0 A_R\right)$$

que nous dérivons en fonction de α :

$$\frac{\partial^2 l(\alpha;\beta_0)}{\partial \alpha^2} = \sum_{n=1}^{N} w_n \frac{1}{\alpha^2} \left(\beta_n(\beta_n - 1) \left(\frac{\alpha}{\alpha_n}\right)^{\beta_n} \frac{\Gamma((\beta_n+1)/\beta_0)}{\Gamma(1/\beta_0)} + (\beta_0^2 + \beta_0) \left(\frac{\alpha_n}{\alpha}\right)^{\beta_0} \frac{\Gamma((\beta_0+1)/\beta_n)}{\Gamma(1/\beta_n)} \right)$$

La dérivée d'ordre 2 est donc strictement positive si l'ensemble des β_n est strictement supérieur à 1. Attention, $\beta_n > 1$ n'est pas une condition nécessaire pour avoir la stricte positivité de la dérivée d'ordre



(a) (b)

Figure A.5 – (a) Dans l'espace Θ nous représentons les vecteurs de paramètres $(\theta_n)_{n=1}^N$ par des cercles gris et le barycentre est représenté par une grosse croix noire. La courbe munie de petites croix noires représente une courbe où s'annule la dérivée partielle en α $\partial_\alpha l(\theta)$ alors que la courbe munie de petites croix grises représente un courbe où s'annule la dérivée partielle en β $\partial_\beta l(\theta)$. (b) en haut, la dérivée partielle en β $\partial_\beta l(\theta)$ calculée le long de la courbe $\partial_\alpha l(\theta) = 0$. (b) en bas la trace et le déterminant de la matrice hessienne montrant la stricte positivité des valeurs propres de la matrice hessienne $\mathcal{H}l(\theta)$

2, mais nous nous en contenterons pour le moment. Cela signifie que si les vecteurs de paramètres $(\theta_n)_{n=1}^N$ vérifient $\beta_n > 0$ alors la fonction de coût $l(\theta)$ est strictement convexe sur la droite $\beta = \beta_0$.

Nous avons alors l'existence et l'unicité d'un minimum pour la fonction de coût tel que $\beta = \beta_0$. Cette valeur annule également la dérivée partielle en α d'ordre 1. Numériquement, nous pouvons chercher cet unique minimum et l'afficher dans l'espace Θ. En faisant varier la valeur de β, il est possible de démontrer qu'il existe une seul et unique paire (α, β) qui annule la dérivée partielle en α. Nous proposons alors de tracer la courbe, qui existe, constituée de tous les points annulant la dérivée partielle en α.

La figure A.5.(a) représente dans l'espace Θ le jeu de paramètres $(\theta_n)_{n=1}^N$ par des cercles gris et le barycentre $\bar{\theta}$ par une grande croix noire. Deux courbes sont tracées, la courbe noire et épaisse présente la courbe annulant la dérivée partielle $\frac{\partial l(\theta)}{\partial \alpha}$ alors que la courbe fine grise représente une suite de points annulant la dérivée partielle $\frac{\partial l(\theta)}{\partial \beta}$. La première courbe existe et est unique puisque l'ensemble des paramètres de forme $\beta_n > 1$ mais la deuxième courbe n'est pas théoriquement unique. Le travail effectué pour la dérivée partielle en α n'est pas généralisable simplement à la dérivée partielle en β et les détails des calculs ne seront pas énoncés pour aller à l'essentiel. Comme nous pourrions nous y attendre, le barycentre $\bar{\theta}$ est élément des courbes annulant chacune des dérivées et fait partie des croisements des deux courbes.

Nous étudions alors la dérivée partielle $\frac{\partial l(\theta)}{\partial \beta}$ que sur l'ensemble des θ tel que $\frac{\partial l(\theta)}{\partial \alpha} = 0$. Nous cherchons

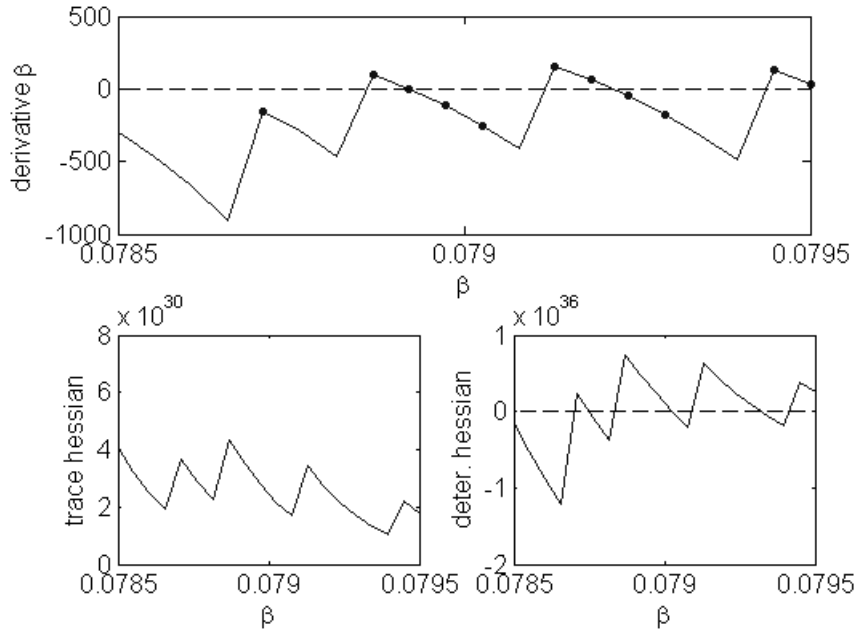


Figure A.6 – En haut, la dérivée partielle en β $\partial_\beta l(\theta)$ le long de la courbe $\partial_\alpha l(\theta) = 0$ pour de faibles valeurs de β. En bas, le déterminant et la trace de la matrice informent sur le signe des deux valeurs propres.

tous les θ annulant la deuxième dérivée partielle $\frac{\partial l(\theta)}{\partial \beta}$ pour avoir tous les minima. Dans un voisinage du nuage de points $(\theta_n)_{n=1}^N$, nous constatons que la dérivée partielle en β ne s'annule qu'une fois, et il semblerait que tout fonctionne comme nous le souhaitions. Sur la figure A.6 l'étude de la courbe est portée jusqu'à la limite $\beta \to 0$. Nous constatons que la valeur de la dérivée $\frac{\partial l(\theta)}{\partial \beta}$ n'est plus stable et de nombreux sauts sont effectués entre valeurs positives et valeur négatives. L'étude du déterminant et de la trace de la matrice hessienne $\mathcal{H}l(\theta)$ de la fonction de coût montre que les deux valeurs propres sont positives lorsque la dérivée en β s'annule montrant, de ce fait, un minimum. Néanmoins la grande instabilité numérique pousse à éviter les cas limites et nous choisirons des valeurs de β strictement supérieure à 0,1 pour assurer une stabilité de l'algorithme.

Cette section a trouvé l'ensemble des minima pour les dérivées partielles et leurs points communs se résument en le barycentre qui est trouvé par la méthode de descente de gradient. Aussi le caractère lisse de la fonction de coût l remet en question les résultats obtenus à la limite ou β est presque nul. L'unicité de la courbe annulant la dérivée partielle en β permettrait de valider la démonstration expérimentale, mais elle n'est pas disponible. Le fait de présenter un barycentre extérieur au nuage de points initial peut sembler illogique. Aussi pour éclaircir la véracité de ce minimum qui est extérieur, la section suivante

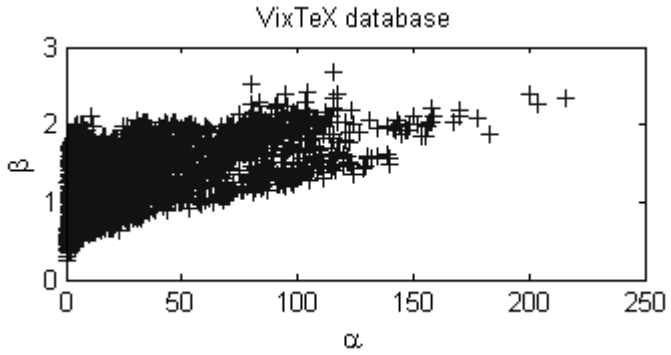


Figure A.7 – Base de données d'images texturée VisTex, paramètres estimés suivant la vraisemblance $p(x \mid \theta)$ GGD

présente le cas de l'inégalité triangulaire.

4 Inégalité triangulaire faible

Nous constatons que notre recherche de l'ensemble des minima ne peut pas être faite sur tous l'espace Θ et qu'il est alors nécessaire de réduire l'espace de recherche. L'espace peut être réduit suivant la connaissance *a priori* sur les données que nous classifions, peut-être que l'espace des vecteurs de paramètres observés est borné sur l'espace Θ. La figure A.7 présente, pour une application précise, les valeurs observées des vecteurs de paramètres θ. Utiliser une approche en bornant α et β séparément est possible mais les inégalités vérifiées sont trop larges pour pouvoir déduire la stricte positivité de la matrice hessienne $\mathcal{H}l(\theta)$. Il serait judicieux de construire une fonction bornant mieux les valeurs jugées possibles *a priori*.

Nous allons supposer que le barycentre n'est pas éloigné des vecteurs de paramètres $(\theta_n)_{n=1}^N$, mais la notion de distance est reliée, sur l'espace des paramètres Θ, à la géométrie de l'espace $G(\theta)$ et donc à la divergence de Jeffrey choisie. La proposition 7 formalise mathématiquement l'idée.

Proposition 7. *Soient $(\theta_n)_{n=1}^N \in \Theta$ un jeu de vecteurs de paramètres et l une fonction de coût déduite de ce jeu de paramètres. Soit R la divergence maximum existante entre deux vecteurs de paramètres :*

$$R = \max_{n,n'=1,\dots,N} J(\theta_n, \theta_{n'}),$$

n_0 l'index du vecteur minimisant la fonction coût :

$$\theta_{n_0} = arg\min_{n=1,\dots,N} l(\theta_n).$$

S'il existe ϵ un réel strictement positif tel que :

$$\epsilon J(\theta_1, \theta_3) \leqslant J(\theta_1, \theta_2) + J(\theta_2, \theta_3)$$

alors le barycentre $\bar{\theta}$ se trouve localement proche du vecteur θ_{n_0} minimisant la fonction de coût :

$$\bar{\theta} \in \left\{ \theta | J(\theta, \theta_n) \leqslant \frac{2R}{\epsilon} \right\}.$$

Démonstration. Nous raisonnerons par contradiction, supposons que θ minimise la fonction de coût l et vérifie :

$$J(\theta, \theta_{n_0}) > \frac{2R}{\epsilon}$$

Nous allons utiliser l'inégalité triangulaire faible :

$$\epsilon J(\theta, \theta_{n_0}) \leqslant J(\theta, \theta_n) + J(\theta_n, \theta_{n_0})$$

pour minorer la valeur de la fonction de coût l :

$$\begin{aligned} l &\geqslant \sum_{n=1}^{N} w_n \left[\epsilon J(\theta, \theta_{n_0}) - J(\theta_n, \theta_{n_0}) \right] \\ &= \epsilon J(\theta, \theta_{n_0}) - \sum_{n=1}^{N} w_n J(\theta_n, \theta_{n_0}) \\ &> \epsilon \frac{2R}{\epsilon} - \sum_{n=1}^{N} w_n R = R \end{aligned}$$

Maintenant si nous évaluons la fonction de coût l en θ_{n_0} :

$$l(\theta_{n_0}) = \sum_{n=1}^{N} w_n J(\theta, \theta_n) \leqslant R < l(\theta)$$

Nous pouvons donc trouver un vecteur de paramètres θ' sur lequel la fonction de coût est inférieure à la valeur en θ, θ ne peut donc pas être le minimum de la fonction de coût l. □

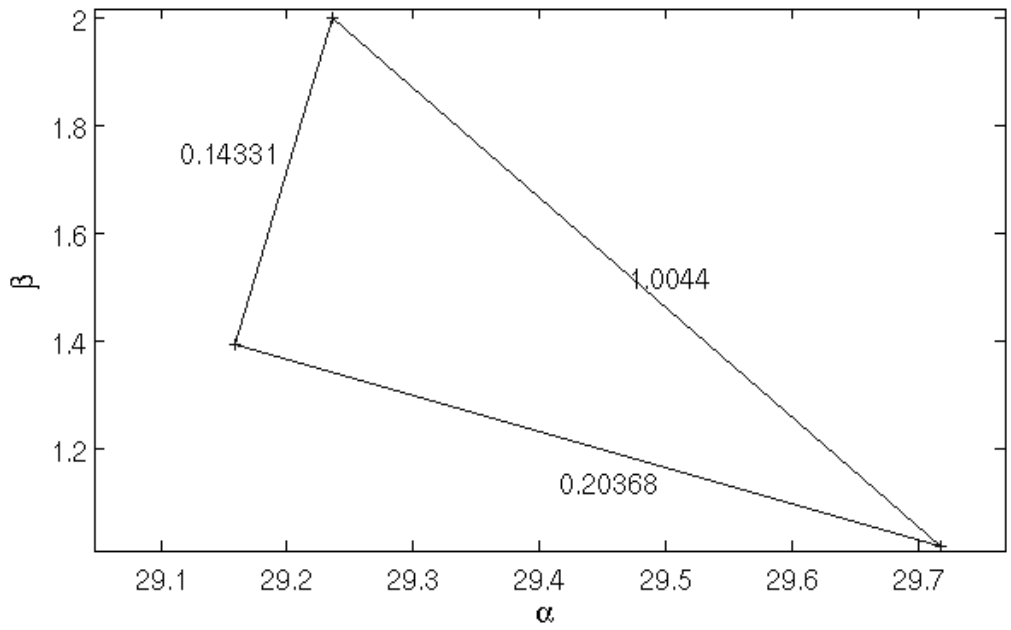


Figure A.8 – Trois vecteurs de paramètres vérifiant $\alpha \in [29; 30]$ et $\beta \in [1; 2]$ et la divergence de Jeffrey calculée entre chaque couple possible des vecteurs.

La proposition 7 repose sur l'inégalité triangulaire « faible ». En effet, une divergence ne satisfait pas à deux propriétés de la mesure que sont la symétrie et l'inégalité triangulaire. La divergence de Jeffrey corrige la divergence de Kullback-Leibler sur la symétrie mais l'inégalité triangulaire n'existe pas. L'inexistence de l'inégalité triangulaire indique que pour θ_1 et θ_2 deux vecteurs de paramètres sur lesquels nous pouvons calculer la divergence de Jeffrey il peut exister un vecteur de paramètre θ_3 tel que la somme de J entre θ_1 et θ_3 et de J entre θ_3 et θ_2 soit inférieure à la divergence entre θ_1 et θ_2.

La divergence est localement équivalente à une distance, donc localement elle vérifie l'inégalité triangulaire. Cela signifie que si le troisième vecteur existe, il n'est donc pas local au sens euclidien à l'un des deux vecteurs. La figure A.8 présente trois vecteurs de paramètres, entre chaque couple de vecteurs de paramètres, nous calculons la divergence de Jeffrey pour constater que deux vecteurs sont trop dissimilaires pour vérifier l'inégalité triangulaire. Afin d'appliquer la proposition 7, la divergence devrait être multipliée par $\epsilon = 1/2,8945 = 3,4548 \times 10^{-1}$.

Nous pouvons montrer que l'inégalité triangulaire faible existe dans un ensemble borné de l'espace de paramètres Θ, de ce fait, nous pouvons vérifier la proposition 7 au sein de cet ensemble borné. Mais le soucis relevé précédemment sur l'élaboration de cet espace ne rend pas possible de minorer les valeurs propres de la matrice hessienne $\mathcal{H}l$ de la fonction de coût. La figure A.9 montre que les divergences

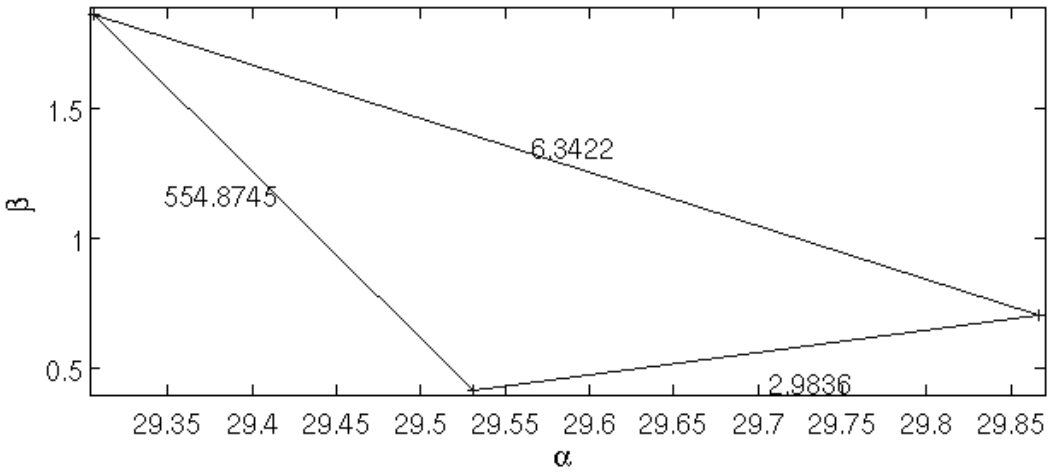


Figure A.9 – Trois vecteurs de paramètres vérifiant $\alpha \in [29; 30]$ et $\beta \in [0.4; 2]$ et la divergence de Jeffrey calculée entre chaque couple possible des vecteurs.

calculées explosent exponentiellement par rapport à l'élargissement de l'ensemble borné. Nous pouvions estimer $\epsilon = 3,4548 \times 10^{-1}$ pour des valeurs de $\beta > 1$ mais cette valeur atteint $\epsilon = 1,6807 \times 10^{-2}$ lorsque $\beta > 0,4$. Autrement dit, l'inégalité triangulaire faible disparaît rapidement pour un espace borné un peu trop large.

Dans un cadre opérationnel, l'inégalité triangulaire faible doit pouvoir être vérifiée sur, au minimum, l'espace décrit par les vecteurs paramétriques estimés. Dans le paragraphe précédent, nous comparions les valeurs de ϵ entre deux régions plus ou moins large qui ne sont qu'une partie de l'espace total. Il n'est pas possible d'établir cette inégalité triangulaire faible sur tous l'espace. Par conséquent la proposition 7 ne peut être utilisée.

Dans la section suivante est étudiée la convergence sur l'enveloppe convexe autour du jeu de vecteurs de paramètres $(\theta_n)_{n=1}^N$. L'idée est alors de démontrer que même cette étude très locale ne peut garantir la convexité et que ce concept de convexité ne peut être utilisé pour prouver l'unicité du barycentre.

5 Convexité locale seulement si le nuage de vecteurs de paramètres est dense

Notre dernière étude sera plus locale sur la variété que précédemment, nous allons nous contraindre à l'enveloppe convexe (au sens de la distance euclidienne) des vecteurs de paramètres $(\theta_n)_{n=1}^N$. Nous

cherchons à montrer la stricte convexité de la fonction de coût au sein de l'enveloppe convexe, pour ce faire, nous tirons aléatoirement et uniformément des vecteurs de paramètres dans l'enveloppe convexe.

Première étape, l'enveloppe convexe de trois vecteurs de paramètres $(\theta_n)_{n=1}^3$ au sens de la distance euclidienne. Une ensemble est dit convexe si pour tout couple θ, θ' de l'ensemble convexe, le segment reliant θ à θ' est lui aussi compris dans l'ensemble convexe. Par conséquent l'enveloppe convexe de trois points, qui est l'espace convexe minimal contenant les trois points, est le triangle plein. Nous allons maintenant paramétrer notre triangle au moyen de u_1 et $u_2 \in [0;1]$. Alors tout vecteur θ tel que

$$\theta = u_1\theta_1 + u_2(1-u_1)\theta_2 + (1-u2)(1-u1)\theta_3 \tag{A.1}$$

appartient au triangle $\theta_1\hat{\theta_2}\theta_3$. Pour générer aléatoirement et uniformément notre vecteur θ dans le triangle, nous allons générer uniformément u_1 et u_2 dans le cube $[0;1]^2$. Puis nous construisons θ au moyen de l'équation A.1.

L'extension pour N vecteurs de paramètres ne se fera pas avec l'hypercube $[0;1]^N$, sinon la définition aléatoire de θ présentera un produit de plus de 4 réels inférieurs à 1 et, numériquement, seront des valeurs négligeables. L'ensemble de l'enveloppe convexe ne peut être couverte, nous allons alors reprendre le concept développé avec trois vecteurs de paramètres que nous choisirons aléatoirement et sans répétition parmi les N vecteurs de paramètres. Nous modelons alors l'enveloppe convexe comme l'union de tous les triangles possibles.

La figure A.10 présente deux exemples d'enveloppes convexes, à gauche la variance du jeu de vecteurs de paramètres $(\theta_n)_{n=1}^N$ est de 0,2 alors qu'il n'est que de 0,05 à droite. Nous comparons donc un comportement local sur la droite à un comportement non local à gauche. Nous avons donc généré un vecteur de paramètre θ uniformément dans l'enveloppe convexe. Pour chaque θ généré, nous le représentons par un carré. Les carrés sont colorés en noir ou rouge. Si la fonction de coût l est strictement convexe en θ, le carré sera noir, sinon il sera rouge. Nous rappelons que la stricte convexité de la fonction de coût l est obtenue avec la stricte positivité des deux valeurs propres de la matrice hessienne $\mathcal{H}l$. Une condition suffisante pour leur stricte positivité, est la positivité du déterminant et de la trace de la matrice hessienne $\mathcal{H}l$. Parmi l'ensemble des valeurs calculées, le déterminant minimum et la trace minimum sont affichées au dessus de chaque figure.

La figure A.10 à droite montre que la fonction de coût reste strictement convexe sur toute l'enveloppe convexe du nuage $(\theta_n)_{n=1}^N$ ce qui correspond au comportement que nous sommes en droit d'attendre. La figure A.10 à gauche présente deux couleurs distinctes, le noir et le rouge. Nous ne conservons pas la

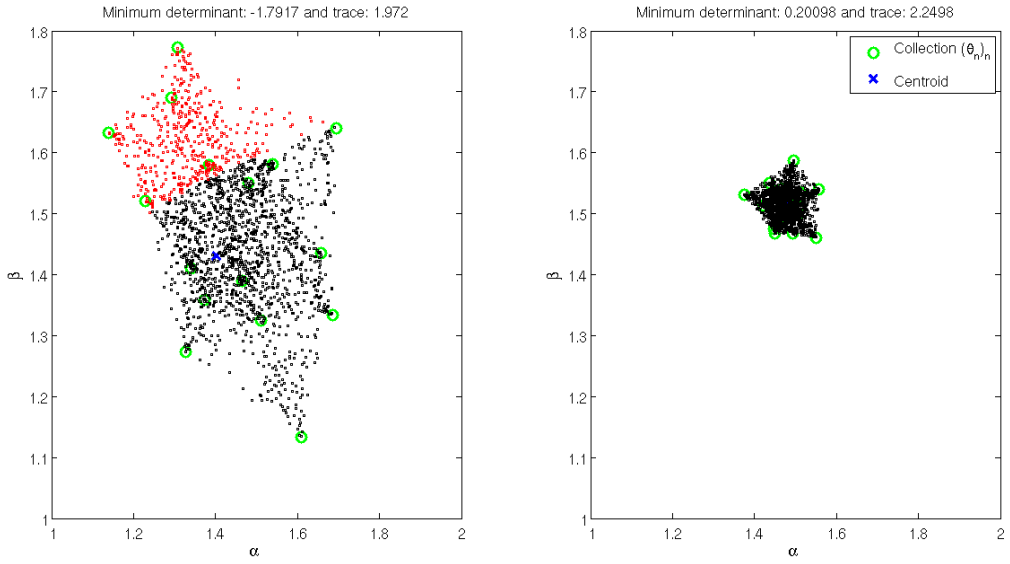


Figure A.10 – Le jeu de vecteurs de paramètres $(\theta_n)_{n=1}^N$ représenté par des cercles verts dispose d'une variance 4 fois plus forte à gauche qu'à droite. Le barycentre est représenté par une croix bleue. Chaque θ choisi aléatoirement dans l'enveloppe convexe est représenté par un carré coloré, noir si l est convexe en θ et rouge sinon.

stricte convexité sur toute l'enveloppe convexe. Attention toutefois, la perte de la convexité n'implique pas forcément la concavité et nous ne pouvons pas déduire plus de propriétés de cette perte de convexité.

6 Conclusion

Notre recherche précédente consistait à réduire l'espace et montrer le caractère strictement convexe. Nous venons de montrer par une réalisation numérique que nous n'avons pas la stricte convexité au sein de l'enveloppe convexe au sens de la distance euclidienne. Sans la stricte convexité, il nous reste à montrer la proposition 6 qui nécessite de trouver l'ensemble de tous les points annulant le gradient ∇l et de montrer que ce sont des minima. Cette recherche a montré l'unicité du minimum si nous omettons le cas limite ou le paramètre de forme $\beta \to 0$ amenant une certaine instabilité dans le calcul de la fonction de coût et de ses dérivées partielles.

Numériquement la méthode du recuit simulé nous a permis de montrer l'existence d'un minimum global et l'absence de minimum local a été montrée par l'égalité entre le barycentre obtenu par descente de gradient et le barycentre obtenu avec le recuit simulé. L'instabilité de la descente de gradient s'explique par l'instabilité de la fonction de coût l et de ses dérivées lorsque le paramètre de forme $\beta \to 0$.

Théoriquement nous n'avons ni l'existence ni l'unicité du barycentre. Comme l'expression explicite de

la fonction de coût l associée à la loi *a priori* $p(\theta)$ construite autour de la vraisemblance $p(x \mid \theta)$ GGD n'est pas lipshitzienne, nous ne pouvons pas déduire par réalisations numériques ni que la surface associée à l soit lisse ni que le minimum existe et est unique. En effet, rien n'empêche la présence d'oscillations très haute fréquence provoquant une infinité de minima pour la fonction de coût l.

Bibliographie

[1] Antonio Fernández, Marcos X Álvarez, and Francesco Bianconi. Texture description through histograms of equivalent patterns. *Journal of mathematical imaging and vision*, 45(1) :76–102, 2013. ix, 8, 10, 22

[2] Gertjan J. Burghouts and Jan-Mark Geusebroek. Material-specific adaptation of color invariant features. *Pattern Recognition Letters*, 30(3) :306–313, 2009. ix, 14, 15, 21, 49

[3] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering : a review. *ACM Comput. Surv.*, 31 :264–323, Sep. 1999. xi, 57

[4] M.N. Do and M. Vetterli. Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance. *Image Processing, IEEE Transactions on*, 11(2) :146 –158, feb 2002. xvii, 27, 28, 30, 31, 32, 33, 35, 37, 38, 89, 111, 128, 130, 158

[5] Josef Sivic and Zisserman andrew. Video google : A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003. 4

[6] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 4

[7] R.M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-3(6) :610–621, 1973. 9, 128, 141

[8] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1) :51 – 59, 1996. 9

[9] B. Julesz. Visual pattern discrimination. *Information Theory, IRE Transactions on*, 8(2) :84–92, February 1962. 9, 144

[10] M. Varma and A. Zisserman. A statistical approach to material classification using image patch exemplars. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(11) :2032–2047, Nov 2009. 12, 14, 128, 129, 132, 154, 156, 157, 159, 161, 162, 165, 167

[11] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62 :61–81, 2005. 10.1023/B :VISI.0000046589.39864.ee. 13, 14, 21, 24, 156

[12] Li Liu, Paul Fieguth, David Clausi, and Gangyao Kuang. Sorted random projections for robust rotation-invariant texture classification. *Pattern Recognition*, 45(6) :2405–2418, 2012. 13, 131, 132

[13] Steven W. Zucker and Demetri Terzopoulos. Finding structure in co-occurrence matrices for texture analysis. *Computer Graphics and Image Processing*, 12(3) :286 – 308, 1980. 14, 141

[14] Pierre Chainais. Towards dictionary learning from images with non gaussian noise. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2012. 16

[15] Mingyuan Zhou, Haojun Chen, John Paisley, Lu Ren, Lingbo Li, Zhengming Xing, David Dunson, Guillermo Spiro, and Lawrence Carin. Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images. *IEEE Transactions on Image Processing*, 21 :130–144, Jan. 2012. 16

[16] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 17

[17] Honglak Lee, Alexis Battle, Rajat Raina, and Ng andrew Y. Efficient sparse coding algorithms. *NIPS*, 2007. 17

[18] J.A. Tropp and S.J. Wright. Computational methods for sparse solution of linear inverse problems. In *Procedings of IEEE*, volume 98, pages 948–958, 2010. 19

[19] I.F. Gorodnitsky and B. D. Rao. A new iterative weighted noiserm minimization algorithm and its applications. In *Workshop on Statistical Signal and Array Processing*, number 6 in 1, pages 412–415, 1992. 19

[20] Aharon and al. K-svd : An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 2006. 20

[21] Jin Xie, D. Zhang, and J. You. Texture classification via patch-based sparse texton learning. In *Image Processing (ICIP), International Conference on*, pages 2737–2740, Sept. 26-29 2010.

[22] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discrminative learned dictionaries for local image analysis. In *Computer Vision and Pattern and Vision (CCVPV), IEEE Conference on*, Anchorage, Alaska, USA, 2008.

[23] G. Peyré. Sparse modelling of textures. *Journal of Mathematical Analysis and Applications*, 34(1) :17–31, 2009. ISSN 0924-9907. 20

[24] K. Engan, S. O. Aase, and J. H. Husoy. Frame based signal compression using method of optimal directions (mod). In *Circuits and Systems. Proceedings of the IEEE International Symposium on*, volume 4, pages 1–4, Jul 1999. 20

[25] Lingqiao Liu, Lei Wang, and Xinwang Liu. In defense of soft-assignment coding. *ICCV*, pages 2486–2493, 2011. 23

[26] Michael Unser. Local linear transforms for texture measurements. *Signal Processing*, 11(1) :61 – 79, 1986. 27, 28, 29, 32, 88, 131

[27] M. R. Turner. Texture discrimination by gabor functions. *Biological cybernetics*, 55(2-3) :71–82, 1986. PMID : 3801538. 27

[28] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biological cybernetics*, 61(2) :103–113, 1989. 27

[29] D. Dunn, W. Higgins, and J. Wakeley. Texture segmentation using 2-d gabor eleelement funcitions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1994. 27

[30] J. S. De Bonnet. Multiresolution sampling processing for analysis and synthesis of texture images. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques, SIGGRAPH '97*, pages 361–368, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. 27

[31] J. Portilla and E.P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1) :49–70, Jun. 9 2000. 27, 52

[32] G. Tzagkarakis, B. Beferull-Lozano, and P. Tsakalides. Rotation-invariant texture retrieval via signature alignment based on steerable sub-gaussian modeling. *Image Processing, IEEE Transactions on*, 17(7) :1212–1225, july 2008. 27, 130

[33] S.G. Mallat. A theory for multiresolution signal decomposition : The wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7) :674–693, 1989. 27, 32

[34] Michael Unser and M. Eden. Multiresolution feature extraction and selection for texture segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7) :717–728, Jul 1989. 28

[35] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1) :106, 1962. 28

[36] John G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20(10) :847 – 856, 1980.

[37] B. Julesz and J.R. Bergen. Human factors and behavioral science : Textons, the fundamental elements in preattentive vision and perception of textures. *Bell System Technical Journal, The*, 62(6) :1619–1645, July 1983.

[38] Jacob Beck, Anne Sutter, and Richard Ivry. Spatial frequency channels and perceptual grouping in texture segregation. *Computer Vision, Graphics and Image Processing*, 37(2) :299 – 325, 1987. 28

[39] O Faugeras. Texture analysis and classification using a human visual model. In *Proc. 3rd International Conference on Pattern Recognition, Tokyo, Japan*, pages 549–552, 1978. 28

[40] G. H. Granlund. Description of texture using the generalogeneral approach. In *Proceddings of the 5th International Conference on Pattern Recognition*, pages 776–779, 1980.

[41] Diederich Wermser and Claus E. Liedtke. Texture analysis using a model of the visual system. In *Proceedings of 6th International Conference on Pattern Recognition*, pages 1070–1080, 1982. 28

[42] S. Watanabe. Karhunen-loève expansion and factor analysis. In *Trans. 4th Prague Conf. in Information Theory*, Czechoslovak Academy of Sciences, Prague, 1965. 29

[43] M. Unser. On the approximation of the karhunen-loève transform of stationary processes. *Signal Processing*, 7(3) :231–249, December 1984. 29

[44] S.-K. Choy and C.-S. Tong. Supervised texture classification using characteristic generalized gaussian density. *Journal of Mathematical Imaging and Vision*, 29 :35–47, Aug. 31 2007. 30, 32, 33, 35, 41, 83, 89, 111, 119, 128, 131, 143, 144

[45] Y. Stitou, N. Lasmar, and Y. Berthoumieu. Copulas based multivariate gamma modeling for texture classification. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1045–1048, april 2009. 32, 33, 130

[46] S.K. Choy and C.S. Tong. Statistical wavelet subband characterization based on generalized gamma density and its application in texture retrieval. *Image Processing, IEEE Transactions on*, 19(2) :281 –289, feb. 2010. 30, 31, 32, 33, 35, 38, 83, 131

[47] M.S. Allili, N. Bouguila, and D. Ziou. Finite generalized gaussian mixture modeling and applications to image and video foreground segmentation. In *Computer and Robot Vision, 2007. CRV '07. Fourth Canadian Conference on*, pages 183 –190, may 2007. 30, 32, 74

[48] Nour-Eddine Lasmar. *Modélisation stochastique pour l'analyse d'images texturées : Approches Bayésiennes pour la caractérisation dans le domaine des transformées.* PhD thesis, Université Bordeaux 1, école doctorale des sciences physiques et de l'ingénieur, Apr. 8 2013. 31, 32, 33, 38, 47, 83, 128, 135

[49] J. Rabin, G. Peyré, J. Delon, and M. Bernot. Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision (SSMV), International Conference on*, volume 8, 2011. 32

[50] M.N. Do and M. Vetterli. Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden markov models. *Multimedia, IEEE Transactions on*, 4(4) :517 – 527, dec 2002. 32, 33, 130

[51] Geert Verdoolaege, S. De Backer, and P. Scheunders. Multiscale colour texture retrieval using the geodesic distance between multivariate generalized gaussian models. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 169–172, 2008. 32

[52] G. Verdoolaege, Y. Rosseel, M. Lambrechts, and P. Scheunders. Wavelet-based colour texture retrieval using the kullback-leibler divergence between bivariate generalized gaussian models. In *Image Processing (ICIP), 2009 16th IEEE International Conferenceon*, pages 265 –268, Nov. 2009. 130

[53] L. Bombrun, Y. Berthoumieu, N.-E. Lasmar, and Geert Verdoolaege. Multivariate texture retrieval using the geodesic distance between elliptically distributed random variables. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 3637–3640, Sept 2011. 32

[54] R. Kwitt, P. Meerwald, and A. Uhl. Efficient texture image retrieval using copulas in a bayesian framework. *Image Processing, IEEE Transactions on*, 20(7) :2063–2077, july 2011. 32, 130

[55] N.-E. Lasmar and Y. Berthoumieu. Multivariate statistical modeling for texture analysis using wavelet transforms. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 790–793, March 2010. 32, 41, 42, 47

[56] L. Bombrun, N.-E. Lasmar, Y. Berthoumieu, and G. Verdoolaege. Multivariate texture retrieval using the sirv representation and the geodesic distance. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 865 –868, may 2011. 32, 33, 113

[57] M. Allili. Wavelet modelling using finite mixtures of generalized gaussian distributions : Application to texture discrimination and retrieval. *Image Processing, IEEE Transactions on*, 21(99) :1452 – 1464, april 2012. 35, 74, 83, 128

[58] M. K. Varanasi. Parameter estimation of the generalized gaussian noise model. Master's thesis, Department of Electrical and Computer Engineering, Rice University, Houston, TX, 1986. 37

[59] S. D. Silvey. *Statistical inference, Monographs on Applied Probability and Statistics.* Wiley, New York, 1975. 37

[60] S. M. Zabin and H. V. Poor. Parameter estimation for middleton class a interference channels. *IEEE Trans. Commun.*, 10, 1989.

[61] M. K. Varanasi and B. Aazhang. Parametric generalized gaussian density estimation. *The Journal of the Acoustical Society of America*, 86(4) :1404–1415, 1989. 37

[62] Olivier Schwander, Aurelien J Schutz, Frank Nielsen, and Yannick Berthoumieu. k-mle for mixtures of generalized gaussians. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2825–2828. IEEE, 2012. 38, 89, 167

[63] Aurélien Schutz, Yannick Berthoumieu, Flavius Turcu, Corina Nafornita, and Alexandru Isar. Barycentric distribution estimation for texture clustering based on information-geometry tools. In *Electronics and Telecommunications (ISETC), 2012 10th International Symposium on*, pages 343–346. IEEE, 2012. 63, 111, 113

[64] Aurélien Schutz, Lionel Bombrun, and Yannick Berthoumieu. Centroid-based texture classification using the sirv representation. *IEEE International Conference on Image Processing*, pages 3810–3814, 2013. 48, 117

[65] A. Schutz, L. Bombrun, and Y. Berthoumieu. K-centroids-based supervised classification of texture images : Handling the intra-class diversity. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 1498–1502, 2013. 56, 113, 145

[66] Aurélien Schutz, Lionel Bombrun, Yannick Berthoumieu, and Mohamed Najim. Centroid-based texture classification using the generalized gamma distribution. In *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, pages 1–5. IEEE, 2013. 41, 113

[67] Aurélien Schutz, Lionel Bombrun, Yannick Berthoumieu, et al. Classification d'images texturées basée sur k-barycentres : meilleure gestion de la diversité intra-classe. *Groupe d'Etudes du Traitement du Signal et des Images (GRETSI)*, pages 1–4, 2013. 38, 56, 63, 145, 167

[68] Kai-Sheng Song. Globally convergent algorithms for estimating generalized gamma distributions in fast signal and image processing. *Image Processing, IEEE Transactions on*, 17(8) :1233–1250, Aug 2008. 39, 40

[69] K. Fang, S. Kots, and K. Ng. *Symmetric Multivariate and Related Distributions*. London, U.K. : Chapmann & Hall, 1990. 44

[70] S. Zozor and C. Vignat. Some results on the denoising problem in the elliptically distributed context. *IEEE Trans. Signal Process.*, 58(1) :134–150, Jan. 2010. 44

[71] F. Chitour, Y.and Pascal. Exact maximum likelihood estimates for sirv covariance matrix : existence and algorithm analysis. *IEEE Trans. Signal Process.*, 56(10) :4563–4573, Oct. 2008. 44

[72] F. Gini and M. V. Greco. Covariance matrix estimation for CFAR detection in correlated heavy tailed clutter. *Signal Processing*, 82(12) :1847–1859, 2002. 45

[73] F. Pascal, Y. Chitour, J. Ovarlez, P. Forster, and P. Larzabal. Covariance structure maximum-likelihood estimates in compound gaussian noise : Existence and algorithm analysis. *Signal Processing, IEEE Transactions on*, 56(1) :34–48, Jan 2008. 45

[74] K. Krishnamoorthy. *Handbook of Statistical Distributions with Applications*. Chapman & Hall, Tailor & Francis Group, LLC, 2006. 45

[75] Aurélien Schutz, Lionel Bombrun, and Yannick Berthoumieu. K-centroids-based supervised classification of texture images using the sirv modeling. In *Geometric Science of Information*, pages 140–148. Springer Berlin Heidelberg, 2013. 48, 56, 63, 117, 145, 167

[76] Hirotugu Akaike. Information theory and an extension of the maximum likelihood principle. *Second International Symposium on Information Theory*, pages 267–281, 1973. 57, 154

[77] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6) :716–723, 1974. 57, 154

[78] Gideon E. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2) :461–464, 1978. 57, 154

[79] J. Burbea and C. Rao. On the convexity of some divergence measures based on entropy functions. *Information Theory, IEEE Transactions on*, 28(3) :489–495, 1982. 59, 82

[80] G. Verdoolaege and P. Scheunders. The Geometry of Multivariate Generalized Gaussian Models — Part I : Metric and Geodesic Equations. The part 2 exist also, Jan. 2009. 60

[81] M.M. Deza and E. Deza. *Encyclopedia of Distances*. Springer Verlag, 2009. 60

[82] C Radhakrishna Rao. Diversity and dissimilarity coefficients : a unified approach. *Theoretical Population Biology*, 21(1) :24–43, 1982. 61, 82

[83] S. Amari and H. Nagaoka. *Methods of information geometry*. Amer Mathematical Society, 2007. 61, 137

[84] Aurélien Schutz, Lionel Bombrun, and Yannick Berthoumieu. Intrinsic prior for bayesian classification of texture images. *IEEE International Conference on Acoustic, Speech and Signal Processing*, pages 4392–4396, 2014. 63, 73, 145, 167

[85] Xavier Pennec. Probabilities and statistics on riemannian manifolds : Basic tools for geometric measurements. In *Proc. of Nonlinear Signal and Image Processing (NSIP'99)*, volume 1, pages 194–198, 1999. 65, 67

[86] Richard A. Johnson. Asymptotic expansions associated with posterior distributions. *The Annals of Mathematical Statistics*, 41(3) :851–864, Jun. 1970. 69

[87] D.V. Lindley. Approximate bayesian methods. *Trabajos de Estadistica Y de Investigacion Operativa*, 31(1) :223–245, 1980.

[88] Luke Tierney and Joseph B. Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393) :82–86, 1986. 69

[89] Yoichi Miyata. Fully exponential laplace approximations using asymptotic modes. *Journal of the American Statistical Association*, 99(468) :1037–1049, 2004. 69, 70

[90] Luke Tierney, Robert E. Kass, and Joseph B. Kadane. Fully exponential laplace approximations to expectations and variances of nonpositive functions. *Journal of the American Statistical Association*, 84(407) :710–716, 1989. 70

[91] C. C. Heyde and I. M. Johnstone. On asymptotic posterior normality for stochastic processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(2) :184–189, 1979. 70

[92] Salem Said, Lionel Bombrun, and Yannick Berthoumieu. New riemannian priors on the univariate normal model. *Entropy*, 16(7) :4015–4031, 2014. 73

[93] N. Mitianoudis and T. Stathaki. Overcomplete source separation using laplacian mixture models. *IEEE Signal Processing Letters*, 12(4) :277–280, 2005. 74

[94] A. Banerjee, S. Merugu, I.S. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *The Journal of Machine Learning Research*, 6 :1705–1749, 2005. 74, 86

[95] F. Nielsen. k-mle : A fast algorithm for learning statistical mixture models. *arXiv*, Mar. 23 2012. 74, 75, 167

[96] H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 1977. 82

[97] G.A. Galperin. A concept of the mass center of a system of material points in the constant curvature spaces. *Communications in Mathematical Physics*, 154(1) :63–84, 1993. 82

[98] Dario A. Bini and Bruno Iannazzo. Computing the karcher mean of symmetric positive definite matrices. *Linear Algebra and its Applications*, 438(4) :1700 – 1710, 2013. 16th {ILAS} Conference Proceedings, Pisa 2010. 82

[99] Maher Moakher. On the averaging of symmetric positive-definite tensors. *Journal of Elasticity*, 82(3) :273–296, 2006. 82

[100] F Barbaresco. New foundation of radar doppler signal processing based on advanced differential geometry of symmetric spaces : Doppler matrix cfar and radar application. In *International Radar Conference*, 2009. 82

[101] J. Lapuyade-Lahorgue and F. Barbaresco. Radar detection using siegel distance between autoregressive processes, application to hf and x-band radar. In *Radar Conference, 2008. RADAR '08. IEEE*, pages 1–6, May 2008. 82

[102] P. G. Batchelor, M. Moakher, D. Atkinson, F. Calamante, and A. Connelly. A rigorous framework for diffusion tensor calculus. *Magnetic Resonance in Medicine*, 53(1) :221–225, 2005. 82

[103] P. T. Fletcher and S. Joshi. Riemannian geometry for the statistical analysis of diffusion tensor data. *Signal Processing*, 87(2) :250 – 262, 2007. <ce :title>Tensor Signal Processing</ce :title>.

[104] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1) :41–66, 2006. 82

[105] Y. Rathi, A Tannenbaum, and O. Michailovich. Segmenting images on the tensor manifold. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007. 82

[106] F. Nielsen and R. Nock. Sided and symmetrized bregman centroids. *IEEE Transactions on Information Theory*, 55(6) :2048–2059, June 2009. 85, 86, 87, 98, 139

[107] Frank Nielsen and Vincent Garcia. Statistical exponential families : A digest with flash cards. *arXiv :0911.4863*, 1 :1–27, Nov. 25 2009. 88, 116

[108] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999. 89

[109] S. Amari and S.C. Douglas. Why natural gradient ? In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 1213 –1216 vol.2, may 1998. 95, 107, 166

[110] T. J. Dekker. Finding a zero by means of successive linear interpolation. In B. Dejon and P. Henrici, editors, *Constructive aspects of the fundamental theorem of algebra*. Interscience, New York, 1969. 117

[111] R. P. Brent. An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*, 14(4) :422–425, 1971. 117

[112] Ray A Jarvis. A perspective on range finding techniques for computer vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1(2) :122–139, 1983. 128

[113] Stan Z Li. *Markov random field modeling in computer vision*. Springer-Verlag New York, Inc., 1995.

[114] David A Forsyth and Jean Ponce. *Computer vision : a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[115] Chi-hau Chen, Louis-François Pau, and Patrick Shen-pei Wang. *Handbook of pattern recognition and computer vision*. World Scientific, 2010. 128

[116] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(8) :837–842, 1996. 128

[117] Anne H Schistad Solberg, Torfinn Taxt, and Anil K Jain. A markov random field model for classification of multisource satellite imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 34(1) :100–113, 1996.

[118] Christodoulos I Christodoulou, Silas C Michaelides, and Constantinos S Pattichis. Multifeature texture analysis for the classification of clouds in satellite imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 41(11) :2662–2668, 2003. 128

[119] C-C Chen, John S DaPonte, and Martin D Fox. Fractal feature analysis and classification in medical imaging. *Medical Imaging, IEEE Transactions on*, 8(2) :133–142, 1989. 128

[120] Anthony Yezzi Jr, Satyanad Kichenassamy, Arun Kumar, Peter Olver, and Allen Tannenbaum. A geometric snake model for segmentation of medical imagery. *Medical Imaging, IEEE Transactions on*, 16(2) :199–209, 1997.

[121] Tsai andy, Anthony Yezzi Jr, William Wells, Clare Tempany, Dewey Tucker, Ayres Fan, W Eric Grimson, and Alan Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *Medical Imaging, IEEE Transactions on*, 22(2) :137–154, 2003. 128

[122] Paul Ramdohr. *The ore minerals and their intergrowths.* Pergamon press, Oxford, New York, Toronto, Sidney, Paris, Frankfurt, 1980. 128

[123] RP King. Linear stochastic models for mineral liberation. *Powder Technology*, 81(3) :217–234, 1994.

[124] E Donskoi, SP Suthers, SB Fradd, JM Young, JJ Campbell, TD Raynlyn, and JMF Clout. Utilization of optical image analysis and automatic texture classification for iron ore particle characterisation. *Minerals Engineering*, 20(5) :461–471, 2007.

[125] A Suresh, USN Raju, A Nagaraja Rao, and V Vijaya Kumar. An innovative technique of marble texture description based on grain components. *International Journal of Computer Science and Network Security*, 8(2) :122–126, 2008. 128

[126] Prithvi S Kandhal, John B Motter, and Maqbool Ahmed Khatri. Evaluation of particle shape and texture : manufactured versus natural sands. Technical report, National Center for Asphalt Technology, 1991. 128

[127] Topi Mäenpää, Markus Turtinen, and Matti Pietikäinen. Real-time surface inspection by texture. *Real-Time Imaging*, 9(5) :289–296, 2003.

[128] Xianghua Xie. A review of recent advances in surface defect detection using texture analysis techniques. *Electronic Letters on Computer Vision and Image Analysis*, 7(3) :1–22, 2008. 128

[129] A.K. Jain, R. P W Duin, and Jianchang Mao. Statistical pattern recognition : a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1) :4–37, 2000. 128

[130] R.M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5) :786–804, 1979. 128

[131] Hans J. Bremermann. Pattern recognition, functionals and entropy. *Biomedical Engineering, IEEE Transactions on*, BME-15(3) :201 –207, july 1968.

[132] H. Raïffa and R. Schlaifer. *Applied statistical decision theory.* M.I.T. Press, 1968.

[133] R.M. Haralick and G. L. Kelly. Pattern recognition with measurement space and spatial clustering for multiple images. *Proceedings of the IEEE*, 57(4) :654–665, 1969.

[134] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., Wiley-Interscience, Stanford Research Institute, Menlo Park, California, 1973.

[135] J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles*. Coden : APMCC, Addison-Wesley Publishing Company, Inc., 1974. 128

[136] MIT Vision and Modeling Group. Vision Texture. Disponible : http ://vismod.media.mit.edu/pub/VisTex. 130

[137] J. Puzicha, T. Hofmann, and J.M. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *Computer Vision and Pattern Recognition, 1997. Proceedings. 1997 IEEE Computer Society Conference on*, pages 267–272, jun 1997. 130

[138] P. Brodatz. *Textures : a photographic album for artists and designers*, volume 66. Dover New York, 1966. 130

[139] Jing Zhang, Lei Wang, and Longzheng Tong. Feature reduction and texture classification in mri-texture analysis of multiple sclerosis. In *Complex Medical Engineering, 2007. CME 2007. IEEE/ICME International Conference on*, pages 752 –757, may 2007. 131

[140] G. Van de Wouwer, P. Scheunders, and D. Van Dyck. Statistical texture characterization from discrete wavelet representations. *Image Processing, IEEE Transactions on*, 8(4) :592–598, 2002. 131

[141] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7) :971–987, Jul 2002. 131

[142] Fernando Roberti de Siqueira, William Robson Schwartz, and Helio Pedrini. Multi-scale gray level co-occurrence matrices for texture description. *Neurocomputing*, 1(0) :–, 2013. 131

[143] Computer Vision Laboratory Colombia university. Columbia-utrecht reflectance and texture database. 132

[144] W Weibull. A statistical distribution function of wide applicability. *Journal of Applied Mechanics*, 13 :293–297, 1951. 133

[145] R. V. Hogg and A. T. Craig. *Introduction to Mathematical Statistics, 4th edition.* New York : Macmillan, 1978. 133

[146] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500) :2319–2323, 2000. 139

[147] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerishe Mathematik*, 1 :269–271, 1959. 140

[148] R. Chellappa and S. Chatterjee. Classification of textures using gaussian markov random fields. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 33(4) :959 – 963, aug 1985. 141

[149] D. Gomez and J. Montero. Determining the accuracy in image supervised classification problems. *EUSFLAT*, 1 - 1 :342–349, Jul. 2011. 141

[150] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20 :37–46, 1960. 142

[151] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 2001. 156

[152] D. Fowlkes, C.and Martin, X. Ren, and J. Malik. Detecting and localizing boundaries in natural images. Technical report, University of California at Berkeley, 2002. 156

[153] T. Chang and C.-C.J. Kuo. Texture analysis and classification with tree-structured wavelet transform. *Image Processing, IEEE Transactions on*, 2(4) :429–441, 1993. 158

Index

Le travail présenté dans cette thèse a pour objectif de proposer un algorithme de classification supervisée d'images texturées basée sur la modélisation multivariée de champs texturaux. Inspiré des algorithmes de classification dits à « Sac de Mots Visuels » (SMV), nous proposons une extension originale au cas des descripteurs paramétriques issus de la modélisation multivariée des coefficients des sous-bandes d'une décomposition en ondelettes. Différentes contributions majeures de cette thèse peuvent être mises en avant. La première concerne l'introduction d'une loi *a priori* intrinsèque à l'espace des descripteurs par la définition d'une loi gaussienne concentrée. Cette dernière étant caractérisée par un barycentre $\bar{\theta}$ et une variance σ^2, nous proposons un algorithme d'estimation de ces deux quantités. Nous proposons notamment une application au cas des modèles multivariés SIRV (*Spherically Invariant Random Vector*), en séparant le problème complexe d'estimation du barycentre comme la résolution de deux problèmes d'estimation plus simples (un sur la partie gaussienne et un sur le multiplieur). Afin de prendre en compte la diversité naturelle des images texturées (contraste, orientation, ...), nous proposons une extension au cas des modèles de mélanges permettant ainsi de construire le dictionnaire d'apprentissage. Enfin, nous validons cet algorithme de classification sur diverses bases de données d'images texturées et montrons de bonnes performances de classification vis-à-vis d'autres algorithmes de la littérature.

Mots clés : classification, diversité intra-classe, loi a priori intrinsèque, modèles multivariés, texture.

The prime objective of this thesis is to propose an unsupervised classification algorithm of textured images based on multivariate stochastic models. Inspired from classification algorithm named "Bag of Words" (BoW), we propose an original extension to parametric descriptors issued from the multivariate modeling of wavelet subband coefficients. Some major contributions of this thesis can be outlined. The first one concerns the introduction of an intrinsic prior on the parameter space by defining a Gaussian concentrated distribution. This latter being characterized by a centroid $\bar{\theta}$ and a variance σ^2, we propose an estimation algorithm for those two quantities. Next, we propose an application to the multivariate SIRV (*Spherically Invariant Random Vector*) model, by resolving the difficult centroid estimation problem as the solution of two simpler ones (one for the Gaussian part and one for the multiplier part). To handle with the intra-class diversity of texture images (scene enlightenment, orientation ...), we propose an extension to mixture models allowing the construction of the training dictionary. Finally, we validate this classification algorithm on various texture image databases and show interesting classification performances compared to other state-of-the-art algorithms.

Keywords : classification, intra-class diversity, intrinsic prior, multivariate models, texture.